



開發人員指南

# Amazon Transcribe



# Amazon Transcribe: 開發人員指南

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon 的商標和商業外觀不得用於任何非 Amazon 的產品或服務，也不能以任何可能造成客戶混淆、任何貶低或使 Amazon 名譽受損的方式使用 Amazon 的商標和商業外觀。所有其他非 Amazon 擁有的商標均為其各自擁有者的財產，這些擁有者可能附屬於 Amazon，或與 Amazon 有合作關係，亦或受到 Amazon 贊助。

# Table of Contents

什麼是 Amazon Transcribe ? .....	1
Amazon Transcribe和 HIPAA 資格 .....	1
定價 .....	1
區域可用性和配額 .....	2
可用的功能 .....	4
支援的語言 .....	6
支援的程式設計語言 .....	13
字元集 .....	14
阿布哈茲文 .....	18
南非荷蘭文 .....	20
Arabic .....	21
阿斯圖里亞文 .....	22
亞塞拜然文 .....	22
亞美尼亞文 .....	23
巴什基爾文 .....	24
巴斯克文 .....	27
白俄羅斯文 .....	27
孟加拉文 .....	28
波士尼亞文 .....	30
保加利亞文 .....	31
加泰隆尼亞文 .....	32
中庫德文 .....	33
簡體中文 .....	34
繁體中文 .....	35
克羅埃西亞文 .....	36
捷克文 .....	36
丹麥文 .....	37
荷蘭文 .....	38
英文 .....	39
Estonian .....	39
波斯文 .....	39
芬蘭文 .....	41
French .....	41
加利西亞文 .....	42

喬治亞文 .....	43
德文 .....	44
Greek .....	45
古吉拉特文 .....	46
豪沙文 .....	48
Hebrew .....	48
北印度文 .....	49
匈牙利文 .....	52
冰島文 .....	52
印尼文 .....	53
義大利文 .....	53
日文 .....	54
卡拜爾文 .....	54
坎那達文 .....	55
哈薩克文 .....	57
盧安達文 .....	58
韓文 .....	59
吉爾吉斯文 .....	59
拉脫維亞文 .....	62
立陶宛文 .....	62
盧干達文 .....	63
馬其頓文 .....	63
馬來文 .....	66
馬來亞拉姆文 .....	66
馬爾他文 .....	68
馬拉地文 .....	69
草原馬里文 .....	71
Mongolian .....	73
書面挪威語 .....	75
歐迪亞文 .....	76
普什圖文 .....	78
Polish .....	80
葡萄牙文 .....	80
旁遮普文 .....	82
羅馬尼亞文 .....	83
俄文 .....	84



塞爾維亞文 .....	85
僧伽羅文 .....	87
斯洛伐克文 .....	90
斯洛維尼亞文 .....	90
索馬利亞文 .....	91
西班牙文 .....	91
巽他文 .....	92
史瓦西里文 .....	93
瑞典文 .....	93
他加祿語/菲律賓文 .....	94
坦米爾文 .....	94
韃靼語 .....	96
特拉古 .....	98
Thai .....	100
Turkish .....	102
烏克蘭文 .....	103
維吾爾語 .....	104
烏茲別克文 .....	107
越南文 .....	108
威爾斯文 .....	112
沃洛夫文 .....	113
祖魯文 .....	114
運作方式 .....	115
資料輸入和輸出 .....	115
媒體格式 .....	116
音訊頻道 .....	117
取樣率 .....	117
輸出 .....	117
轉錄號碼 .....	120
入門 .....	126
註冊AWS 帳戶 .....	126
安裝AWS CLI和軟體開發套件 .....	127
設定IAM認證 .....	127
建立Amazon S3值區 .....	128
建立 IAM 政策 .....	128
使用轉錄AWS Management Console .....	130

使用轉錄AWS CLI .....	140
開始新的轉錄工作 .....	140
取得轉錄任務的狀態 .....	141
列出您的轉錄工作 .....	142
刪除轉錄工作 .....	143
使用 SDK 進行轉錄 AWS .....	143
使用 AWS 軟體開發套件 .....	155
使用 HTTP 或進行轉錄 WebSockets .....	156
串流轉錄 .....	158
最佳實務 .....	159
串流和部分結果 .....	159
部分結果穩定 .....	161
設定串流轉錄 .....	165
事件串流編碼 .....	178
資料框架 .....	180
Job 佇列 .....	181
啟用任務務務務務務 .....	181
標記 資源 .....	185
標籤型存取控制 .....	186
將標籤新增到您的Amazon Transcribe資源 .....	186
分隔發言者 (日記) .....	190
在批次轉錄中對發言者進行分隔 .....	190
在串流轉錄中分隔發言者 .....	194
範例輸出 .....	196
轉錄多聲道音訊 .....	202
在批次轉錄中使用頻道識別 .....	203
在串流轉錄中使用頻道識別 .....	207
範例輸出 .....	208
識別語言 .....	216
批次語言識別 .....	216
識別多語言音訊中的語言 .....	217
提高語言識別準確性 .....	218
結合語言識別與其他 Amazon Transcribe 功能 .....	218
使用語言識別與批次轉錄 .....	219
串流語言識別 .....	225
識別多語言音訊中的語言 .....	226

透過串流媒體使用語言識別 .....	227
替代轉錄 .....	233
請求替代轉錄 .....	235
提高轉錄 .....	240
自訂詞彙 .....	241
自訂詞彙表與清單 .....	242
使用資料表建立自訂詞彙 .....	242
使用清單建立自訂字彙 .....	252
使用自訂字彙 .....	255
自訂語言模型 .....	262
資料來源 .....	262
訓練與調整資料 .....	263
建立自訂語言模型 .....	263
使用自訂語言模型 .....	268
篩選單字 .....	276
建立字彙篩選器 .....	277
建立自訂字彙篩選器 .....	278
使用自訂詞彙篩選器 .....	282
在批次轉錄中使用自訂詞彙篩選 .....	255
在串流轉錄中使用自訂詞彙篩選器 .....	259
偵測有毒語音 .....	291
使用有毒語音檢測 .....	292
在批量轉錄中使用有毒語音檢測 .....	292
範例輸出 .....	295
編輯成績單 .....	298
在批次工作中修訂 PII .....	298
在即時串流中修訂或識別 PII .....	304
範例輸出 .....	310
編輯輸出的示例 ( 批處理 ) .....	310
編輯流輸出示例 .....	312
PII 識別輸出範例 .....	314
組建字幕 .....	316
產生字幕檔 .....	317
分析客服中心音訊 .....	320
常用案例 .....	320
考量事項和其他資訊 .....	321

區域可用性和配額 .....	322
通話後分析 .....	323
通話後見解 .....	323
建立類別 .....	325
開始轉錄 .....	336
通話後分析輸出 .....	345
啟用生成式通話摘要 .....	358
即時通話分析 .....	362
即時見解 .....	362
建立類別 .....	364
具有即時轉錄功能的通話後分析 .....	371
開始轉錄 .....	378
即時通話分析輸出 .....	386
轉錄Amazon Chime呼叫 .....	391
程式碼範例 .....	392
動作 .....	393
CreateVocabulary .....	393
DeleteMedicalTranscriptionJob .....	396
DeleteTranscriptionJob .....	399
DeleteVocabulary .....	403
GetTranscriptionJob .....	405
GetVocabulary .....	408
ListMedicalTranscriptionJobs .....	410
ListTranscriptionJobs .....	415
ListVocabularies .....	422
StartMedicalTranscriptionJob .....	425
StartStreamTranscriptionAsync .....	436
StartTranscriptionJob .....	440
UpdateVocabulary .....	459
案例 .....	462
建立和優化自訂詞彙 .....	463
轉錄音訊並取得工作資料 .....	473
跨服務範例 .....	483
建置 Amazon Transcribe 應用程式 .....	484
建置 Amazon Transcribe 串流應用程式 .....	484
將文字轉換為語音然後返回文字 .....	485

安全性 .....	487
身分和存取權管理 .....	487
物件 .....	488
使用身分驗證 .....	488
使用政策管理存取權 .....	491
如何與 IAM Amazon Transcribe 搭配使用 .....	493
預防混淆代理人 .....	499
身分型政策範例 .....	500
故障診斷 .....	508
資料保護 .....	509
網際網路流量隱私權 .....	510
資料加密 .....	510
選擇不使用您的資料來改善服務 .....	513
監控 Amazon Transcribe .....	513
使用監控 CloudWatch .....	514
Amazon Transcribe 使用監控 CloudTrail .....	514
Amazon EventBridge 搭配使用 Amazon Transcribe .....	518
法規遵循驗證 .....	524
恢復能力 .....	525
基礎設施安全性 .....	526
弱點分析與管理 .....	526
VPC 端點 (AWS PrivateLink) .....	526
共用子網路 .....	528
安全最佳實務 .....	528
Amazon Transcribe Medical .....	530
區域可用性和配額 .....	531
醫療 .....	532
轉錄醫學術語和測量 .....	533
轉錄號碼 .....	535
轉錄醫學對話 .....	537
轉錄音訊檔 .....	538
轉錄即時串流 .....	542
啟用喇叭分割 .....	545
轉錄多聲道音訊 .....	554
轉錄醫療聽寫 .....	561
轉錄音訊檔案 .....	562

轉錄流式醫療聽寫 .....	567
創建和使用醫療自定義詞彙 .....	569
為您的醫療自定義詞彙創建文本文件 .....	570
使用文本文件創建醫學自定義詞彙 .....	574
使用醫學自定義詞彙轉錄音頻文件 .....	575
使用醫學自定義詞彙轉錄實時流 .....	577
字符集, 為,Amazon Transcribe 醫學 .....	580
識別成績單中的 PHI .....	581
識別音頻文件中的 PHI .....	582
識別即時串流中的 PHI .....	586
生成替代轉錄 .....	588
VPC 端點 (AWS PrivateLink) .....	590
的注意事項Amazon Transcribe醫療 VPC 端點 .....	591
為建立介面 VPC 端點Amazon Transcribe醫療 .....	591
建立 VPC 端點原則Amazon Transcribe醫學串流 .....	591
共用子網路 .....	592
AWS HealthScribe .....	593
文字記錄檔案 .....	594
臨床文件檔案 .....	595
開始 AWS HealthScribe 工作 .....	596
範例輸出 .....	598
AWS HealthScribe 的靜態資料加密 .....	610
建立客戶受管金鑰 .....	611
為 AWS HealthScribe 指定客戶受管金鑰 .....	613
AWS KMS 加密內容 .....	613
文件歷史紀錄 .....	614
AWS 詞彙表 .....	622
.....	dcxxiii

# 什麼是 Amazon Transcribe ？

Amazon Transcribe是一種自動語音辨識服務，使用機器學習模型將音訊轉換為文字。您可以Amazon Transcribe作為獨立轉錄服務使用，也可以將speech-to-text功能添加到任何應用程式中。

您可以利用Amazon Transcribe語言自訂功能來改善特定使用案例的準確性、篩選內容以確保客戶隱私或受眾適用的語言、分析多聲道音訊中的內容、分割個別發言者的語音等。

您可以即時轉錄媒體 (串流)，也可以轉錄位於Amazon S3值區 (批次) 中的媒體檔案。若要查看每種類型的轉錄支援哪些語言，請參閱[支援的語言和特定語言功能](#)表格。

## 主題

- [Amazon Transcribe和 HIPAA 資格](#)
- [定價](#)
- [區域可用性和配額](#)

請參閱[什麼是Amazon Transcribe ？](#) 對於這項服務的短視頻導覽。

如需進一步了解，請參閱[Amazon Transcribe 的運作方式](#)和[Amazon Transcribe 入門](#)。

### Tip

有關 Amazon TranscribeAPI 的信息位於 [API 參考](#)中。

## Amazon Transcribe和 HIPAA 資格

Amazon Transcribe受 HIPAA 資格和 BAA AWS 的涵蓋範圍，該資格要求 BAA 客戶在使用時對靜態和傳輸中的所有 PHI 進行加密。自動 PHI 識別功能可在所有Amazon Transcribe運作地區免費使用。如需詳細資訊，請參閱 [HIPAA 資格和 B AA](#)。

## 定價

Amazon Transcribe是一項pay-as-you-go服務；定價是以錄製音訊的秒數為基礎，按月計費。

用量以 1 秒遞增量改進方式進行計費，最低計費標準為 15 秒。請注意，PII 內容編輯和自訂語言模型等功能需要支付額外費用。

如需各項的成本資訊AWS 區域，請參閱[Amazon Transcribe定價](#)。


## 區域可用性和配額

Amazon Transcribe支援下列項目AWS 區域：

區域	轉錄類型
af-south-1 (開普敦)	批次組合, 串流
ap-east-1 (香港) — ap-east-1 (香港)	批次
ap-northeast-1 (東京)	批次組合, 串流
ap-northeast-2 (首爾)	批次組合, 串流
ap-south-1 (孟買)	批次組合, 串流
ap-southeast-1 (新加坡)	批次組合, 串流
ap-southeast-2 (雪梨)	批次組合, 串流
ca-central-1 (加拿大中部) (加拿大、中部)	批次組合, 串流
eu-central-1 (法蘭克福)	批次組合, 串流
eu-north-1 (斯德哥爾摩)	批次
eu-west-1 (愛爾蘭)	批次組合, 串流
eu-west-2 (倫敦)	批次組合, 串流
eu-west-3 (巴黎)	批次
me-south-1 (巴林)	批次
sa-east-1 (聖保羅)	批次組合, 串流
us-east-1 (維吉尼亞北部)	批次組合, 串流
us-east-2 (俄亥俄州) (俄亥俄)	批次組合, 串流



區域	轉錄類型
us-gov-east-1 (GovCloud, 美國東部)	批次組合, 串流
us-gov-west-1 (GovCloud, 美國西部)	批次組合, 串流
us-west-1 (舊金山)	批次
us-west-2 (奧勒岡)	批次組合, 串流

 Important

Amazon Transcribe、[Amazon Transcribe Medical](#)和[呼叫分析](#)的區域支援不同。

若要取得每個支援區域的端點，請參閱AWS一般參考中的[服務端點](#)。

如需與您轉錄相關的配額清單，請參閱一般參考資料中的[服務配額](#)。AWS部分配額可根據要求變更。如果「可調整」欄包含「是」，您可以要求增加。若要這樣做，請選取提供的連結。


# Amazon Transcribe features

為了協助您決定哪種 Amazon Transcribe 解決方案最適合您的使用案例，下表提供了功能比較。

請注意，「批次處理」和 'post-call' 指轉錄位於 Amazon S3 儲存貯體中的檔案和「串流」，且 'real-time' 指即時轉錄媒體。

功能	Amazon Transcribe	<a href="#">Amazon Transcribe Medical</a> <sup>1</sup>	<a href="#">Amazon Transcribe 通話分析</a>
<b>組態選項</b>			
<a href="#">替代轉錄</a>	批次，串流	批次，串流	編號
<a href="#">聲道識別</a>	批次，串流	批次，串流	post-call, real-time
<a href="#">工作佇列</a>	批次	編號	post-call
<a href="#">語言識別</a>	批次，串流	編號	post-call
<a href="#">多語言識別</a>	批次，串流	編號	編號
<a href="#">發言者日記</a>	批次，串流	批次，串流	post-call
<a href="#">轉錄數字</a> <sup>2</sup>	批次，串流	批次，串流	post-call, real-time
<b>對話分析</b>			
<a href="#">通話特徵</a>	編號	編號	post-call
<a href="#">通話摘要</a> <sup>2</sup>	編號	編號	post-call
<a href="#">自訂分類</a>	編號	編號	post-call
<a href="#">即時類別事件</a>	編號	編號	real-time
<a href="#">即時問題偵測</a> <sup>2</sup>	編號	編號	real-time
<a href="#">即時發言者情緒</a>	編號	編號	real-time
<a href="#">發言者情緒</a>	編號	編號	post-call

功能	Amazon Transcribe	<a href="#">Amazon Transcribe Medical</a> <sup>1</sup>	<a href="#">Amazon Transcribe 通話分析</a>
語言自訂			
<a href="#">自訂語言模型</a> <sup>2</sup>	批次，串流	編號	post-call, real-time
<a href="#">自訂詞彙</a>	批次，串流	批次，串流	post-call, real-time
資源組織			
<a href="#">標記</a>	批次	批次	post-call
敏感資料			
<a href="#">識別個人健康資訊</a> <sup>2</sup>	編號	批次，串流	編號
<a href="#">識別個人身分識別資訊</a> <sup>2</sup>	串流	編號	real-time
<a href="#">修訂音訊</a> <sup>2</sup>	編號	編號	post-call, real-time
<a href="#">修訂文字記錄</a> <sup>2</sup>	批次，串流	編號	post-call, real-time
<a href="#">詞彙篩選</a>	批次，串流	編號	post-call, real-time
影片			
<a href="#">字幕</a>	批次	編號	編號

 <sup>1</sup> Amazon Transcribe 醫療服務僅提供美國英語版本。

<sup>2</sup> 此功能不適用所有語言；如需詳細資訊，請參閱 [支援的語言和特定語言功能](#) 表格。

## 支援的語言和特定語言功能

下表列出支援 Amazon Transcribe 的語言；也列出了特定語言的功能。在繼續進行轉錄前，請確認您要使用的功能是否支援媒體中的語言。

若要檢視完整的 Amazon Transcribe 功能清單，請參閱[功能摘要](#)。

在下表中，「批處理」是指轉錄位於 Amazon S3 存儲桶中的媒體文件，而「流」是指實時轉錄流媒體。對於「呼叫分析」轉錄，'post-call'是指轉錄位於 Amazon S3 值區中的媒體檔案，'real-time'是指即時轉錄串流媒體。

語言	語言代碼	<a href="#">資料輸入</a>	<a href="#">轉錄號碼</a>	<a href="#">縮寫</a>	<a href="#">自訂語言模型</a>	<a href="#">修訂</a>	<a href="#">通話分析</a> * -
<a href="#">阿布哈茲文</a>	ab-GE	批次	編號	批次	編號	編號	編號
<a href="#">南非荷蘭文</a>	af-ZA	批次	編號	批次	編號	編號	編號
<a href="#">Arabic, 海灣</a>	ar-AE	批次	編號	編號	編號	編號	post-call
<a href="#">Arabic, 現代標準</a>	ar-SA	批次	編號	編號	編號	編號	編號
<a href="#">亞美尼亞文</a>	hy-AM	批次	編號	批次	編號	編號	編號
<a href="#">阿斯圖里亞文</a>	ast-ES	批次	編號	批次	編號	編號	編號
<a href="#">亞塞拜然文</a>	az-AZ	批次	編號	批次	編號	編號	編號
<a href="#">巴什基爾文</a>	ba-RU	批次	編號	批次	編號	編號	編號
<a href="#">巴斯克文</a>	eu-ES	批次	編號	批次	編號	編號	編號

語言	語言代碼	資料輸入	轉錄號碼	縮寫	自訂語言模型	修訂	通話分析 * -
<a href="#">白俄羅斯文</a>	be-BY	批次	編號	批次	編號	編號	編號
<a href="#">孟加拉文</a>	bn-IN	批次	編號	批次	編號	編號	編號
<a href="#">波士尼亞文</a>	bs-BA	批次	編號	批次	編號	編號	編號
<a href="#">保加利亞文</a>	bg-BG	批次	編號	批次	編號	編號	編號
<a href="#">加泰隆尼亞文</a>	ca-ES	批次	編號	批次	編號	編號	編號
<a href="#">中庫德文</a> ， <a href="#">伊朗</a>	ckb-IR	批次	編號	批次	編號	編號	編號
<a href="#">中庫德文</a> ， <a href="#">伊拉克</a>	ckb-IQ	批次	編號	批次	編號	編號	編號
<a href="#">簡體中文</a>	zh-CN	批次，串流	編號	編號	編號	編號	post-call
<a href="#">繁體中文</a>	zh-TW	批次	編號	編號	編號	編號	編號
<a href="#">克羅埃西亞文</a>	hr-HR	批次	編號	批次	編號	編號	編號
<a href="#">捷克文</a>	cs-CZ	批次	編號	批次	編號	編號	編號
<a href="#">丹麥文</a>	da-DK	批次	編號	批次	編號	編號	編號
<a href="#">荷蘭文</a>	nl-NL	批次	編號	批次	編號	編號	編號
<a href="#">英文</a> ， <a href="#">澳洲</a>	en-AU	批次，串流	批次，串流	批次，串流	批次，串流	串流	post-call, real-time

語言	語言代碼	<a href="#">資料輸入</a>	<a href="#">轉錄號碼</a>	<a href="#">縮寫</a>	<a href="#">自訂語言模型</a>	<a href="#">修訂</a>	<a href="#">通話分析</a> * -
<a href="#">英文</a> , 英國	en-GB	批次, 串流	批次, 串流	批次, 串流	批次, 串流	串流	post-call, real-time
<a href="#">英文</a> , 印度	en-IN	批次	批次	批次	編號	編號	post-call
<a href="#">英文</a> , 愛爾蘭	en-IE	批次	批次	批次	編號	編號	post-call
<a href="#">英文</a> , 紐西蘭	en-NZ	批次	批次	批次	編號	編號	編號
<a href="#">英文</a> , 蘇格蘭	en-AB	批次	批次	批次	編號	編號	post-call
<a href="#">英文</a> , 南非	en-ZA	批次	批次	批次	編號	編號	編號
<a href="#">英文</a> , 美國	en-US	批次, 串流	批次, 串流	批次, 串流	批次, 串流	批次, 串流	post-call, real-time
<a href="#">英文</a> , 威爾斯	en-WL	批次	批次	批次	編號	編號	post-call
<a href="#">Estonian</a>	et-ET	批次	編號	批次	編號	編號	編號
<a href="#">波斯文</a>	fa-IR	批次	編號	編號	編號	編號	編號
<a href="#">芬蘭文</a>	fi-FI	批次	編號	批次	編號	編號	編號
<a href="#">French</a>	fr-FR	批次, 串流	編號	批次, 串流	編號	編號	post-call, real-time
<a href="#">French</a> , 加拿大	fr-CA	批次, 串流	編號	批次, 串流	編號	編號	post-call, real-time

語言	語言代碼	資料輸入	轉錄號碼	縮寫	自訂語言模型	修訂	通話分析 * -
<a href="#">加利西亞文</a>	gl-ES	批次	編號	批次	編號	編號	編號
<a href="#">喬治亞文</a>	ka-GE	批次	編號	批次	編號	編號	編號
<a href="#">德文</a>	de-DE	批次，串流	批次，串流	批次，串流	批次，串流	編號	post-call, real-time
<a href="#">德文, 瑞士</a>	de-CH	批次	批次	批次	編號	編號	post-call
<a href="#">Greek</a>	el-GR	批次	編號	批次	編號	編號	編號
<a href="#">古吉拉特文</a>	gu-IN	批次	編號	批次	編號	編號	編號
<a href="#">豪沙文</a>	ha-NG	批次	編號	批次	編號	編號	編號
<a href="#">Hebrew</a>	he-IL	批次	編號	編號	編號	編號	編號
<a href="#">北印度文, 印度</a>	hi-IN	批次，串流	編號	批次，串流	批次	編號	post-call
<a href="#">匈牙利文</a>	hu-HU	批次	編號	批次	編號	編號	編號
<a href="#">冰島文</a>	is-IS	批次	編號	批次	編號	編號	編號
<a href="#">印尼文</a>	id-ID	批次	編號	批次	編號	編號	編號
<a href="#">義大利文</a>	it-IT	批次，串流	編號	批次，串流	編號	編號	post-call, real-time
<a href="#">日文</a>	ja-JP	批次，串流	編號	編號	批次，串流	編號	post-call
<a href="#">卡拜爾文</a>	kab-DZ	批次	編號	批次	編號	編號	編號
<a href="#">坎那達文</a>	kn-IN	批次	編號	批次	編號	編號	編號

語言	語言代碼	資料輸入	轉錄號碼	縮寫	自訂語言模型	修訂	通話分析 * -
<a href="#">哈薩克文</a>	kk-KZ	批次	編號	批次	編號	編號	編號
<a href="#">盧安達文</a>	rw-RW	批次	編號	批次	編號	編號	編號
<a href="#">韓文</a>	ko-KR	批次, 串流	編號	編號	編號	編號	post-call
<a href="#">吉爾吉斯文</a>	ky-KG	批次	編號	批次	編號	編號	編號
<a href="#">拉脫維亞文</a>	lv-LV	批次	編號	批次	編號	編號	編號
<a href="#">立陶宛文</a>	lt-LT	批次	編號	批次	編號	編號	編號
<a href="#">盧干達文</a>	lg-IN	批次	編號	批次	編號	編號	編號
<a href="#">馬其頓文</a>	mk-MK	批次	編號	批次	編號	編號	編號
<a href="#">馬來文</a>	ms-MY	批次	編號	批次	編號	編號	編號
<a href="#">馬來亞拉姆文</a>	ml-IN	批次	編號	批次	編號	編號	編號
<a href="#">馬爾他文</a>	mt-MT	批次	編號	批次	編號	編號	編號
<a href="#">馬拉地文</a>	mr-IN	批次	編號	批次	編號	編號	編號
<a href="#">草原馬里文</a>	mhr-RU	批次	編號	批次	編號	編號	編號
<a href="#">Mongolian</a>	mn-MN	批次	編號	批次	編號	編號	編號
<a href="#">書面挪威語</a>	no-NO	批次	編號	批次	編號	編號	編號
<a href="#">歐迪亞文</a>	or-IN	批次	編號	批次	編號	編號	編號



語言	語言代碼	資料輸入	轉錄號碼	縮寫	自訂語言模型	修訂	通話分析 * -
<a href="#">普什圖文</a>	ps-AF	批次	編號	批次	編號	編號	編號
<a href="#">Polish</a>	pl-PL	批次	編號	批次	編號	編號	編號
<a href="#">葡萄牙文</a>	pt-PT	批次	編號	批次	編號	編號	post-call
<a href="#">葡萄牙文, 巴西</a>	pt-BR	批次, 串流	編號	批次, 串流	編號	編號	post-call, real-time
<a href="#">旁遮普文</a>	pa-IN	批次	編號	批次	編號	編號	編號
<a href="#">羅馬尼亞文</a>	ro-RO	批次	編號	批次	編號	編號	編號
<a href="#">俄文</a>	ru-RU	批次	編號	編號	編號	編號	編號
<a href="#">塞爾維亞文</a>	sr-RS	批次	編號	批次	編號	編號	編號
<a href="#">僧伽羅文</a>	si-LK	批次	編號	批次	編號	編號	編號
<a href="#">斯洛伐克文</a>	sk-SK	批次	編號	批次	編號	編號	編號
<a href="#">斯洛維尼亞文</a>	sl-SI	批次	編號	批次	編號	編號	編號
<a href="#">索馬利亞文</a>	so-SO	批次	編號	批次	編號	編號	編號
<a href="#">西班牙文</a>	es-ES	批次	編號	批次	編號	編號	post-call
<a href="#">西班牙文, 美國</a>	es-US	批次, 串流	編號	批次, 串流	批次, 串流	批次, 串流	post-call, real-time
<a href="#">巽他文</a>	su-ID	批次	編號	批次	編號	編號	編號

語言	語言代碼	資料輸入	轉錄號碼	縮寫	自訂語言模型	修訂	通話分析 * -
<a href="#">史瓦西里文</a> ，肯亞	sw-KE	批次	編號	批次	編號	編號	編號
<a href="#">史瓦西里文</a> ，蒲隆地	sw-BI	批次	編號	批次	編號	編號	編號
<a href="#">史瓦西里文</a> ，盧安達	sw-RW	批次	編號	批次	編號	編號	編號
<a href="#">史瓦西里文</a> ，坦尚尼亞	sw-TZ	批次	編號	批次	編號	編號	編號
<a href="#">史瓦西里文</a> ，烏干達	sw-UG	批次	編號	批次	編號	編號	編號
<a href="#">瑞典文</a>	sv-SE	批次	編號	批次	編號	編號	編號
<a href="#">他加祿語/菲律賓文</a>	tl-PH	批次	編號	批次	編號	編號	編號
<a href="#">坦米爾文</a>	ta-IN	批次	編號	編號	編號	編號	編號
<a href="#">鞑靼語</a>	tt-RU	批次	編號	批次	編號	編號	編號
<a href="#">特拉古</a>	te-IN	批次	編號	編號	編號	編號	編號
<a href="#">Thai</a>	th-TH	批次，串流	編號	批次，串流	編號	編號	編號
<a href="#">Turkish</a>	tr-TR	批次	編號	批次	編號	編號	編號
<a href="#">烏克蘭文</a>	uk-UA	批次	編號	批次	編號	編號	編號

語言	語言代碼	<a href="#">資料輸入</a>	<a href="#">轉錄號碼</a>	<a href="#">縮寫</a>	<a href="#">自訂語言模型</a>	<a href="#">修訂</a>	<a href="#">通話分析</a> * -
<a href="#">維吾爾語</a>	ug-CN	批次	編號	批次	編號	編號	編號
<a href="#">烏茲別克文</a>	uz-UZ	批次	編號	批次	編號	編號	編號
<a href="#">越南文</a>	vi-VN	批次	編號	批次	編號	編號	編號
<a href="#">威爾斯文</a>	cy-WL	批次	編號	批次	編號	編號	編號
<a href="#">沃洛夫文</a>	wo-SN	批次	編號	批次	編號	編號	編號
<a href="#">祖魯文</a>	zu-ZA	批次	編號	批次	編號	編號	編號

\*以下通話分析見解僅在特定英語方言中支援：

- [呼叫摘要](#)：en-\*(所有英文方言)
- [問題偵測](#)：en-AU、en-GB、en-US

## 支援的程式設計語言

Amazon Transcribe 支援下列 AWS 軟體開發套件：

批次轉錄	串流轉錄
<a href="#">.NET</a>	串流不支援 .NET。
<a href="#">AWS 命令 CLI 介面</a>	CLI 不支援串流。
<a href="#">C++</a>	<a href="#">C++</a>
<a href="#">Go</a>	<a href="#">Go</a>
<a href="#">Java V2</a>	<a href="#">Java V2</a>
<a href="#">JavaScript</a>	<a href="#">JavaScript V3</a>

批次轉錄	串流轉錄
<a href="#">PHP V3</a>	<a href="#">PHP V3</a>
<a href="#">Python Boto3</a>	<a href="#">Python 流媒體開發套件 Amazon Transcribe</a>
<a href="#">Ruby V3</a>	<a href="#">Ruby V3</a>
<a href="#">Rust</a>	<a href="#">Rust</a>

如需搭配使用 SDK 的詳細資訊 Amazon Transcribe，請參閱[使用 SDK 進行轉錄 AWS](#)。

有關所有可用 AWS SDK 和構建器工具的更多信息，請參閱[構建工具](#)。AWS

### Tip

您可以在下列 GitHub 儲存庫中找到 SDK 程式碼範例：

- [AWS 程式碼範例](#)
- [Amazon Transcribe 範例](#)

## 自訂詞彙和詞彙篩選的字元集

對於 Amazon Transcribe 支援的每種語言，都有一組特定的 Amazon Transcribe 字元可以識別。當您建立自訂詞彙或詞彙篩選時，請僅使用語言字元集中列出的字元。如果您使用不支援的字元，您的自訂詞彙或詞彙篩選將無法正常使用。

### Important

請務必檢查您的自訂詞彙檔案是否僅使用下列字元集中列出的支援 Unicode 字碼點和字碼點序列。

常用字體中許多 Unicode 字元看似相同，即使使用不同的字碼點。僅支援本指南中列出的字碼點。例如，法文單字 déjà 可以使用預先編寫字元 (其中一個 Unicode 值代表重音字元) 或分解字元 (其中兩個 Unicode 值代表重音字元，一個值用於基本字元，另一個用於重音字元) 轉譯。

- 預先編寫版本：0064 **00E9** 006A **00E0**(轉譯為 déjà)

- 分解版本：0064 **0065 0301** 006A **0061 0300**(轉譯為 déjà)

## 主題

- [阿布哈茲文字元集](#)
- [南非荷蘭文字元集](#)
- [阿拉伯文字元集](#)
- [阿斯圖里亞文字元集](#)
- [亞塞拜然文字元集](#)
- [亞美尼亞文字元集](#)
- [巴什基爾文字元集](#)
- [巴斯克文字元集](#)
- [白俄羅斯文字元集](#)
- [孟加拉文字元集](#)
- [波士尼亞文字元集](#)
- [保加利亞文字元集](#)
- [加泰隆尼亞文字元集](#)
- [中庫德文字元集](#)
- [中文, 普通話 \(中國大陸\), 簡體字集](#)
- [中文, 國語 \(台灣\), 繁體字集](#)
- [克羅埃西亞文字元集](#)
- [捷克文字元集](#)
- [丹麥文字元集](#)
- [荷蘭文字元集](#)
- [英文字元集](#)
- [愛沙尼亞文字元集](#)
- [波斯文字元集](#)
- [芬蘭文字元集](#)
- [法文字元集](#)
- [加利西亞文字元集](#)

- [喬治亞文字元集](#)
- [德文字元集](#)
- [希臘文字元集](#)
- [古吉拉特文字元集](#)
- [豪薩文字元集](#)
- [希伯來文字元集](#)
- [印地文字元集](#)
- [匈牙利文字元集](#)
- [冰島文字元集](#)
- [印尼文字元集](#)
- [義大利文字元集](#)
- [日文字元集](#)
- [卡拜爾文字元集](#)
- [坎那達文字元集](#)
- [哈薩克文字元集](#)
- [盧安達文字元集](#)
- [韓文字元集](#)
- [吉爾吉斯文字元集](#)
- [拉脫維亞文字元集](#)
- [立陶宛文字元集](#)
- [盧干達文字元集](#)
- [馬其頓文字元集](#)
- [馬來文字元集](#)
- [馬來亞拉姆文字元集](#)
- [馬爾他文字元集](#)
- [馬拉地文字元集](#)
- [草原馬里文字元集](#)
- [蒙古文字元集](#)

- [書面挪威語字元集](#)
- [歐迪亞文字元集](#)
- [普什圖文字元集](#)
- [波蘭文字元集](#)
- [葡萄牙文字元集](#)
- [旁遮普文字元集](#)
- [羅馬尼亞文字元集](#)
- [俄文字元集](#)
- [塞爾維亞文字元集](#)
- [僧伽羅文字元集](#)
- [斯洛伐克文字元集](#)
- [斯洛維尼亞文字元集](#)
- [索馬利亞文字元集](#)
- [西班牙文字元集](#)
- [巽他文字元集](#)
- [史瓦希里文字元集](#)
- [瑞典文字元集](#)
- [他加祿語/菲律賓文字元集](#)
- [坦米爾文字元集](#)
- [韃靼語字元集](#)
- [特拉古文字元集](#)
- [泰文字元集](#)
- [土耳其文字元集](#)
- [烏克蘭文字元集](#)
- [維吾爾語字元集](#)
- [烏茲別克文字元集](#)
- [越南字元集](#)
- [威爾斯語字元集](#)
- [沃洛夫文字元集](#)

- [祖魯文字元集](#)

## 阿布哈茲文字元集

對於阿布哈茲文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
а	0430	лъ	0459
б	0431	ь	045A
в	0432	ћ	045B
г	0433	ќ	045C
д	0434	#	045D
е	0435	ђ	045E
ж	0436	џ	045F
з	0437	г	0491
и	0438	ґ	0493
й	0439	ж	0497
к	043A	з	0499
л	043B	қ	049B
м	043C	к	049F



字元	代碼	字元	代碼
н	043D	к	04A1
о	043E	қ	04A3
п	043F	ғ	04A5
р	0440	қ	04A9
с	0441	җ	04AB
т	0442	т	04AD
у	0443	ү	04AF
ф	0444	ұ	04B1
х	0445	х	04B3
ц	0446	ц	04B5
ч	0447	ч	04B7
ш	0448	h	04BB
щ	0449	е	04BD
ъ	044A	ё	04BF
ы	044B	#	04CA
ь	044C	ă	04D1
э	044D	ä	04D3
ю	044E	ě	04D7
я	044F	ə	04D9
#	0450	з	04E1

字元	代碼	字元	代碼
ë	0451	й	04E3
ђ	0452	ö	04E7
í	0453	ө	04E9
є	0454	ÿ	04EF
s	0455	ÿ	04F1
i	0456	ÿ	04F3
ï	0457	#	04F7
j	0458	Ы	04F9
#	0525		

## 南非荷蘭文字元集

對於南非荷蘭文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以使用下列 Unicode 字元：

字元	代碼	字元	代碼
á	00E1	ï	00EF
è	00E8	ó	00F3
é	00E9	ô	00F4
ê	00EA	ö	00F6

字元	代碼	字元	代碼
ë	00EB	ú	00FA
í	00ED	û	00FB
î	00EE	ü	00FC

## 阿拉伯文字元集

對於阿拉伯文自訂詞彙，您可以在 Phrase 欄位中使用下列 Unicode 字元。您也可以使用連字號 (-) 字元分隔單字。

字元	代碼	字元	代碼
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

字元	代碼	字元	代碼
ċ	062E	ċ	0646
ċ	062F	o	0647
ċ	0630	o	0648
ċ	0631	o	0649
ċ	0632	o	064A

## 阿斯圖里亞文字元集

對於阿斯圖里亞文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在此 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
á	00E1	ñ	00F1
é	00E9	ó	00F3
í	00ED	ú	00FA
ü	00FC		

## 亞塞拜然文字元集

對於亞塞拜然文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z

- -(連字號)
- .(句點)

您也可以在 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
ä	00E4	ǧ	011F
ç	00E7	ı	0131
ö	00F6	ş	015F
ü	00FC	ə	0259
.	0307		

## 亞美尼亞文字元集

對於亞美尼亞文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
մ	0561	ւ	0574
բ	0562	յ	0575
գ	0563	ի	0576
դ	0564	շ	0577

字元	代碼	字元	代碼
ɛ	0565	ɪ	0578
ɑ	0566	ʒ	0579
ɛ	0567	ɯ	057A
ɛ	0568	ɛ	057B
p	0569	ɪ	057C
ɔ	056A	u	057D
ɪ	056B	ɪ	057E
ɪ	056C	ɪ	057F
ɪ	056D	ɪ	0580
ɔ	056E	g	0581
ɪ	056F	ɪ	0582
h	0570	ɪ	0583
ɑ	0571	f	0584
ɪ	0572	o	0585
ɪ	0573	ɪ	0586

## 巴什基爾文字元集

對於巴什基爾文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
а	0430	лъ	0459
б	0431	њ	045A
в	0432	ћ	045B
г	0433	ќ	045C
д	0434	#	045D
е	0435	ђ	045E
ж	0436	џ	045F
з	0437	ѓ	0491
и	0438	Ѡ	0493
й	0439	ѡ	0497
к	043A	ѣ	0499
л	043B	Ѥ	049B
м	043C	Ѧ	049F
н	043D	ѧ	04A1
о	043E	Ѩ	04A3
п	043F	ѩ	04A5
р	0440	Ѫ	04A9
с	0441	Ѭ	04AB
т	0442	ѭ	04AD

字元	代碼	字元	代碼
у	0443	У	04AF
ф	0444	Ф	04B1
х	0445	Х	04B3
ц	0446	Ц	04B5
ч	0447	Ч	04B7
ш	0448	Һ	04BB
щ	0449	Ǝ	04BD
ъ	044A	Ǝ̇	04BF
ы	044B	#	04CA
ь	044C	ă	04D1
э	044D	ä	04D3
ю	044E	ě	04D7
я	044F	ə	04D9
#	0450	з	04E1
ë	0451	й	04E3
ђ	0452	ö	04E7
í	0453	ө	04E9
є	0454	ȳ	04EF
s	0455	ÿ	04F1
i	0456	ÿ̇	04F3



字元	代碼	字元	代碼
ï	0457	#	04F7
j	0458	ÿ	04F9

## 巴斯克文字元集

對於巴斯克文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在此 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
á	00E1	ñ	00F1
é	00E9	ó	00F3
í	00ED	ú	00FA
ü	00FC		

## 白俄羅斯文字元集

對於白俄羅斯文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在此 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
а	0430	с	0441
б	0431	т	0442
в	0432	у	0443
г	0433	ф	0444
д	0434	х	0445
е	0435	ц	0446
ж	0436	ч	0447
з	0437	ш	0448
й	0439	ы	044B
к	043A	ь	044C
л	043B	э	044D
м	043C	ю	044E
н	043D	я	044F
о	043E	ё	0451
п	043F	і	0456
р	0440	ŷ	045E

## 孟加拉文字元集

對於孟加拉文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)

- .(句點)

您也可以在此欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
ু	0981	দ	09A6
ং	0982	ধ	09A7
ঃ	0983	ন	09A8
অ	0985	প	09AA
আ	0986	ফ	09AB
ঈ	0987	ব	09AC
ঐ	0988	ভ	09AD
ঊ	0989	ম	09AE
ঋ	098A	য	09AF
ঌ	098B	র	09B0
এ	098F	ল	09B2
ঐ	0990	শ	09B6
ও	0993	ষ	09B7
ঔ	0994	স	09B8
ক	0995	হ	09B9
খ	0996	.	09BC
গ	0997	#	09BD
ঘ	0998	†	09BE

字元	代碼	字元	代碼
ঙ	0999	ঃ	09BF
চ	099A	঄	09C0
ছ	099B	অ	09C1
জ	099C	আ	09C2
ঝ	099D	ই	09C3
ঞ	099E	ঈ	09C4
ট	099F	ঊ	09C7
ঠ	09A0	ঋ	09C8
ড	09A1	ঌ	09CB
ঢ	09A2	঍	09CC
ণ	09A3	঎	09CD
ত	09A4	#	09CE
থ	09A5	ঐ	09D7

## 波士尼亞文字元集

對於波士尼亞文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以使用下列 Unicode 字元：

字元	代碼	字元	代碼
ć	0107	đ	0111
č	010D	š	0161
ž	017E		

## 保加利亞文字元集

對於保加利亞文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在此 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
а	0430	п	043F
б	0431	р	0440
в	0432	с	0441
г	0433	т	0442
д	0434	у	0443
е	0435	ф	0444
ж	0436	х	0445
з	0437	ц	0446
и	0438	ч	0447

字元	代碼	字元	代碼
й	0439	ш	0448
к	043A	щ	0449
л	043B	ъ	044A
м	043C	ь	044C
н	043D	ю	044E
о	043E	я	044F

## 加泰隆尼亞文字元集

對於加泰隆尼亞文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在此 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
à	00E0	ï	00EF
ç	00E7	ò	00F2
è	00E8	ó	00F3
é	00E9	ú	00FA
í	00ED	ü	00FC
ı	0140		

## 中庫德文字元集

對於中庫德文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在此 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
ئ	0626	م	0645
ا	0627	ن	0646
ب	0628	و	0648
ت	062A	پ	067E
ج	062C	چ	0686
ح	062D	ر	0695
خ	062E	ژ	0698
د	062F	ف	06A4
ر	0631	ک	06A9
ز	0632	گ	06AF
س	0633	ل	06B5
ش	0634	ھ	06BE
ع	0639	ؤ	06C6
غ	063A	ۇ	06C7

字元	代碼	字元	代碼
ف	0641	س	06CC
ق	0642	ئ	06CE
ج	0644	ه	06D5

## 中文, 普通話 (中國大陸), 簡體字集

對於簡體中文自訂詞彙，Phrase 欄位可以使用下列檔案中列出的任何字元。

- [zh-cn-character-set](#)

SoundsLike 欄位可以包含下列檔案中列出的拼音音節：

- [pinyin-character-set](#)

您在 SoundsLike 欄位中使用拼音音節時，請使用連字號 (-) 分隔音節。

Amazon Transcribe 使用數字代表簡體中文的四個音調。下表顯示如何為單字「ma」顯示音調標記。

音調	音調標記	音調數字
音調 1	mā	ma1
音調 2	má	ma2
音調 3	m	ma3
音調 4	mà	ma4

### Note

對於第 5 個 (中性) 音調，您可以使用音調 1，但「er」除外，必須對應到音調 2。例如，打转儿將以「da3-zhuan4-er2」表示。



簡體中文自訂詞彙不會使用此 IPA 欄位，但您仍必須在自訂詞彙表中加入 IPA 標頭。

以下範例是文字格式的輸入檔案。此範例會使用空格來對齊欄。您的輸入檔案應使用 TAB 字元分隔欄。僅在 DisplayAs 欄中包含空格。

Phrase	SoundsLike	IPA	DisplayAs
##	kang1-jian4		
##	qian3-ze2		
####	guo2-fang2-da4-chen2		
#####	shi4-jie4-bo2-lan3-hui4		###

## 中文, 國語 (台灣), 繁體字集

對於繁體中文自訂詞彙，Phrase 欄位可以使用下列檔案中列出的任何字元。

- [zh-tw-character-set](#)

SoundsLike 欄位可以包含下列檔案中列出的注音音節。

- [zhuyin-character-set](#)

您在 SoundsLike 欄位中使用注音音節時，請使用連字號 (-) 分隔音節。

Amazon Transcribe 使用數字表示繁體中文的四個音調。下表顯示如何為單字「ma」顯示音調標記。

音調	音調標記
音調 1	□Y
音調 2	□Y´
音調 3	□Y˘
音調 4	□Y`

繁體中文自訂詞彙不會使用此 IPA 欄位，但您仍必須在自訂詞彙表中加入 IPA 標頭。

以下範例是文字格式的輸入檔案。此範例會使用空格來對齊欄。您的輸入檔案應使用 TAB 字元分隔欄。僅在 DisplayAs 欄中包含空格。

Phrase	SoundsLike	IPA	DisplayAs
##	###`-##		
##	###~-##'		
####	###'-##'-##`-##~		
#####	#`-###~-##'-##~-###`	###	

## 克羅埃西亞文字元集

對於克羅埃西亞文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以可以在 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
ć	0107	đ	0111
č	010D	š	0161
ž	017E		

## 捷克文字元集

對於捷克文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以可以在 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
á	00E1	ď	010F
é	00E9	ě	011B
í	00ED	ň	0148
ó	00F3	ř	0159
ú	00FA	š	0161
ý	00FD	ť	0165
č	010D	ů	016F
ž	017E		

## 丹麥文字元集

對於丹麥文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- A - Z
- -(連字號)
- .(句點)

您也可以在此 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
Å	00C5	æ	00E6
Æ	00C6	é	00E9
Ø	00D8	ø	00F8
å	00E5		

## 荷蘭文字元集

對於荷蘭文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- A - Z
- '(單引號)
- -(連字號)
- .(句點)

您也可以在此 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
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

## 英文字元集

對於英文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- A - Z
- '(單引號)
- -(連字號)
- .(句點)

## 愛沙尼亞文字元集

對於愛沙尼亞文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在此 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
ä	00E4	ü	00FC
õ	00F5	š	0161
ö	00F6	ž	017E

## 波斯文字元集

對於波斯文自訂詞彙，您可以在 Phrase 欄位中使用下列字元。

字元	代碼	字元	代碼
ع	0621	ب	0638

字元	代碼	字元	代碼
آ	0622	ع	0639
أ	0623	غ	063A
ؤ	0624	ف	0641
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		

## 芬蘭文字元集

對於芬蘭文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在此 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
ä	00E4	ö	00F6
å	00E5	š	0161
ž	017E		

## 法文字元集

對於法文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- A - Z
- '(單引號)
- -(連字號)
- .(句點)

您也可以在此 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
À	00C0	à	00E0
Â	00C2	â	00E2

字元	代碼	字元	代碼
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

## 加利西亞文字元集

對於加利西亞文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以使用下列 Unicode 字元：

字元	代碼	字元	代碼
á	00E1	ñ	00F1



字元	代碼	字元	代碼
é	00E9	ó	00F3
í	00ED	ú	00FA
ü	00FC		

## 喬治亞文字元集

對於喬治亞文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在此 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
ა	10D0	ბ	10E0
ბ	10D1	ს	10E1
გ	10D2	ძ	10E2
დ	10D3	წ	10E3
ე	10D4	ჭ	10E4
ვ	10D5	ქ	10E5
ზ	10D6	ყ	10E6
თ	10D7	ყ	10E7
ო	10D8	შ	10E8
პ	10D9	ჩ	10E9

字元	代碼	字元	代碼
œ	10DA	ß	10EA
ð	10DB	đ	10EB
ḃ	10DC	ḡ	10EC
ṁ	10DD	ḟ	10ED
ǝ	10DE	ḥ	10EE
ɸ	10DF	Ꝥ	10EF
ḿ	10F0		

## 德文字元集

對於德文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- A - Z
- '(單引號)
- -(連字號)
- .(句點)

您也可以使用下列 Unicode 字元：

字元	代碼	字元	代碼
ä	00E4	Ä	00C4
ö	00F6	Ö	00D6
ü	00FC	Ü	00DC
ß	00DF		

## 希臘文字元集

對於希臘文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在此 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
ά	03AC	ν	03BD
έ	03AD	ξ	03BE
ή	03AE	ο	03BF
ί	03AF	π	03C0
ϋ	03B0	ρ	03C1
α	03B1	ς	03C2
β	03B2	σ	03C3
γ	03B3	τ	03C4
δ	03B4	υ	03C5
ε	03B5	φ	03C6
ζ	03B6	χ	03C7
η	03B7	ψ	03C8
θ	03B8	ω	03C9
ι	03B9	ϊ	03CA

字元	代碼	字元	代碼
κ	03BA	ü	03CB
λ	03BB	ó	03CC
μ	03BC	ώ	03CE
ï	0390		

## 古吉拉特文字元集

對於古吉拉特文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在此 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
ઁ	0A81	ઃ	0AA6
ઃ	0A82	ઘ	0AA7
ઃ	0A83	ઙ	0AA8
અ	0A85	ચ	0AAA
અ	0A86	જ	0AAB
ઇ	0A87	ઘ	0AAC
ઇ	0A88	ઙ	0AAD
ઉ	0A89	ઘ	0AAE
ઉ	0A8A	ચ	0AAF

字元	代碼	字元	代碼
ઋ	0A8B	૨	0AB0
ૠ	0A8D	૩	0AB2
ૡ	0A8F	૪	0AB3
ૢ	0A90	૫	0AB5
ૣ	0A91	૬	0AB6
૤	0A93	૭	0AB7
૥	0A94	૮	0AB8
૦	0A95	૯	0AB9
૧	0A96	.	0ABC
૨	0A97	।	0ABE
૩	0A98	૿	0ABF
૪	0A99	ૺ	0AC0
૫	0A9A	ૻ	0AC1
૬	0A9B	ૼ	0AC2
૭	0A9C	૽	0AC3
૮	0A9D	૾	0AC5
૯	0A9E	૿	0AC7
૦	0A9F	ૺ	0AC8
૧	0AA0	ૻ	0AC9
૨	0AA1	ૼ	0ACB

字元	代碼	字元	代碼
ᳵ	0AA2	ᳶ	0ACC
᳷	0AA3	᳸	0ACD
᳹	0AA4	ᳺ	0AD0
᳻	0AA5	᳼	0AE0

## 豪薩文字元集

對於豪薩文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在此 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
k	0199	ᵏ	0253
y	01B4	ᵐ	0257
~	0303		

## 希伯來文字元集

對於希伯來文自訂詞彙，您可以在 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
-	002D	֊	05DD
κ	05D0	ן	05DE

字元	代碼	字元	代碼
ב	05D1	ך	05DF
ג	05D2	ג	05E0
ד	05D3	ס	05E1
ה	05D4	ע	05E2
ו	05D5	ף	05E3
ז	05D6	פ	05E4
ח	05D7	ץ	05E5
ט	05D8	צ	05E6
י	05D9	ק	05E7
ך	05DA	ר	05E8
כ	05DB	ש	05E9
ל	05DC	ת	05EA

## 印地文字元集

對於印地文自訂詞彙，您可以在 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
-	002D	थ	0925
.	002E	द	0926
ॆ	0901	ध	0927
ॆ	0902	न	0928

字元	代碼	字元	代碼
:	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
ऑ	0911	ष	0937
ओ	0913	स	0938
औ	0914	ह	0939
क	0915	ट	093E
ख	0916	डि	093F
ग	0917	ति	0940
घ	0918	उ	0941
ङ	0919	ॆ	0942
च	091A	ॆ	0943
छ	091B	ॆ	0945



字元	代碼	字元	代碼
ज	091C	˘	0947
झ	091D	˙	0948
ञ	091E	ँ	0949
ट	091F	े	094B
ठ	0920	ै	094C
ड	0921	ँ	094D
ढ	0922	ज़	095B
ण	0923	ड़	095C
त	0924	ढ़	095D

Amazon Transcribe 對應以下字元：

字元	對應至
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)

## 匈牙利文字元集

對於匈牙利文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在此欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
á	00E1	ö	00F6
é	00E9	ú	00FA
í	00ED	ü	00FC
ó	00F3	ő	0151
ű	0171		

## 冰島文字元集

對於冰島文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在此欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
á	00E1	ú	00FA
é	00E9	ý	00FD

字元	代碼	字元	代碼
ð	00F0	þ	00FE
í	00ED	æ	00E6
ó	00F3	ö	00F6

## 印尼文字元集

對於印尼文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- A - Z
- '(單引號)
- -(連字號)
- .(句點)

## 義大利文字元集

對於義大利文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- A - Z
- '(單引號)
- -(連字號)
- .(句點)

您也可以使用下列 Unicode 字元：

字元	代碼	字元	代碼
À	00C0	à	00E0
Ä	00C4	ä	00E4

字元	代碼	字元	代碼
Ç	00C7	ç	00E7
È	00C8	è	00E8
É	00C9	é	00E9
Ê	00CA	ê	00EA
Ë	00CB	ë	00EB
Ì	00CC	ì	00EC
Ò	00D2	ò	00F2
Ù	00D9	ù	00F9
Ü	00DC	ü	00FC

## 日文字元集

對於日文自訂詞彙，DisplayAs 欄位支援所有平假名、片假名和漢字字元，以及全形羅馬字大寫字母。

此 Phrase 欄位支援下列檔案中列出的字元：

- [ja-jp-character-set](#)

## 卡拜爾文字元集

對於卡拜爾文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在此 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
ï	00EF	đ	1E0D
č	010D	ħ	1E25
ř	0159	ŗ	1E5B
ǧ	01E7	ş	1E63
ε	025B	ţ	1E6D
γ	0263	ẏ	1E93

## 坎那達文字元集

對於坎那達文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在此 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
ೠ	0C82	ೡ	0CA7
ೢ	0C83	ೣ	0CA8
೤	0C85	೥	0CAA
೦	0C86	೧	0CAB
ೡ	0C87	ೢ	0CAC
ೣ	0C88	೤	0CAD

字元	代碼	字元	代碼
ಉ	0C89	ಋ	0CAE
ಊ	0C8A	ೠ	0CAF
ಋ	0C8B	ೡ	0CB0
ಎ	0C8E	ೢ	0CB2
ಏ	0C8F	ೣ	0CB3
ಐ	0C90	೤	0CB5
ಒ	0C92	೥	0CB6
ಓ	0C93	೦೩	0CB7
ಔ	0C94	೦೪	0CB8
ಕ	0C95	೦೫	0CB9
ಖ	0C96	#	0CBC
ಗ	0C97	#	0CBD
ಘ	0C98	೦೬	0CBE
ಙ	0C99	೦೭	0CBF
ಚ	0C9A	೦೮	0CC0
ಛ	0C9B	೦೯	0CC1
ಜ	0C9C	೦೯	0CC2
ಝ	0C9D	೦೯	0CC3
ಞ	0C9E	೦೯	0CC6
ಟ	0C9F	೦೯	0CC7

字元	代碼	字元	代碼
అ	0CA0	ఆ	0CC8
ఇ	0CA1	ఐ	0CCA
ఈ	0CA2	ఊ	0CCB
ఐ	0CA3	ఋ	0CCC
ఊ	0CA4	ౠ	0CCD
ఋ	0CA5	ౡ	0CD5
ౠ	0CA6	ౢ	0CD6
ఘ	0CE0		

## 哈薩克文字元集

對於哈薩克文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在此 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
т	0442	ы	044B
б	0431	я	044F
о	043E	с	0441
п	043F	h	04BB
ш	0448	д	0434

字元	代碼	字元	代碼
и	0438	р	0440
ч	0447	г	0433
н	043D	ё	0451
қ	049B	й	0439
і	0456	ө	04E9
щ	0449	в	0432
е	0435	э	044D
ә	04D9	ң	04A3
ю	044E	л	043B
з	0437	ф	0444
х	0445	к	043A
ц	0446	у	0443
ү	04AF	ж	0436
м	043C	ғ	0493
ь	044C	а	0430
ъ	044A	ұ	04B1

## 盧安達文字元集

對於盧安達文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)



- .(句點)

您也可以在 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
á	00E1	ó	00F3
â	00E2	ô	00F4
ã	00E3	ú	00FA
ç	00E7	ü	00FC
è	00E8	ā	0101
é	00E9	ē	0113
ê	00EA	ī	012B
ë	00EB	ō	014D
í	00ED	ū	016B
ï	00EF	’	0301

## 韓文字元集

對於韓文自訂詞彙，您可以在 Phrase 欄位中使用諺文音節。如需詳細資訊，請參閱維基百科上的[諺文音節](#)。

## 吉爾吉斯文字元集

對於吉爾吉斯文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
а	0430	лъ	0459
б	0431	њ	045A
в	0432	ћ	045B
г	0433	ќ	045C
д	0434	#	045D
е	0435	ђ	045E
ж	0436	џ	045F
з	0437	ѓ	0491
и	0438	Ѡ	0493
й	0439	ѡ	0497
к	043A	Ѣ	0499
л	043B	ѣ	049B
м	043C	Ѥ	049F
н	043D	Ѧ	04A1
о	043E	ѧ	04A3
п	043F	Ѩ	04A5
р	0440	ѩ	04A9
с	0441	Ѫ	04AB
т	0442	ѫ	04AD

字元	代碼	字元	代碼
у	0443	У	04AF
ф	0444	Ф	04B1
х	0445	Х	04B3
ц	0446	Ц	04B5
ч	0447	Ч	04B7
ш	0448	Һ	04BB
щ	0449	Ӈ	04BD
ъ	044A	ӈ	04BF
ы	044B	#	04CA
ь	044C	ӡ	04D1
э	044D	Ӣ	04D3
ю	044E	ӥ	04D7
я	044F	ӈ	04D9
#	0450	ӑ	04E1
ӓ	0451	ӣ	04E3
ӥ	0452	ӧ	04E7
ӧ	0453	Ӊ	04E9
Ӊ	0454	ӫ	04EF
ӫ	0455	ӭ	04F1
ӭ	0456	Ӯ	04F3

字元	代碼	字元	代碼
ï	0457	#	04F7
j	0458	ÿ	04F9

## 拉脫維亞文字元集

對於拉脫維亞文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在此 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
ā	0101	ķ	0137
č	010D	ļ	013C
ē	0113	ņ	0146
ģ	0123	š	0161
ī	012B	ū	016B
ž	017E		

## 立陶宛文字元集

對於立陶宛文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在此欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
ą	0105	į	012F
č	010D	š	0161
ę	0119	ų	0173
è	0117	ū	016B
ž	017E		

## 盧干達文字元集

對於盧干達文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在此欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
ÿ	00FF	ŋ	014B

## 馬其頓文字元集

對於馬其頓文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在此欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
а	0430	лъ	0459
б	0431	њ	045A
в	0432	ћ	045B
г	0433	ќ	045C
д	0434	#	045D
е	0435	ђ	045E
ж	0436	џ	045F
з	0437	ѓ	0491
и	0438	Ѡ	0493
й	0439	ѡ	0497
к	043A	Ѣ	0499
л	043B	ѣ	049B
м	043C	Ѥ	049F
н	043D	Ѧ	04A1
о	043E	ѧ	04A3
п	043F	Ѩ	04A5
р	0440	Ѱ	04A9
с	0441	ѱ	04AB
т	0442	Ѳ	04AD

字元	代碼	字元	代碼
у	0443	У	04AF
ф	0444	Ф	04B1
х	0445	Х	04B3
ц	0446	Ц	04B5
ч	0447	Ч	04B7
ш	0448	Ш	04BB
щ	0449	Щ	04BD
ъ	044A	Ъ	04BF
ы	044B	Ы	04CA
ь	044C	Ь	04D1
э	044D	Э	04D3
ю	044E	Ю	04D7
я	044F	Я	04D9
#	0450	З	04E1
ë	0451	Й	04E3
ђ	0452	Ö	04E7
í	0453	Ө	04E9
є	0454	ȳ	04EF
s	0455	ÿ	04F1
i	0456	ŷ	04F3

字元	代碼	字元	代碼
ï	0457	#	04F7
j	0458	ÿ	04F9

## 馬來文字元集

對於馬來文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- A - Z
- '(單引號)
- -(連字號)
- .(句點)

## 馬來亞拉姆文字元集

對於馬來亞拉姆文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在此 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
◌	0D02	᳚	0D28
᳚	0D03	᳛	0D2A
᳛	0D05	᳜	0D2B
᳜	0D06	᳝	0D2C



字元	代碼	字元	代碼
ഇ	0D07	ഭ	0D2D
ഇ൬	0D08	മ	0D2E
ഉ	0D09	യ	0D2F
ഉ൬	0D0A	ര	0D30
ഋ	0D0B	റ	0D31
എ	0D0E	ല	0D32
ഏ	0D0F	ള	0D33
ഐ	0D10	ഴ	0D34
ഒ	0D12	വ	0D35
ഓ	0D13	ശ	0D36
ഔ	0D14	ഷ	0D37
ക	0D15	സ	0D38
ഖ	0D16	ഹ	0D39
ഗ	0D17	ഘ	0D3E
ഘ	0D18	ങ	0D3F
ങ	0D19	ച	0D40
ച	0D1A	ഛ	0D41
ഛ	0D1B	ജ	0D42
ജ	0D1C	ഝ	0D43
ഝ	0D1D	ഞ	0D46

字元	代碼	字元	代碼
ḡ	0D1E	ḡ	0D47
ḡ	0D1F	ḡ	0D48
o	0D20	ḡ	0D4A
ḡ	0D21	ḡ	0D4B
ḡ	0D22	ḡ	0D4C
ḡ	0D23	ḡ	0D4D
ḡ	0D24	#	0D7A
ḡ	0D25	#	0D7B
ḡ	0D26	#	0D7C
ḡ	0D27	#	0D7D

## 馬爾他文字元集

對於馬爾他文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以使用下列 Unicode 字元：

字元	代碼	字元	代碼
à	00E0	ù	00F9
è	00E8	ć	010B
ì	00EC	đ	0121

字元	代碼	字元	代碼
ò	00F2	ñ	0127
ž	017C		

## 馬拉地文字元集

對於馬拉地文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在此 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
ॠ	0901	थ	0925
ॡ	0902	द	0926
ः	0903	ध	0927
अ	0905	न	0928
आ	0906	प	092A
इ	0907	फ	092B
ई	0908	ब	092C
उ	0909	भ	092D
ऊ	090A	म	092E
ऋ	090B	य	092F
ॠ	090D	र	0930

字元	代碼	字元	代碼
ए	090F	ल	0932
ऐ	0910	ळ	0933
ऑ	0911	व	0935
ओ	0913	श	0936
औ	0914	ष	0937
क	0915	स	0938
ख	0916	ह	0939
ग	0917	.	093C
घ	0918	।	093E
ङ	0919	ि	093F
च	091A	ी	0940
छ	091B	ु	0941
ज	091C	ू	0942
झ	091D	े	0943
ञ	091E	ै	0945
ट	091F	ो	0947
ठ	0920	्	0948
ड	0921	ँ	0949
ढ	0922	ं	094B
ण	0923	ँ	094C

字元	代碼	字元	代碼
т	0924	、	094D
Ѡ	0950		

## 草原馬里文字元集

對於草原馬里文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在此 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
а	0430	љ	0459
б	0431	њ	045A
в	0432	ћ	045B
г	0433	ќ	045C
д	0434	#	045D
е	0435	ђ	045E
ж	0436	џ	045F
з	0437	ѓ	0491
и	0438	ѣ	0493
й	0439	џ	0497
к	043A	џ	0499

字元	代碼	字元	代碼
л	043B	ќ	049B
м	043C	к	049F
н	043D	ќ	04A1
о	043E	ћ	04A3
п	043F	н	04A5
р	0440	џ	04A9
с	0441	џ	04AB
т	0442	џ	04AD
у	0443	у	04AF
ф	0444	у	04B1
х	0445	х	04B3
ц	0446	ц	04B5
ч	0447	ч	04B7
ш	0448	h	04BB
щ	0449	е	04BD
ъ	044A	е	04BF
ы	044B	#	04CA
ь	044C	ă	04D1
э	044D	ä	04D3
ю	044E	ě	04D7

字元	代碼	字元	代碼
я	044F	ə	04D9
#	0450	з	04E1
ë	0451	й	04E3
ђ	0452	ö	04E7
í	0453	ө	04E9
є	0454	ÿ	04EF
s	0455	ÿ	04F1
i	0456	ÿ	04F3
ï	0457	#	04F7
j	0458	Ы	04F9

## 蒙古文字元集

對於蒙古文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以使用下列 Unicode 字元：

字元	代碼	字元	代碼
a	0430	ль	0459
б	0431	нь	045A
в	0432	ћ	045B

字元	代碼	字元	代碼
г	0433	ќ	045C
д	0434	#	045D
е	0435	ђ	045E
ж	0436	џ	045F
з	0437	г	0491
и	0438	Ғ	0493
й	0439	ж	0497
к	043A	з	0499
л	043B	қ	049B
м	043C	к	049F
н	043D	т	04A1
о	043E	ң	04A3
п	043F	н	04A5
р	0440	Ѡ	04A9
с	0441	с	04AB
т	0442	т	04AD
у	0443	у	04AF
ф	0444	ү	04B1
х	0445	х	04B3
ц	0446	ц	04B5



字元	代碼	字元	代碼
ч	0447	ч	04B7
ш	0448	h	04BB
щ	0449	є	04BD
ъ	044A	є	04BF
ы	044B	#	04CA
ь	044C	ă	04D1
э	044D	ä	04D3
ю	044E	ě	04D7
я	044F	ə	04D9
#	0450	з	04E1
ë	0451	й	04E3
ђ	0452	ö	04E7
í	0453	ө	04E9
є	0454	ÿ	04EF
s	0455	ÿ	04F1
i	0456	ÿ	04F3
ï	0457	#	04F7
j	0458	ÿ	04F9

## 書面挪威語字元集

對於書面挪威語自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
å	00E5	æ	00E6
ø	00F8		

## 歐迪亞文字元集

對於歐迪亞文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
◌̣	0B01	◌̥	0B26
◌̤	0B02	◌̦	0B27
◌̧	0B03	◌̨	0B28
◌̩	0B05	◌̪	0B2A
◌̫	0B06	◌̬	0B2B
◌̭	0B07	◌̮	0B2C
◌̯	0B08	◌̰	0B2D

字元	代碼	字元	代碼
ଉ	0B09	ଠ	0B2E
ଊ	0B0A	ଡ	0B2F
ଋ	0B0B	ର	0B30
ଏ	0B0F	ଲ	0B32
ଏଁ	0B10	ଳ	0B33
ଓ	0B13	ଶ	0B36
ଓଁ	0B14	ଷ	0B37
କ	0B15	ସ	0B38
ଖ	0B16	ହ	0B39
ଗ	0B17	.	0B3C
ଘ	0B18		0B3E
ଙ	0B19	^	0B3F
ଚ	0B1A	୧	0B40
ଛ	0B1B	୨	0B41
ଜ	0B1C	୩	0B42
ଝ	0B1D	୪	0B43
ଞ	0B1E	୫	0B47
ଟ	0B1F	୬	0B48
ଠ	0B20	୬#	0B4B
ଡ	0B21	୬୧	0B4C

字元	代碼	字元	代碼
ا	0B22	ـ	0B4D
ا	0B23	ـ	0B56
ا	0B24	ا	0B5F
ا	0B25	ا	0B60
#	0B71		

## 普什圖文字元集

對於普什圖文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在此 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
آ	0622	و	0648
ا	0623	ﻻ	064A
و	0624	=	064B
س	0626	ﻻ	064C
ل	0627	=	064D
ب	0628	ـ	064E
ت	062A	ﻻ	064F
ث	062B	ـ	0650

字元	代碼	字元	代碼
ج	062C	٣	0651
ح	062D	°	0652
خ	062E	#	0654
د	062F	'	0670
ذ	0630	ٲ	067C
ر	0631	پ	067E
ز	0632	ح	0681
س	0633	خ	0685
ش	0634	چ	0686
ص	0635	ٲ	0689
ض	0636	ر	0693
ط	0637	ر	0696
ظ	0638	ژ	0698
ع	0639	بن	069A
غ	063A	ک	06A9
ف	0641	گ	06AB
ق	0642	گ	06AF
ل	0644	ٲ	06BC
م	0645	ی	06CC
ن	0646	ی	06CD

字元	代碼	字元	代碼
o	0647	ś	06D0

## 波蘭文字元集

對於波蘭文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在此 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
ó	00F3	ł	0142
ą	0105	ń	0144
ć	0107	ś	015B
ę	0119	ź	017A
ż	017C		

## 葡萄牙文字元集

對於葡萄牙文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- A - Z
- '(單引號)
- -(連字號)
- .(句點)

您也可以在 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
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

## 旁遮普文字元集

對於旁遮普文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在此 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
ਅ	0A05	ਧ	0A27
ਆ	0A06	ਠ	0A28
ਇ	0A07	ਪ	0A2A
ਈ	0A08	ਫ	0A2B
ਉ	0A09	ਬ	0A2C
ਊ	0A0A	ਭ	0A2D
ਏ	0A0F	ਮ	0A2E
ਐ	0A10	ਯ	0A2F
ਓ	0A13	ਰ	0A30
ਔ	0A14	ਲ	0A32
ਕ	0A15	ਵ	0A35
ਖ	0A16	ਸ	0A38
ਗ	0A17	ਹ	0A39
ਘ	0A18	.	0A3C



字元	代碼	字元	代碼
ਛ	0A19	ਟ	0A3E
ਚ	0A1A	ਫ	0A3F
ਛ	0A1B	ਫ	0A40
ਜ	0A1C	ੁ	0A41
ੜ	0A1D	ੰ	0A42
ੜ	0A1E	ੱ	0A47
ਟ	0A1F	ੰ	0A48
ਠ	0A20	ੰ	0A4B
ਛ	0A21	ੰ	0A4C
ਛ	0A22	ੰ	0A4D
ਣ	0A23	ੜ	0A5C
ਤ	0A24	ੰ	0A70
ਥ	0A25	ੰ	0A71
ਦ	0A26	ੜ	0A72
ੳ	0A73		

## 羅馬尼亞文字元集

對於羅馬尼亞文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
ă	0103	#	0219
â	00E2	#	021B
î	00EE	ş	015F
ț	0163		

## 俄文字元集

對於俄文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

字元	代碼	字元	代碼
'	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
д	0434	ц	0446
е	0435	ч	0447
ж	0436	ш	0448
з	0437	щ	0449

字元	代碼	字元	代碼
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

## 塞爾維亞文字元集

對於塞爾維亞文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以使用下列 Unicode 字元：

字元	代碼	字元	代碼
ć	0107	ı	0456
č	010D	ï	0457
đ	0111	ј	0458
š	0161	љ	0459
ž	017E	њ	045A
а	0430	ћ	045B

字元	代碼	字元	代碼
б	0431	ќ	045C
в	0432	#	045D
г	0433	ђ	045E
д	0434	џ	045F
е	0435	ѓ	0491
ж	0436	ѣ	0493
з	0437	жѝ	0497
и	0438	џ	0499
й	0439	ќ	049B
к	043A	к	049F
л	043B	ќ	04A1
м	043C	ќ	04A3
н	043D	ќ	04A5
о	043E	ќ	04A9
п	043F	ќ	04AB
р	0440	ќ	04AD
с	0441	ќ	04AF
т	0442	ќ	04B1
у	0443	ќ	04B3
ф	0444	ќ	04B5

字元	代碼	字元	代碼
x	0445	ч	04B7
ц	0446	h	04BB
ч	0447	е	04BD
ш	0448	ё	04BF
щ	0449	#	04CA
ъ	044A	ă	04D1
ы	044B	ä	04D3
ь	044C	ě	04D7
э	044D	ө	04D9
ю	044E	з	04E1
я	044F	й	04E3
#	0450	ö	04E7
ë	0451	ө	04E9
ђ	0452	ÿ	04EF
í	0453	ÿ	04F1
є	0454	ÿ	04F3
s	0455	#	04F7
ÿ	04F9		

## 僧伽羅文字元集

對於僧伽羅文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
#	0D82	#	0DAF
#	0D83	#	0DB0
#	0D85	#	0DB1
#	0D86	#	0DB3
#	0D87	#	0DB4
#	0D88	#	0DB5
#	0D89	#	0DB6
#	0D8A	#	0DB7
#	0D8B	#	0DB8
#	0D8C	#	0DB9
#	0D8D	#	0DBA
#	0D91	#	0DBB
#	0D92	#	0DBD
#	0D93	#	0DC0
#	0D94	#	0DC1
#	0D95	#	0DC2

字元	代碼	字元	代碼
#	0D96	#	0DC3
#	0D9A	#	0DC4
#	0D9B	#	0DC5
#	0D9C	#	0DC6
#	0D9D	#	0DCA
#	0D9E	#	0DCF
#	0D9F	#	0DD0
#	0DA0	#	0DD1
#	0DA1	#	0DD2
#	0DA2	#	0DD3
#	0DA3	#	0DD4
#	0DA4	#	0DD6
#	0DA5	#	0DD8
#	0DA7	#	0DD9
#	0DA8	##	0DDA
#	0DA9	#	0ddb
#	0DAA	##	0DDC
#	0DAB	###	0DDD
#	0DAC	##	0DDE
#	0DAD	#	0DDF

字元	代碼	字元	代碼
#	0DAE	#	0DF2

## 斯洛伐克文字元集

對於斯洛伐克文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在此 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
á	00E1	ň	0148
ä	00E4	ó	00F3
č	010D	ô	00F4
d'	010F	í	0155
é	00E9	š	0161
í	00ED	t'	0165
í	013A	ú	00FA
ř	013E	ý	00FD
ž	017E		

## 斯洛維尼亞文字元集

對於斯洛維尼亞文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：



- a - z
- -(連字號)
- .(句點)

您也可以在此欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
č	010D	š	0161
ž	017E		

## 索馬利亞文字元集

對於索馬利亞文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在此欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
s	0073	d	0064
t	0074	a	0061
a	0061	r	0072
n	006E	d	0064

## 西班牙文字元集

對於西班牙文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- A - Z
- '(單引號)
- -(連字號)
- .(句點)

您也可以在 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
Á	00C1	á	00E1
É	00C9	é	00E9
Í	00CD	í	00ED
Ó	00D3	ó	0XF3
Ú	00DA	ú	00FA
Ñ	00D1	ñ	0XF1
ü	00FC		

## 巽他文字元集

對於巽他文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
s	0073	d	0064
t	0074	a	0061
a	0061	r	0072
n	006E	d	0064

## 史瓦希里文字元集

對於史瓦希里文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在此欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
s	0073	d	0064
t	0074	a	0061
a	0061	r	0072
n	006E	d	0064

## 瑞典文字元集

對於瑞典文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- A - Z

- '(單引號)
- -(連字號)
- .(句點)

您也可以在此欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
Ä	00C4	ä	00E4
Å	00C5	å	00E5
Ö	00D6	ö	00F6

## 他加祿語/菲律賓文字元集

對於他加祿語/菲律賓文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在此欄位中使用下列 Unicode 字元：

字元	代碼		
ñ	00F1		

## 坦米爾文字元集

對於坦米爾文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

字元	代碼	字元	代碼
அ	0B85	ஈ	0BB0

字元	代碼	字元	代碼
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	ஐ	0B9C
ஐ	0B90	#	0BB6
ஒ	0B92	ஷ	0BB7
ஓ	0B93	ஸ	0BB8
ஔ	0B94	ஹ	0BB9
ஃ	0B83	.	0BCD
க	0B95	ஈ	0BBB
ங	0B99	ீ	0BBF
ச	0B9A	ஊ	0BC0
ஞ	0B9E	஋	0BC1
ட	0B9F	஌	0BC2
ண	0BA3	஍	0BC6
த	0BA4	எ	0BC7
ந	0BA8	ஏ	0BC8

字元	代碼	字元	代碼
Ɑ	0BAA	ᄀ	0BCA
ᄁ	0BAE	ᄂ	0BCB
ᄃ	0BAF	ᄄ	0BCC

## 韃靼語字元集

對於韃靼語自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以使用下列 Unicode 字元：

字元	代碼	字元	代碼
а	0430	ӓ	0459
б	0431	ӕ	045A
в	0432	ӗ	045B
г	0433	ӑ	045C
д	0434	#	045D
е	0435	ӧ	045E
ж	0436	ӕ	045F
з	0437	ӑ	0491
и	0438	ӕ	0493
й	0439	ӗ	0497

字元	代碼	字元	代碼
к	043A	ꝑ	0499
л	043B	Ꝓ	049B
м	043C	ꝓ	049F
н	043D	Ꝕ	04A1
о	043E	ꝕ	04A3
п	043F	Ꝗ	04A5
р	0440	ꝗ	04A9
с	0441	Ꝙ	04AB
т	0442	ꝙ	04AD
у	0443	Ꝛ	04AF
ф	0444	ꝛ	04B1
х	0445	Ꝝ	04B3
ц	0446	ꝝ	04B5
ч	0447	Ꝟ	04B7
ш	0448	h	04BB
щ	0449	є	04BD
ъ	044A	ё	04BF
ы	044B	#	04CA
ь	044C	ă	04D1
э	044D	ä	04D3

字元	代碼	字元	代碼
ю	044E	ě	04D7
я	044F	ə	04D9
#	0450	з	04E1
ë	0451	й	04E3
ђ	0452	ö	04E7
í	0453	ө	04E9
є	0454	ÿ	04EF
s	0455	ÿ	04F1
i	0456	ý	04F3
ï	0457	#	04F7
j	0458	Ы	04F9

## 特拉古文字元集

對於特拉古文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

字元	代碼	字元	代碼
-	002D	ѣ	0C24
с	0C01	ѡ	0C25
о	0C02	Ѣ	0C26
ѣ	0C03	ѣ	0C27
ѣ	0C05	ѣ	0C28



字元	代碼	字元	代碼
ఆ	0C06	ప	0C2A
ఇ	0C07	భ	0C2B
ఈ	0C08	బ	0C2C
ఉ	0C09	భ	0C2D
ఊ	0C0A	మ	0C2E
ఋ	0C0B	య	0C2F
ఠ	0C30	ఎ	0C0E
ఐ	0C31	ఏ	0C0F
అ	0C32	ఐ	0C10
ఆ	0C33	ఒ	0C12
ప	0C35	ఓ	0C13
శ	0C36	ఔ	0C14
ష	0C37	క	0C15
స	0C38	ఖ	0C16
హ	0C39	గ	0C17
౧	0C3E	ఘ	0C18
౨	0C3F	ఙ	0C19
౩	0C40	చ	0C1A
౪	0C41	ఛ	0C1B
౫	0C42	జ	0C1C

字元	代碼	字元	代碼
จ	0C43	ช	0C1D
ฉ	0C44	ฌ	0C1E
ส	0C47	ฮ	0C1F
ศ	0C48	ศ	0C20
ษ	0C4A	ษ	0C21
ฬ	0C4B	ฬ	0C22
ฬ	0C4C	ณ	0C23
เ	0C4D		

## 泰文字元集

對於泰文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- -(連字號)
- .(句點)

您也可以在此欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
ก	0E01	ล	0E25
ข	0E02	ฬ	0E26
ค	0E03	ว	0E27
ด	0E04	ศ	0E28
ต	0E05	ษ	0E29

字元	代碼	字元	代碼
ฃ	0E06	ส	0E2A
ง	0E07	ห	0E2B
จ	0E08	ฬ	0E2C
ฉ	0E09	อ	0E2D
ช	0E0A	ฮ	0E2E
ซ	0E0B	ฯ	0E2F
ฌ	0E0C	๕	0E30
ญ	0E0D	๖	0E31
ฎ	0E0E	๗	0E32
ฏ	0E0F	๘	0E34
ฐ	0E10	๙	0E35
ฑ	0E11	๐	0E36
ฒ	0E12	๑	0E37
ณ	0E13	๒	0E38
ด	0E14	๓	0E39
ต	0E15	๔	0E3A
ถ	0E16	๕	0E40
ท	0E17	๖	0E41
ธ	0E18	๗	0E42
น	0E19	๘	0E43

字元	代碼	字元	代碼
ı	0E1A	ı	0E44
ı	0E1B	ı	0E45
ı	0E1C	ı	0E46
ı	0E1D	ı	0E47
ı	0E1E	ı	0E48
ı	0E1F	ı	0E49
ı	0E20	ı	0E4A
ı	0E21	ı	0E4B
ı	0E22	ı	0E4C
ı	0E23	ı	0E4D
ı	0E24		

## 土耳其文字元集

對於土耳其文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- A - Z
- '(單引號)
- -(連字號)
- .(句點)

您也可以使用下列 Unicode 字元：

字元	代碼	字元	代碼
Ç	00C7	ö	00F6
Ö	00D6	û	00FB
Ü	00DC	ü	00FC
â	00E2	Ĝ	011E
ä	00E4	ğ	011F
ç	00E7	ı	0130
è	00E8	İ	0131
é	00E9	Ş	015E
ê	00EA	ş	015F
í	00ED	š	0161
î	00EE	ž	017E
ó	00F3		

## 烏克蘭文字元集

對於烏克蘭文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在此 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
а	0430	р	0440
б	0431	с	0441
в	0432	т	0442
г	0433	у	0443
д	0434	ф	0444
е	0435	х	0445
ж	0436	ц	0446
з	0437	ч	0447
и	0438	ш	0448
й	0439	щ	0449
к	043A	ь	044C
л	043B	ю	044E
м	043C	я	044F
н	043D	ё	0454
о	043E	і	0456
п	043F	ї	0457
ғ	0491		

## 維吾爾語字元集

對於維吾爾語自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z

- -(連字號)
- .(句點)

您也可以在 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
#	0611	و	0648
#	0613	س	0649
#	0614	ش	064A
ء	0621	=	064B
آ	0622	ا	064C
أ	0623	=	064D
ؤ	0624	ـ	064E
إ	0625	ا	064F
ئ	0626	ـ	0650
ا	0627	ا	0651
ب	0628	و	0652
ة	0629	#	0653
ت	062A	#	0654
ث	062B	#	0657
ج	062C	ـ	0670
ح	062D	ط	0679
خ	062E	ن	067A

字元	代碼	字元	代碼
د	062F	پ	067B
ذ	0630	ت	067C
ر	0631	ث	067D
ز	0632	پ	067E
س	0633	ث	067F
ش	0634	پ	0680
ص	0635	ح	0681
ض	0636	ج	0683
ط	0637	ج	0684
ظ	0638	ح	0685
ع	0639	ج	0686
غ	063A	چ	0687
-	0640	ڈ	0688
ف	0641	د	0689
ق	0642	د	068A
ك	0643	ذ	068C
ل	0644	د	068D
م	0645	ذ	068F
ن	0646	ژ	0691
ه	0647	ر	0693



字元	代碼	字元	代碼
ﺝ	0695		

## 烏茲別克文字元集

對於烏茲別克文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在此 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
т	0442	я	044F
б	0431	с	0441
о	043E	ҳ	04B3
п	043F	д	0434
ш	0448	р	0440
и	0438	ў	045E
ч	0447	г	0433
н	043D	ё	0451
қ	049B	й	0439
е	0435	в	0432
ю	044E	э	044D
з	0437	л	043B

字元	代碼	字元	代碼
x	0445	φ	0444
ц	0446	κ	043A
м	043C	у	0443
ь	044C	ж	0436
ъ	044A	ƒ	0493
a	0430		

## 越南字元集

Amazon Transcribe 使用數字，代表越南文中的六個音調。下表顯示如何為單字「ma」顯示音調標記。

音調名稱	音調標記	音調數字
ngang	ma	ma1
sắc	má	ma2
huyền	mà	ma3
hỏi	mả	ma4
ngã	mã	ma5
nặng	mạ	ma6

對於越南文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- A - Z
- '(單引號)

- -(連字號)
- .(句點)
- &(與符)
- ;(分號)
- \_(底線)

您也可以在此 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
à	00E0	À	00C0
á	00E1	Á	00C1
â	00E2	Â	00C2
ã	00E3	Ã	00C3
è	00E8	È	00C8
é	00E9	É	00C9
ê	00EA	Ê	00CA
ì	00EC	Ì	00CC
í	00ED	Í	00CD
ò	00F2	Ò	00D2
ó	00F3	Ó	00D3
ô	00F4	Ô	00D4
õ	00F5	Õ	00D5
ù	00F9	Ù	00D9
ú	00FA	Ú	00DA

字元	代碼	字元	代碼
ý	00FD	Ý	00DD
ă	0103	Ă	0102
đ	0111	Đ	0110
ĩ	0129	Ĩ	0128
ũ	0169	Ũ	0168
ơ	01A1	Ơ	01A0
ư	01B0	Ư	01AF
ạ	1EA1	Ạ	1EA0
â	1EA3	Â	1EA2
ã	1EA5	Ã	1EA4
ä	1EA7	Ä	1EA6
å	1EA9	Å	1EA8
ẵ	1EAB	Ẵ	1EAA
â	1EAD	Â	1EAC
ă̂	1EAF	Ă̂	1EAE
ă̄	1EB1	Ă̄	1EB0
ă̆	1EB3	Ă̆	1EB2
ă̈	1EB5	Ă̈	1EB4
ă̊	1EB7	Ă̊	1EB6
ē	1EB9	Ē	1EB8

字元	代碼	字元	代碼
è	1EBB	È	1EBA
ě	1EBD	ě	1EBC
é	1EBF	É	1EBE
è	1EC1	È	1EC0
ě	1EC3	Ě	1EC2
ě	1EC5	Ě	1EC4
ê	1EC7	Ê	1EC6
ï	1EC9	Ï	1EC8
ì	1ECB	Ì	1ECA
ọ	1ECD	Ọ	1ECC
ỏ	1ECF	Ỏ	1ECE
ố	1ED1	Ố	1ED0
ồ	1ED3	Ồ	1ED2
ỗ	1ED5	Ỗ	1ED4
ỗ	1ED7	Ỗ	1ED6
ộ	1ED9	Ộ	1ED8
ớ	1EDB	Ớ	1EDA
ờ	1EDD	Ờ	1EDC
ở	1EDF	Ở	1EDE
ỡ	1EE1	Ỡ	1EE0

字元	代碼	字元	代碼
ø	1EE3	Ø	1EE2
ұ	1EE5	Ҫ	1EE4
ů	1EE7	Ǫ	1EE6
ú	1EE9	Ў	1EE8
ù	1EEB	Ǯ	1EEA
ű	1EED	Ǫ̂	1EEC
ű	1EEF	Ǫ̃	1EEE
ұ	1EF1	Ҫ	1EF0
ỳ	1EF3	Ỳ	1EF2
ʏ	1EF5	Ỳ	1EF4
ÿ	1EF7	Ỳ̂	1EF6
ÿ	1EF9	Ỳ̃	1EF8

## 威爾斯語字元集

對於威爾斯語自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在此 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
à	00E0	ò	00F2

字元	代碼	字元	代碼
á	00E1	ó	00F3
â	00E2	ô	00F4
ä	00E4	ö	00F6
è	00E8	ù	00F9
é	00E9	ú	00FA
ê	00EA	û	00FB
ë	00EB	ü	00FC
ì	00EC	ý	00FD
í	00ED	ÿ	00FF
î	00EE	ŵ	0175
ï	00EF	ÿ	0177
ỳ	1EF3		

## 沃洛夫文字元集

對於沃洛夫文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在此 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
à	00E0	ê	00EA

字元	代碼	字元	代碼
ã	00E3	ë	00EB
ç	00E7	ñ	00F1
è	00E8	ó	00F3
é	00E9	ô	00F4
η	014B		

## 祖魯文字元集

對於祖魯文自訂詞彙，您可以在 Phrase 欄位中使用下列字元：

- a - z
- -(連字號)
- .(句點)

您也可以在此 Phrase 欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
s	0073	d	0064
t	0074	a	0061
a	0061	r	0072
n	006E	d	0064



# Amazon Transcribe 的運作方式

Amazon Transcribe 使用機器學習模型將語音轉換為文字。

除了轉錄文字之外，謄本還包含轉錄內容的相關資料，包括每個字詞或標點符號的置信度分數和時間戳記。若要查看輸出範例，請參閱「[資料輸入和輸出](#)」一節。如需可套用至轉錄功能的完整清單，請參閱[功能摘要](#)。


轉錄方法可分為兩個主要類別：

- Batch 轉錄：Transcribe 已上傳至 Amazon S3 值區的媒體檔案。您可以使用 [AWS CLI](#)、[AWS Management Console](#)、和各種 [AWS SDK](#) 進行批次轉錄。
- 串流轉錄：即時 Transcribe 媒體串流。您可以使 [AWS Management Console](#) 用 [HTTP/2](#) 和各種 [AWS SDK](#) 進行串流轉錄。[WebSockets](#)

請注意，批次和串流轉錄的功能和語言支援會有所不同。如需詳細資訊，請參閱 [Amazon Transcribe features](#) 和 [支援的語言](#)。

主題

- [資料輸入和輸出](#)
- [轉錄數字和標點符號](#)

 協助您開始使用的 API 作業

Batch：[StartTranscriptionJob](#)

流媒體：[StartStreamTranscription](#), [StartStreamTranscriptionWebSocket](#)

## 資料輸入和輸出

Amazon Transcribe 將音頻數據作為 Amazon S3 存儲桶或媒體流中的媒體文件，並將其轉換為文本數據。

如果您要轉錄儲存在 Amazon S3 值區中的媒體檔案，表示您正在執行批次轉錄。如果您要轉錄媒體串流，表示您正在執行串流轉錄。這兩個流程有不同的規則和要求。

**Job 佇列** 如果您不需要同時處理所有轉錄工作，則可以使用批次轉錄。這允許 Amazon Transcribe 跟踪您的轉錄作業，並在可用插槽時進行處理。

### Note

Amazon Transcribe 可能會暫時儲存您的內容，以持續改善其分析模型的品質。若要進一步了解，請參閱 [Amazon Transcribe 常見問答集](#)。若要要求刪除可能儲存的內容 Amazon Transcribe，請使用開啟案例 [AWS Support](#)。

## 主題

- [媒體格式](#)
- [音訊頻道](#)
- [取樣率](#)
- [輸出](#)

## 媒體格式

批次轉錄和串流轉錄的支援媒體類型會有所不同，但兩者都建議使用無失真格式。如需詳細資訊，請參閱下表：

	批次	串流
支援的格式	<ul style="list-style-type: none"> <li>• AMR</li> <li>• 弗拉克</li> <li>• M4A</li> <li>• MP3</li> <li>• MP4</li> <li>• 奧格</li> <li>• WebM</li> <li>• WAV</li> </ul>	<ul style="list-style-type: none"> <li>• 弗拉克</li> <li>• 奧格作品</li> <li>• PCM 編碼</li> </ul>
推薦格式	<ul style="list-style-type: none"> <li>• 弗拉克</li> <li>• 具備 PCM 16 位元編碼功能的 WAV 檔案</li> </ul>	<ul style="list-style-type: none"> <li>• 弗拉克</li> <li>• PCM 簽署了 16 位小端音頻 ( 請注意，這不包括 WAV )</li> </ul>

為獲得最佳結果，請使用無損格式，例如 FLAC 或具有 PCM 16 位元編碼的 WAV。

### Note

並非所有語言都支援串流轉錄。如需詳細資訊，請參閱[支援語言表格](#)中的「資料輸入」欄。

## 音訊頻道

Amazon Transcribe 支援單通道和雙通道媒體。目前不支援超過兩個頻道的媒體。

如果您的音訊在一個頻道上包含多個揚聲器，而您想要在轉錄輸出中對每個揚聲器進行分區和標記，則可以使用[揚聲器分區 \(日記\)](#)。

如果您的音訊包含兩個不同頻道的語音，您可以使用[頻道識別](#)功能，在逐字稿中分別轉錄每個頻道。

這兩個選項都會產生一個成績單檔案。

### Note

如果您未啟用「[喇叭分割](#)」或「[頻道識別](#)」，您的成績單文字會以一個連續區段的形式提供。

## 取樣率

使用批次轉錄工作時，您可以選擇提供取樣率，但此參數是選擇性的。如果您將其包含在請求中，請確定您提供的值與音訊中的實際取樣率相符。如果您提供的取樣率與音訊不相符，您的工作可能會失敗。

使用串流轉錄時，您必須在請求中包含取樣率。與批次轉錄工作一樣，請確定您提供的值與音訊中的實際取樣率相符。

低保真度音訊 (例如電話錄音) 的取樣率通常使用 8,000 Hz。對於高保真音頻，Amazon Transcribe 支持 16,000 赫茲和 48,000 赫茲之間的數值。

## 輸出

字符轉錄輸出為 JSON 格式。成績單的第一部分以段落形式包含成績單本身，後面接著每個單字和標點符號的其他資料。所提供的資料取決於您要求中包含的功能。您的成績單至少會包含每個字詞的開始時間、結束時間和置信度分數。[以下部分](#)顯示不包含任何其他選項或功能的基本轉錄請求的範例輸出。

所有批次記錄都儲存在Amazon S3值區中。您可以選擇將成績單儲存在自己的Amazon S3值區中，或Amazon Transcribe使用安全的預設值區。若要深入瞭解如何建立和使用Amazon S3值區，請參閱[使用值區](#)。

如果您希望將成績單儲存在您擁有的儲存Amazon S3貯體中，請在轉錄請求中指定值區的URI。在開始批次轉錄工作之前，請確定您已授予此儲存貯體的Amazon Transcribe寫入權限。如果您指定自己的值區，您的成績單會保留在該值區中，直到您將其移除為止。

如果您未指定Amazon S3值區，請Amazon Transcribe使用安全的服務管理值區，並提供您可用來下載成績單的暫時URI。請注意，臨時URI的有效期限為15分鐘。如果您在使用提供的URI時發AccessDenied生錯誤，請提出GetTranscriptionJob要求以取得成績單的新暫時URI。

如果您選擇預設值區，則會在工作到期時(90天)刪除您的成績單。如果您想讓成績單超過此到期日，您必須下載它。

串流文字記錄會透過您在串流中使用的相同方法傳回。

#### Tip

如果您想要將JSON輸出轉換為Word格式的文turn-by-turn字記錄，請參閱此[GitHub 範例 \(適用於Python3\)](#)。此指令碼可搭配通話後分析記錄和啟用日記的標準批次成績單搭配使用。

## 範例輸出

謄本以段落形式提供完整的轉錄，後面接著word-for-word劃分，提供每個字詞和標點符號的資料。這包括開始時間、結束時間、可信度分數和類型(pronunciation或punctuation)。

下列範例來自不包含任何[其他功能](#)的簡單批次轉錄工作。透過您套用至轉錄要求的每個其他功能，您都可以在成績單輸出檔案中取得額外的資料。

基本批次成績單包含兩個主要部分：

1. transcripts：在一個文本塊中包含整個成績單。
2. items：包含部分中每個單詞和標點符號的信transcripts息。

您在轉錄請求中包含的每個其他功能都會在您的成績單中產生額外的資訊。

```
{
  "jobName": "my-first-transcription-job",
  "accountId": "111122223333",
```

```
"results": {
  "transcripts": [
    {
      "transcript": "Welcome to Amazon Transcribe."
    }
  ],
  "items": [
    {
      "start_time": "0.64",
      "end_time": "1.09",
      "alternatives": [
        {
          "confidence": "1.0",
          "content": "Welcome"
        }
      ],
      "type": "pronunciation"
    },
    {
      "start_time": "1.09",
      "end_time": "1.21",
      "alternatives": [
        {
          "confidence": "1.0",
          "content": "to"
        }
      ],
      "type": "pronunciation"
    },
    {
      "start_time": "1.21",
      "end_time": "1.74",
      "alternatives": [
        {
          "confidence": "1.0",
          "content": "Amazon"
        }
      ],
      "type": "pronunciation"
    },
    {
      "start_time": "1.74",
      "end_time": "2.56",
      "alternatives": [
```

```
        {
            "confidence": "1.0",
            "content": "Transcribe"
        }
    ],
    "type": "pronunciation"
},
{
    "alternatives": [
        {
            "confidence": "0.0",
            "content": "."
        }
    ],
    "type": "punctuation"
}
]
},
"status": "COMPLETED"
}
```

## 轉錄數字和標點符號

Amazon Transcribe會自動將標點符號新增至所有支援的語言，並針對在其書寫系統中使用大小寫區分的語言，適當地將文字大寫。

對於大多數語言，數字會轉錄為其單詞形式。但是，如果您的媒體使用英文或德文，Amazon Transcribe則會根據使用數字的上下文而有所不同。

例如，如果發言者說「」Meet me at eight-thirty AM on June first at one-hundred Main Street with three-dollars-and-fifty-cents and one-point-five chocolate bars，則會將其轉錄為：

- 英語和德語方言：Meet me at 8:30 a.m. on June 1st at 100 Main Street with \$3.50 and 1.5 chocolate bars
- 所有其他語言：Meet me at eight thirty a m on June first at one hundred Main Street with three dollars and fifty cents and one point five chocolate bars

要查看與英語和德語口語數字相關的所有規則，請參閱下表。

規則	英語方言 ( 輸入音頻 → 輸出文本 )	德語方言 ( 輸入音頻 → 輸出文本 )
將大於 10 的基數轉換為數字。	<ul style="list-style-type: none"> <li>• "Fifty five" → 55</li> <li>• "a hundred" → 100</li> <li>• "One thousand and thirty one" → 1031</li> <li>• "One hundred twenty-three million four hundred fifty six thousand seven hundred eight nine" → 123,456,789</li> </ul>	<ul style="list-style-type: none"> <li>• "fünfundfünfzig" → 55</li> <li>• "vier tausend sechs hundert einundachtzig" → 4681</li> <li>• "eine Sache" → "eine Sache"</li> </ul>
當「百萬」或「億」後面沒有數字時，將基數後面接著「百萬」或「億」的數字轉換為數字後接著字詞。	<ul style="list-style-type: none"> <li>• "one hundred million" → 100 million</li> <li>• "one billion" → 1 billion</li> <li>• "two point three million" → 2.3 million</li> </ul>	<ul style="list-style-type: none"> <li>• "zehn Millionen Menschen" → 10 Millionen Menschen</li> <li>• "zehn Millionen fünf hundert tausend" → 10.500.000</li> </ul>
將大於 10 的序數轉換為數字。	<ul style="list-style-type: none"> <li>• "Forty third" → 43rd</li> <li>• "twenty sixth avenue" → 26th avenue</li> </ul>	<ul style="list-style-type: none"> <li>• "dreiundzwanzigste" → 23</li> <li>• "vierzigster" → 40</li> <li>• "ich war Erster" → "ich war Erster"</li> </ul>
將分數轉換成數字形式。	<ul style="list-style-type: none"> <li>• "a quarter" → 1/4</li> <li>• "three sixteenths" → 3/16</li> <li>• "a half" → 1/2</li> <li>• "a hundredth" → 1/100</li> </ul>	分數不會轉換成數值格式。  <ul style="list-style-type: none"> <li>• "ein Drittel" → "ein Drittel"</li> </ul>
如果一行中有多個，則將小於 10 的數字轉換為數字。	<ul style="list-style-type: none"> <li>• "three four five" → 345</li> <li>• "My phone number is four two five five five five one two one two" → My phone number is 4255551212</li> </ul>	<ul style="list-style-type: none"> <li>• "eins zwei drei" → 123</li> <li>• "plus vier neun zwei vier eins" → +49241</li> </ul>

規則	英語方言 ( 輸入音頻 → 輸出文本 )	德語方言 ( 輸入音頻 → 輸出文本 )
單詞「點」或「點」顯示為十進制。	<ul style="list-style-type: none"> <li>• "three hundred and three dot five" → 303.5</li> <li>• "three point twenty three" → 3.23</li> <li>• "zero point four" → 0.4</li> <li>• "point three" → 0.3</li> </ul>	小數由「,」表示。 <ul style="list-style-type: none"> <li>• "zweiundzwanzig komma drei" → 22,3</li> </ul>
將數字後的「百分比」轉換成百分比符號 (%)。	<ul style="list-style-type: none"> <li>• "twenty three percent" → 23%</li> <li>• "twenty three point four five percent" → 23.45%</li> </ul>	<ul style="list-style-type: none"> <li>• "fünf Prozent Hürde" → 5% Hürde</li> <li>• "dreiundzwanzig komma vier Prozent" → 23,4%</li> </ul>



規則	英語方言 ( 輸入音頻 → 輸出文本 )	德語方言 ( 輸入音頻 → 輸出文本 )
將貨幣單詞轉換為符號。	<p>將數字之後的單詞「美元」，「US 美元」，「澳元」，「AUD」或「USD」轉換為數字之前的美元符號 ( \$ )。</p> <ul style="list-style-type: none"> <li>• "one dollar and fifteen cents" → \$1.15</li> <li>• "twenty three USD" → \$23</li> <li>• "twenty three Australian dollars" → \$23</li> </ul> <p>將數字後面的字詞「鎊」、 「英鎊」或「GDB」轉換為數字前的英鎊符號 ( £ )。</p> <ul style="list-style-type: none"> <li>• "twenty three pounds" → £23</li> <li>• "I have two thousand pounds" → I have £2,000</li> <li>• "five pounds thirty three pence" → £5.33</li> </ul> <p>將數字後的字詞「盧比」、 「印度盧比」或「INR」轉換為數字前的盧比符號 ( # )。</p> <ul style="list-style-type: none"> <li>• "twenty three rupees" → #23</li> <li>• "fifty rupees thirty paise" → #50.30</li> </ul>	<p>將單詞「歐元」轉換為歐元符號。</p> <ul style="list-style-type: none"> <li>• "ein euro" → 1 €</li> <li>• "ein Euro vierzig" → 1,40 €</li> <li>• "ein Euro vierzig Cent" → 1,40 €</li> </ul>

規則	英語方言 ( 輸入音頻 → 輸出文本 )	德語方言 ( 輸入音頻 → 輸出文本 )
將時間轉換成數字。	<ul style="list-style-type: none"> <li>• "seven a m eastern standard time" → 7 a.m. eastern standard time</li> <li>• "twelve thirty p m" → 12:30 p.m.</li> </ul>	<ul style="list-style-type: none"> <li>• "vierzehn Uhr fünfzehn" → 14:15 Uhr</li> </ul>
轉換日期成為數字。	<ul style="list-style-type: none"> <li>• "May fifth twenty twelve" → May 5th 2012</li> <li>• "May five twenty twelve" → May 5 2012</li> <li>• "five May twenty twelve" → 5 May 2012</li> </ul>	<ul style="list-style-type: none"> <li>• "dritter Dezember neunzehn hundert sechundfünfzig" → 3. Dezember 1956</li> </ul>
用單詞「to」分隔數字的跨度。	<ul style="list-style-type: none"> <li>• "twenty three to thirty seven" → 23 to 37</li> </ul>	不適用
年份表示為四位數；這僅在 20、21 世紀和 22 世紀的年份內有效。	<ul style="list-style-type: none"> <li>• "nineteen sixty two" → 1962</li> <li>• 「the year is twenty twelve 「 → 今年是2012</li> <li>• "twenty nineteen" → 2019</li> <li>• "twenty one thirty" → 2130</li> </ul>	不適用
顯示斜線和破折號。	<ul style="list-style-type: none"> <li>• "fifty-five dash thirteen" → 55-13</li> </ul> <p>不顯示斜線。</p> <ul style="list-style-type: none"> <li>• "fifty-five slash thirteen" → 55 slash 13</li> </ul>	<ul style="list-style-type: none"> <li>• "fünfundfünfzig Schrägstrich dreizehn" → 55/13</li> <li>• "fünfundfünfzig Strich dreizehn" → 55-13</li> </ul>

規則	英語方言 ( 輸入音頻 → 輸出文本 )	德語方言 ( 輸入音頻 → 輸出文本 )
顯示編號段落。	編號段落不會使用段落符號 (§) 顯示。  • "paragraph seventeen" → paragraph 17	• "Paragraf siebzehn" → § 17

# Amazon Transcribe 入門

在建立轉錄之前，您必須先行下列幾項先決條件：

- [註冊AWS 帳戶](#)
- [安裝AWS CLI和 SDK](#) ( 如果您使用AWS Management Console的是轉錄，則可以略過此步驟 )
- [設定IAM認證](#)
- [設定Amazon S3值區](#)
- [建立IAM策略](#)

一旦您完成這些先決條件，您即可進行轉錄。從以下列表中選擇您偏好的轉錄方法以開始使用。

- [AWS CLI](#)
- [AWS Management Console](#)
- [AWS 開發套件](#)
- [HTTP](#)
- [WebSockets](#)

## Tip

如果您不熟悉Amazon Transcribe或想要探索我們的功能，建議您使用 [AWS Management Console](#)。如果您想使用計算機麥克風開始流，這也是最簡單的選擇。

由於使用 HTTP/2 進行串流，而 WebSockets 且比其他轉錄方法複雜，因此建議您在開始使用這些方法之前先檢閱[設定串流轉錄](#)本節。請注意，我們強烈建議您使用 SDK 進行串流轉錄。

## 註冊AWS 帳戶

您可以註冊免費方案帳戶或付費帳戶。這兩個選項都使您可以訪問所有內容AWS 服務。免費方案有試用期，您可以在此期間探索AWS 服務並估計使用量。試用期到期後，您可以遷移到付費帳戶。費用是 pay-as-you-use 基於計算的；如需詳細資訊，請參閱[Amazon Transcribe定價](#)。

**i** Tip

設置帳戶時，請記下您的AWS帳戶ID，因為您需要它來創建IAM實體。

## 安裝AWS CLI和軟體開發套件

若要使用Amazon Transcribe API，您必須先安裝AWS CLI。目前AWS CLI是版本 2。您可以在[AWS Command Line Interface使用者指南](#)中找到Linux、Mac、視窗和泊塢視窗的安裝說明。

AWS CLI安裝完成後，您必須將其[配置](#)為您的安全憑據和AWS區域。

如果您想要搭配SDK使Amazon Transcribe用，請選取您偏好的語言以取得安裝說明：

- [.NET](#)
- [C++](#)
- [Go](#)
- [爪哇](#)
- [JavaScript](#)
- [PHP](#)
- [AWS SDK for Python \(Boto3\)](#)(批次轉錄)
- [蟒蛇](#) (流式轉錄)
- <https://docs.aws.amazon.com/sdk-for-ruby/v3/developer-guide/setup-install.html>
- [鏽](#) ( 批量轉錄 )
- [鏽](#) ( 流轉錄 )

## 設定IAM認證

如果是建立AWS帳戶，您會先有一個登入身分，可以完整存取帳戶中所有AWS服務和資源。此身分稱為AWS帳戶根使用者，是藉由您用來建立帳戶的電子郵件地址和密碼以登入並存取。

強烈建議您不要以根使用者處理日常作業。保護您的根使用者憑證，並將其用來執行只能由根使用者執行的任務。

最佳作法是要求使用者 (包括需要管理員存取權的使用者) 搭配身分提供者使用聯合功能，使用暫時性憑證來存取AWS服務。

聯合身分是使用透過身分來源提供的憑證來源存取AWS服務。聯合身分存取 AWS 帳戶時，會擔任角色，並由角色提供暫時性憑證。

對於集中式存取管理，我們建議您使用 [AWS IAM Identity Center](#)。您可以在中建立使用者和群組 IAM Identity Center。也可以連線並同步到自己身分來源中的一組使用者和群組，以便在您的所有 AWS 帳戶和應用程式中使用。如需詳細資訊，請參閱的 [Identity and Access Management Amazon Transcribe](#)。

若要深入了解IAM最佳做法，請參閱[中的安全性最佳做法IAM](#)。

## 建立Amazon S3值區

Amazon S3是安全的物件儲存服務。Amazon S3將您的文件（稱為對象）存儲在容器中（稱為存儲桶）。

若要執行批次轉錄，您必須先將媒體檔案上傳至Amazon S3值區。如果您沒有為轉錄輸出指定Amazon S3值區，請將文字稿Amazon Transcribe放置在暫時AWS受管Amazon S3儲存貯體中。AWS受管理儲存貯體中的轉錄輸出會在 90 天後自動刪除。

了解如何[建立您的第一個 S3 儲存貯體](#)，並[將物件上傳到儲存貯體](#)。

## 建立 IAM 政策

若要管理中的存取權限AWS，您必須建立政策，然後將其附加到IAM身分（使用者、群組或角色）或AWS資源。原則會定義其所附加之實體的權限。例如，如果您已將政策附加到授予其存取權的角色，則角色只能存取位於儲存Amazon S3貯體中的媒體檔案。如果您想進一步限制該角色，可以改為限制該角色對Amazon S3值區內特定檔案的存取權。

若要深入了解如何使用AWS政策，請參閱：

- [中的策略和權限IAM](#)
- [建立IAM策略](#)
- [如何與 IAM Amazon Transcribe 搭配使用](#)

如需可搭配使用的原則範例Amazon Transcribe，請參閱[Amazon Transcribe 身分型政策範例](#)。如果您要產生自訂原則，請考慮使用[AWS原則產生器](#)。

您可以使用AWS Management ConsoleAWS CLI、或AWS SDK 新增原則。如需指示，請參閱[新增和移除IAM身分權限](#)。

策略的格式如下：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "my-policy-name",
      "Effect": "Allow",
      "Action": [
        "service:action"
      ],
      "Resource": [
        "amazon-resource-name"
      ]
    }
  ]
}
```

Amazon 資源名稱 (ARN) 可唯一識別所有AWS資源，例如Amazon S3儲存貯體。您可以在策略中使用 ARN 來授與使用特定資源的特定動作的權限。例如，如果您想要授與Amazon S3值區及其子資料夾的讀取權限，您可以將下列程式碼新增至信任原則的Statement區段：

```
{
  "Effect": "Allow",
  "Action": [
    "s3:GetObject",
    "s3:ListBucket"
  ],
  "Resource": [
    "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
    "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
  ]
}
```

以下是授與Amazon Transcribe讀取 (GetObject,ListBucket) 和 write (PutObject) 權限給 Amazon S3值區及其子資料夾的範例政策：DOC-EXAMPLE-BUCKET

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

```
    "Action": [
      "s3:GetObject",
      "s3:ListBucket"
    ],
    "Resource": [
      "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
      "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3:PutObject"
    ],
    "Resource": [
      "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
      "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
    ]
  }
]
```

## 使用轉錄AWS Management Console

您可以使用AWS主控台進行批次和串流轉錄。如果您要轉錄位於Amazon S3值區中的媒體檔案，表示您正在執行批次轉錄。如果您要轉錄音訊資料的即時串流，表示您正在執行串流轉錄。

開始批次轉錄之前，您必須先將媒體檔案上傳到Amazon S3值區。若要使用串流轉錄AWS Management Console，您必須使用電腦麥克風。

若要檢視支援的媒體格式和其他媒體需求和限制，請參閱[資料輸入和輸出](#)。

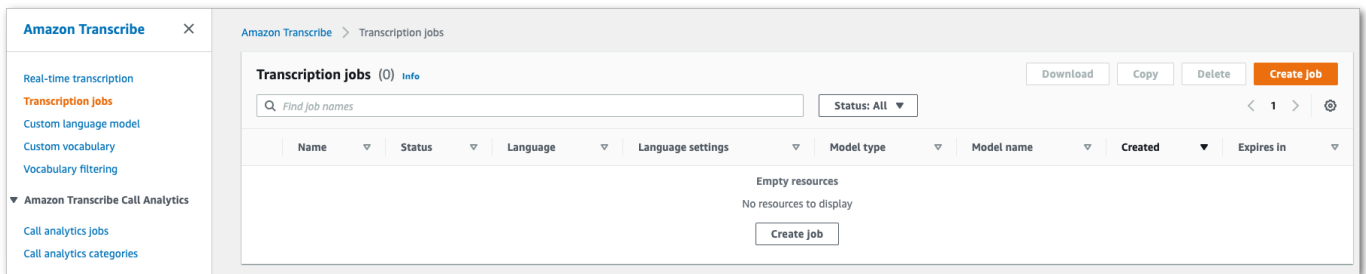
展開下列章節，以取得每個轉錄方法的簡短逐步解說。

### Batch 轉錄

首先，請確保您已上傳要轉錄到Amazon S3存儲桶中的媒體文件。如果您不確定如何執行此操作，請參閱Amazon S3使用者指南：[將物件上傳到值區](#)。

1. 從左側導覽窗格中選取 [轉錄工作]。[AWS Management Console](#)您會移至轉錄工作清單。





選取 [建立工作]。

2. 完成 [指定工作詳細資訊] 頁面上的欄位。

## Specify job details [Info](#)

### Job settings

**Name**

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

**Model type [Info](#)**

Choose the type of model to use for the transcription job.

**General model**

To use a model that is not specialized for a particular use case, choose this option. Configuration options vary between languages.

**Custom language model**

To use a model that you trained for your specific use case, choose this option. This model has fewer configuration options than the general model.

**Language settings**

You can transcribe your audio file in a language that you specify or have Amazon Transcribe identify and transcribe it in the predominant language.

**Specific language [Info](#)**

If you know the language spoken in your source audio, choose this option to get the most accurate results. The options available for additional processing vary between languages.

**Automatic language identification [Info](#)**

If you don't know the language spoken in your audio files, choose this option. You have access to fewer options for additional processing than if you choose **Specific language**.

**Language**

Choose the language of the input audio.

[▶ Additional settings](#)

輸入位置必須是Amazon S3值區內的物件。對於輸出位置，您可以選擇安全的Amazon S3服務管理值區，也可以指定自己的Amazon S3值區。

如果您選擇服務管理的值區，則可以在中檢視成績單預覽AWS Management Console，也可以從工作詳細資料頁面下載成績單 (請參閱下方)。

如果您選擇自己的Amazon S3值區，則無法在中看到預覽版，AWS Management Console而且必須前往Amazon S3值區才能下載成績單。

### Input data [Info](#)

**Input file location on S3**  
Choose an input audio or video file in Amazon S3.

Valid file formats: MP3, MP4, WAV, FLAC, AMR, OGG, and WebM.

### Output data

**Output data location type info [Info](#)**

**Service-managed S3 bucket**  
The output will be removed after 90 days when the job expires.

**Customer specified S3 bucket**  
The output will not be removed from bucket even after the job expires.

**Subtitle file format [Info](#)**

SRT (SubRip)

VTT (WebVTT)

### Tags - optional

A tag is a label you can add to a resource as metadata to help you organize, search, or filter your data. Each tag consists of a key and an optional value, in the form 'key:value'.

No tags associated with the resource.

You can add up to 50 more tags.

選取 Next (下一步)。

3. 在 [設定工作] 頁面上選取任何所需的選項。如果要使用 [自訂詞彙](#) 或 [自訂語言模型](#) 與轉錄一起使用，則必須在開始轉錄工作之前創建這些內容。

## Configure job - optional [Info](#)

### Audio settings

**Audio identification** [Info](#)  
Choose to split multi-channel audio into separate channels for transcription, or identify speakers in the input audio.

---

**Alternative results** [Info](#)  
Enable to view more transcription results

---

### Content removal

Content removal conceals information in the resulting transcript from your source audio file. Amazon Transcribe changes items in the transcript and does not modify the source audio.

**Automatic content redaction** [Info](#)  
Automatic content redaction removes personally identifiable information (PII) in your transcripts. Redactions in transcripts show up as [PII].

---

**Vocabulary filtering** [Info](#)  
Vocabulary filtering can remove, mask or tag specified words in the final transcript.

---

### Customization

**Custom vocabulary** [Info](#)  
A custom vocabulary improves the accuracy of recognizing words and phrases specific to your use case.

Cancel Previous Create job

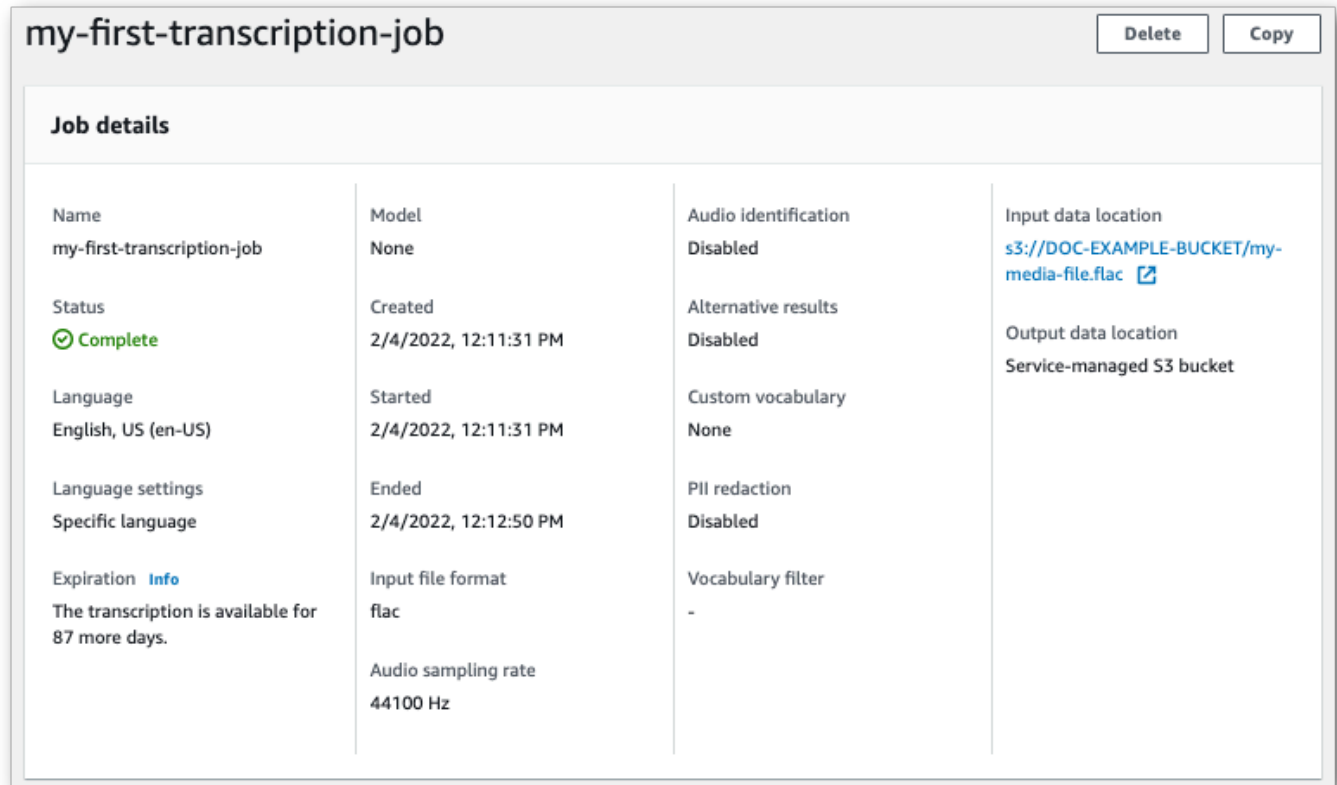
選取 [建立工作]。

- 您現在位於轉錄工作頁面上。您可以看到轉錄工作的狀態。完成後，選擇您的轉錄。

Transcription jobs (5) <a href="#">Info</a>								Download	Copy	Delete	Create job
<input type="text" value="Find job names"/>							Status: All ▼	< 1 > ⚙️			
Name	Status	Language	Language settings	Model type	Model name	Created	Expires in				
<input type="radio"/> my-first-transcription	<span style="color: green;">✔ Complete</span>	English, US (en-US)	Specific language	General	-	October 18 2021, 16:48 (UTC-07:00)	89 days				

5. 您現在正在檢視轉錄的 [Job 詳細資料] 頁面。在這裡，您可以查看設置轉錄工作時指定的所有選項。

若要檢視您的成績單，請在「輸出資料位置」下方的右欄中選取連結的檔案路徑。這將帶您到指定的Amazon S3輸出文件夾。選擇您的輸出文件，該文件現在具有 .json 擴展名。



The screenshot shows the 'Job details' page for a transcription job named 'my-first-transcription-job'. The page has a title bar with 'Delete' and 'Copy' buttons. The details are organized into a table with four columns: Name, Model, Audio identification, and Input data location. The status is 'Complete' with a green checkmark. The language is 'English, US (en-US)'. The job was created on 2/4/2022 at 12:11:31 PM and ended at 12:12:50 PM. The input file format is 'flac' and the audio sampling rate is '44100 Hz'. The output data location is a service-managed S3 bucket.

Job details			
Name	Model	Audio identification	Input data location
my-first-transcription-job	None	Disabled	s3://DOC-EXAMPLE-BUCKET/my-media-file.flac <a href="#">🔗</a>
Status	Created	Alternative results	Output data location
🟢 Complete	2/4/2022, 12:11:31 PM	Disabled	Service-managed S3 bucket
Language	Started	Custom vocabulary	
English, US (en-US)	2/4/2022, 12:11:31 PM	None	
Language settings	Ended	PII redaction	
Specific language	2/4/2022, 12:12:50 PM	Disabled	
Expiration <a href="#">Info</a>	Input file format	Vocabulary filter	
The transcription is available for 87 more days.	flac	-	
	Audio sampling rate		
	44100 Hz		

6. 下載成績單的方式取決於您選擇的是服務管理Amazon S3值區還是自己的Amazon S3值區。
  - a. 如果您選擇服務管理的值區，您可以在轉錄工作的資訊頁面上看到 [轉錄預覽] 窗格，以及 [下載] 按鈕。

**my-first-transcription-job**

---

**Job details**

Name my-first-transcription-job	Model None	Audio identification Disabled	Input data location <a href="#">s3://DOC-EXAMPLE-BUCKET/my-media-file.flac</a>
Status <span style="color: green;">✔ Complete</span>	Created 2/4/2022, 12:11:31 PM	Alternative results Disabled	Output data location Service-managed S3 bucket
Language English, US (en-US)	Started 2/4/2022, 12:11:31 PM	Custom vocabulary None	
Language settings Specific language	Ended 2/4/2022, 12:12:50 PM	PII redaction Disabled	
Expiration <a href="#">Info</a> The transcription is available for 87 more days.	Input file format flac	Vocabulary filter -	
	Audio sampling rate 44100 Hz		

---

**Transcription preview**  ▼

You can see the first 5,000 characters of the transcription text below. To download the full text, choose [Download full transcript](#).

Text
Audio identification
Subtitles

This is a preview of the content of your transcript. If your transcript is long, you may have to scroll to see the complete preview.

選取 [下載] 並選擇 [下載成績單]。

- b. 如果您選擇自己的Amazon S3值區，則轉錄工作資訊頁面上的 [轉錄] 預覽窗格中看不到任何文字。相反地，您會看到一個藍色的資訊方塊，其中包含指向您選擇的Amazon S3值區的連結。

**my-first-transcription-job**

---

**Job details**

Name my-first-transcription-job	Model None	Audio identification Disabled	Input data location <a href="s3://DOC-EXAMPLE-BUCKET/my-media-file.flac">s3://DOC-EXAMPLE-BUCKET/my-media-file.flac</a> <a href="#">↗</a>
Status <span style="color: green;">✔ Complete</span>	Created 2/7/2022, 11:42:17 AM	Alternative results Disabled	Output data location <a href="https://s3.us-west-2.amazonaws.com/DOC-EXAMPLE-BUCKET">https://s3.us-west-2.amazonaws.com/DOC-EXAMPLE-BUCKET</a> <a href="#">↗</a>
Language English, US (en-US)	Started 2/7/2022, 11:42:17 AM	Custom vocabulary None	
Language settings Specific language	Ended 2/7/2022, 11:43:37 AM	PII redaction Disabled	
Expiration <a href="#">Info</a> The transcription is available for 89 more days.	Input file format flac	Vocabulary filter -	
	Audio sampling rate 44100 Hz		

---

**Transcription preview**  ▼

Select download to save a local copy of the transcription.

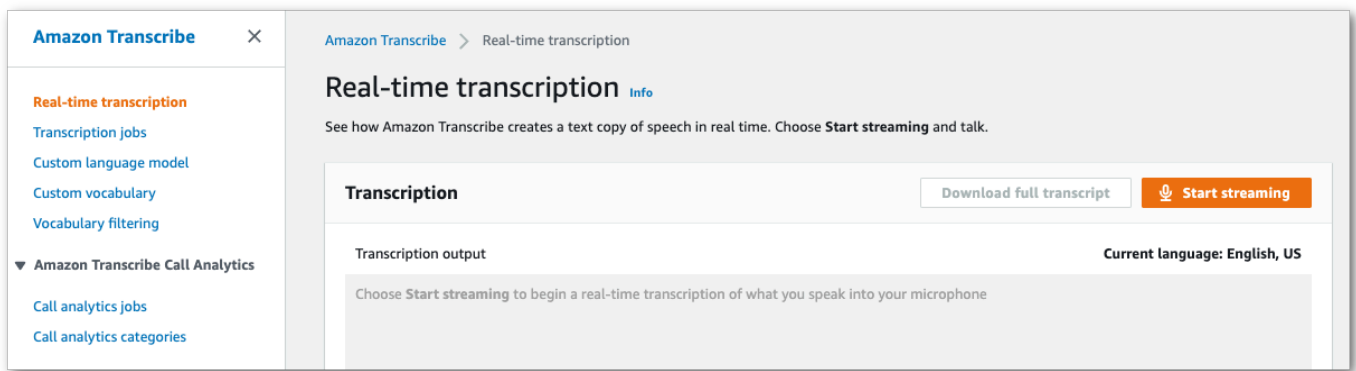
Text
Audio identification
Subtitles

ⓘ When you use your own S3 bucket for transcription output, Amazon Transcribe does not show the output in the console. You open the output file from your [S3 Bucket](#). [↗](#)

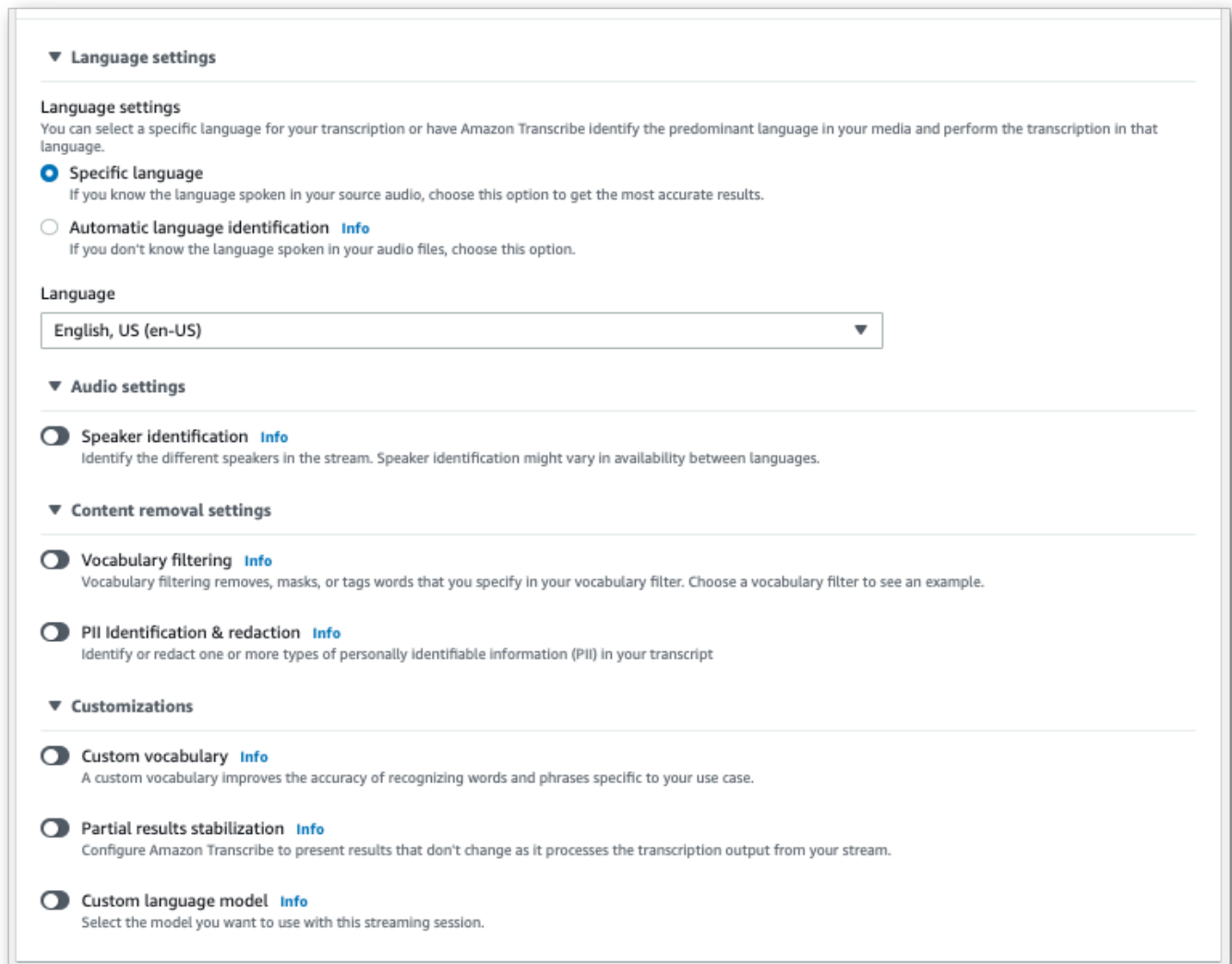
若要存取您的成績單，請使用 [Job 詳細資料] 窗格中 [輸出資料位置] 下的連結，或 [轉錄] 預覽窗格中藍色資訊方塊中的 S3 儲存Amazon S3貯體連結前往指定的儲存貯體。

## 串流轉錄

1. 在左側導覽窗格中 [AWS Management Console](#)，選取 [即時轉錄]。這會帶您前往主要的串流頁面，您可以在此選擇開始直播前的選擇選項。

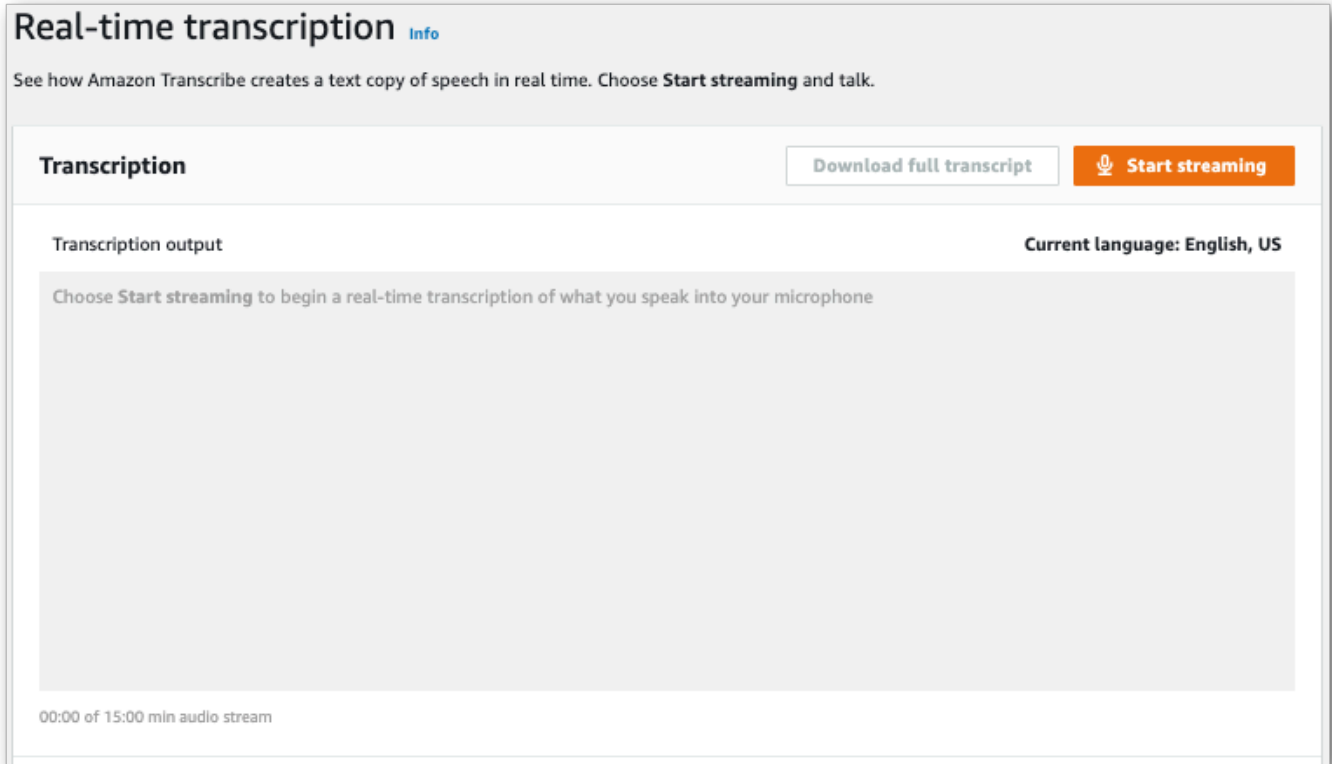


2. 在「轉錄輸出」方塊下方，您可以選擇各種語言和音訊設定。



3. 選取適當的設定後，捲動至頁面頂端並選擇 [開始串流]，然後開始對電腦麥克風說話。您可以實時查看自己的語音轉錄。





**Real-time transcription** [Info](#)

See how Amazon Transcribe creates a text copy of speech in real time. Choose **Start streaming** and talk.

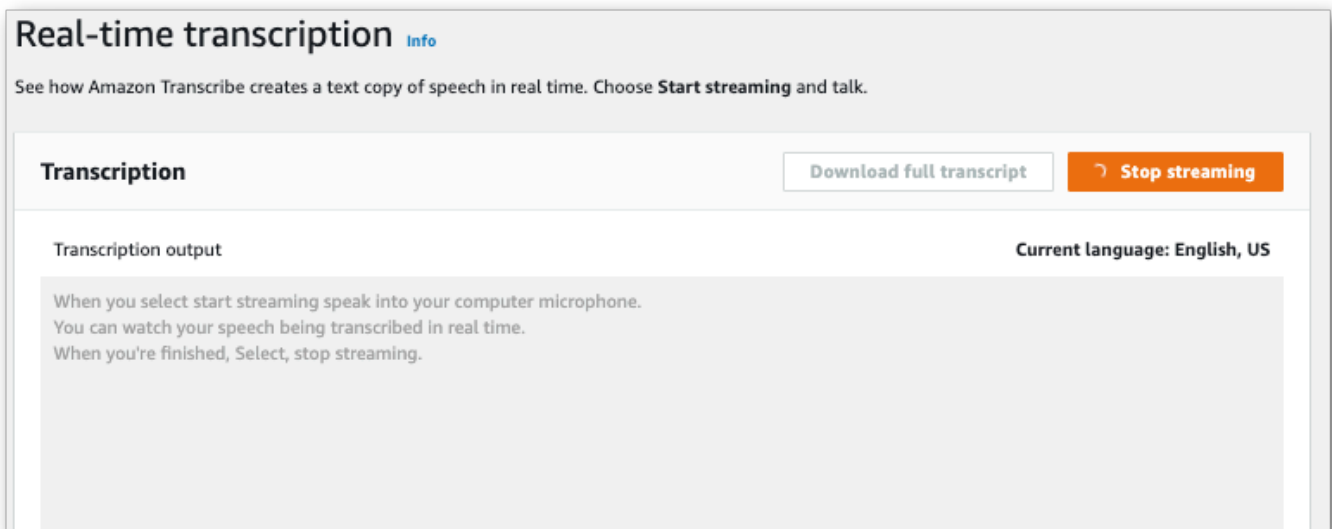
**Transcription** [Download full transcript](#) [Start streaming](#)

Transcription output Current language: English, US

Choose **Start streaming** to begin a real-time transcription of what you speak into your microphone

00:00 of 15:00 min audio stream

4. 完成時，請選擇 Cam (停止串流)。



**Real-time transcription** [Info](#)

See how Amazon Transcribe creates a text copy of speech in real time. Choose **Start streaming** and talk.

**Transcription** [Download full transcript](#) [Stop streaming](#)

Transcription output Current language: English, US

When you select start streaming speak into your computer microphone.  
You can watch your speech being transcribed in real time.  
When you're finished, Select, stop streaming.

您現在可以選取 [下載完整成績單] 來下載成績單。

## 使用轉錄AWS CLI

使用開始AWS CLI轉錄時，您可以在 CLI 層級執行所有命令。或者，您也可以執行您要使用的命令，然後執行包含要求主體的 JSON 檔案的AWS 區域和位置。本指南中的範例展示了這兩種方法；但是，本節著重於前一個方法。

AWS CLI不支援串流轉錄。

在繼續之前，請確保您已經：

- 將您的媒體文件上傳到存Amazon S3儲桶中。如果您不確定如何建立Amazon S3值區或上傳檔案，請參閱[建立您的第一個Amazon S3值區和將物件上傳到值區](#)。
- 安裝了 [AWS CLI](#)。

您可以在《AWS CLI指令[參考](#)》[Amazon Transcribe中找到的所有指AWS CLI令](#)。

### 開始新的轉錄工作

若要開始新的轉錄，請使用start-transcription-job指令。

1. 在終端機視窗中，輸入下列內容：

```
aws transcribe start-transcription-job \
```

下一行會出現> "，您現在可以繼續新增必要的參數，如下一個步驟所述。

您也可以省略 '\' 並附加所有參數，每個參數都用空格分隔。

2. 使用start-transcription-job指令時，您必須包含regiontranscription-job-namemedia、和language-code或identify-language。

如果要指定輸出位置，請在請求output-bucket-name中包含；如果要指定指定輸出存儲桶的子文件夾，也請包括output-key。

```
aws transcribe start-transcription-job \  
  --region us-west-2 \  
  --transcription-job-name my-first-transcription-job \  
  --media MediaFileUri=s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac \  
  --language-code en-US
```

如果附加所有參數，則此請求如下所示：

```
aws transcribe start-transcription-job --region us-west-2 --transcription-job-name my-first-transcription-job --media MediaFileUri=s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac --language-code en-US
```

如果您選擇不使用指定輸出值區 `output-bucket-name`，請將轉錄輸出 Amazon Transcribe 放置在服務管理的值區中。儲存在服務管理儲存貯體中的成績單會在 90 天後過期。

Amazon Transcribe 響應：

```
{
  "TranscriptionJob": {
    "TranscriptionJobName": "my-first-transcription-job",
    "TranscriptionJobStatus": "IN_PROGRESS",
    "LanguageCode": "en-US",
    "Media": {
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
    },
    "StartTime": "2022-03-07T15:03:44.246000-08:00",
    "CreationTime": "2022-03-07T15:03:44.229000-08:00"
  }
}
```

如果從 [TranscriptionJobStatus](#) `IN_PROGRESS` 變更為 `COMPLETED`，您的轉錄工作就會成功。若要查看更新 [TranscriptionJobStatus](#)，請使用 `get-transcription-job` 或 `list-transcription-job` 指令，如下節所示。

## 取得轉錄任務的狀態

若要取得轉錄工作的相關資訊，請使用 `get-transcription-job` 指令。

此命令唯一需要的參數是工作所 AWS 區域在的位置和作業的名稱。

```
aws transcribe get-transcription-job \
  --region us-west-2 \
  --transcription-job-name my-first-transcription-job
```

## Amazon Transcribe響應：

```
{
  "TranscriptionJob": {
    "TranscriptionJobName": "my-first-transcription-job",
    "TranscriptionJobStatus": "COMPLETED",
    "LanguageCode": "en-US",
    "MediaSampleRateHertz": 48000,
    "MediaFormat": "flac",
    "Media": {
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
    },
    "Transcript": {
      "TranscriptFileUri": "https://s3.the-URI-where-your-job-is-located.json"
    },
    "StartTime": "2022-03-07T15:03:44.246000-08:00",
    "CreationTime": "2022-03-07T15:03:44.229000-08:00",
    "CompletionTime": "2022-03-07T15:04:01.158000-08:00",
    "Settings": {
      "ChannelIdentification": false,
      "ShowAlternatives": false
    }
  }
}
```

如果您已為轉錄輸出選取自己的Amazon S3儲存貯體，則會列出此值區TranscriptFileUri。如果您已選取服務管理值區，則會提供暫時 URI；請使用此 URI 下載您的成績單。

### Note

服務管理Amazon S3值區的臨時 URI 僅在 15 分鐘內有效。如果您在使用 URI 時收到AccessDenied錯誤，get-transcription-job請再次執行要求以取得新的暫存 URI。

## 列出您的轉錄工作

若要列出指定的所有轉錄工作AWS 區域，請使用list-transcription-jobs指令。

此命令唯一必要的參數是轉錄工作所AWS 區域在的參數。

```
aws transcribe list-transcription-jobs \
```

```
--region us-west-2
```

Amazon Transcribe 響應：

```
{
  "NextToken": "A-very-long-string",
  "TranscriptionJobSummaries": [
    {
      "TranscriptionJobName": "my-first-transcription-job",
      "CreationTime": "2022-03-07T15:03:44.229000-08:00",
      "StartTime": "2022-03-07T15:03:44.246000-08:00",
      "CompletionTime": "2022-03-07T15:04:01.158000-08:00",
      "LanguageCode": "en-US",
      "TranscriptionJobStatus": "COMPLETED",
      "OutputLocationType": "SERVICE_BUCKET"
    }
  ]
}
```

## 刪除轉錄工作

若要刪除轉錄工作，請使用 `delete-transcription-job` 指令。

此命令唯一需要的參數是工作所 AWS 區域在的位置和作業的名稱。

```
aws transcribe delete-transcription-job \
  --region us-west-2 \
  --transcription-job-name my-first-transcription-job
```

要確認刪除請求是否成功，您可以運行 `list-transcription-jobs` 命令。您的任務應該會顯示在清單中。

## 使用 SDK 進行轉錄 AWS

您可以將 SDK 用於批次和串流轉錄。如果您要轉錄位於 Amazon S3 值區中的媒體檔案，表示您正在執行批次轉錄。如果您要轉錄音訊資料的即時串流，您要執行的是串流轉錄。

如需可搭配使用的程式設計語言清單 Amazon Transcribe，請參閱 [支援的程式設計語言](#)。請注意，並非所有 AWS 有 SDK 都支援串流轉錄。若要檢視支援的媒體格式和其他媒體需求和限制，請參閱 [資料輸入和輸出](#)。

有關所有可用 AWS SDK 和構建器工具的更多信息，請參閱[構建工具](#)。AWS

### Tip

如需使用 AWS SDK 的其他範例，包括特定功能、案例和跨服務範例，請參閱本章。[使用 SDK 進行 Amazon Transcribe 的代碼示例 AWS](#)

您也可以在这些 GitHub 儲存庫中找到 SDK 程式碼範例：

- [AWS 程式碼範例](#)
- [Amazon Transcribe 範例](#)

## 批次轉錄

您可以使用 Amazon S3 儲存貯體中媒體檔案的 URI 建立批次轉錄。如果您不確定如何建立儲存 Amazon S3 貯體或上傳檔案，請參閱[建立第一個 S3 儲存貯體](#)和[將物件上傳到儲存貯體](#)。

## Java

```
import software.amazon.awssdk.auth.credentials.AwsCredentialsProvider;
import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.transcribe.TranscribeClient;
import software.amazon.awssdk.services.transcribe.model.*;
import software.amazon.awssdk.services.transcribestreaming.model.LanguageCode;

public class TranscribeDemoApp {
    private static final Region REGION = Region.US_WEST_2;
    private static TranscribeClient client;

    public static void main(String args[]) {

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

        String transcriptionJobName = "my-first-transcription-job";
        String mediaType = "flac"; // can be other types
        Media myMedia = Media.builder()
            .mediaFileUri("s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-
file.flac")
```

```
        .build();

        String outputS3BucketName = "s3://DOC-EXAMPLE-BUCKET";
        // Create the transcription job request
        StartTranscriptionJobRequest request =
StartTranscriptionJobRequest.builder()
        .transcriptionJobName(transcriptionJobName)
        .languageCode(LanguageCode.EN_US.toString())
        .mediaSampleRateHertz(16000)
        .mediaFormat(mediaType)
        .media(myMedia)
        .outputBucketName(outputS3BucketName)
        .build();

        // send the request to start the transcription job
        StartTranscriptionJobResponse startJobResponse =
client.startTranscriptionJob(request);

        System.out.println("Created the transcription job");
        System.out.println(startJobResponse.transcriptionJob());

        // Create the get job request
        GetTranscriptionJobRequest getJobRequest =
GetTranscriptionJobRequest.builder()
        .transcriptionJobName(transcriptionJobName)
        .build();

        // send the request to get the transcription job including the job status
        GetTranscriptionJobResponse getJobResponse =
client.getTranscriptionJob(getJobRequest);

        System.out.println("Get the transcription job request");
        System.out.println(getJobResponse.transcriptionJob());
    }

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

## JavaScript

```
const { TranscribeClient, StartTranscriptionJobCommand } = require("@aws-sdk/client-transcribe"); // CommonJS import

const region = "us-west-2";
const credentials = {
  "accessKeyId": "",
  "secretAccessKey": "",
};

const input = {
  TranscriptionJobName: "my-first-transcription-job",
  LanguageCode: "en-US",
  Media: {
    MediaFileUri: "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
  },
  OutputBucketName: "DOC-EXAMPLE-BUCKET",
};

async function startTranscriptionRequest() {
  const transcribeConfig = {
    region,
    credentials
  };
  const transcribeClient = new TranscribeClient(transcribeConfig);
  const transcribeCommand = new StartTranscriptionJobCommand(input);
  try {
    const transcribeResponse = await transcribeClient.send(transcribeCommand);
    console.log("Transcription job created, the details:");
    console.log(transcribeResponse.TranscriptionJob);
  } catch(err) {
    console.log(err);
  }
}

startTranscriptionRequest();
```

## Python

```
import time
import boto3
```



```
def transcribe_file(job_name, file_uri, transcribe_client):
    transcribe_client.start_transcription_job(
        TranscriptionJobName = job_name,
        Media = {
            'MediaFileUri': file_uri
        },
        MediaFormat = 'flac',
        LanguageCode = 'en-US'
    )

    max_tries = 60
    while max_tries > 0:
        max_tries -= 1
        job = transcribe_client.get_transcription_job(TranscriptionJobName =
job_name)
        job_status = job['TranscriptionJob']['TranscriptionJobStatus']
        if job_status in ['COMPLETED', 'FAILED']:
            print(f"Job {job_name} is {job_status}.")
            if job_status == 'COMPLETED':
                print(
                    f"Download the transcript from\n"
                    f"\t{job['TranscriptionJob']['Transcript']
['TranscriptFileUri']}".)
                break
            else:
                print(f"Waiting for {job_name}. Current status is {job_status}.")
                time.sleep(10)

def main():
    transcribe_client = boto3.client('transcribe', region_name = 'us-west-2')
    file_uri = 's3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac'
    transcribe_file('Example-job', file_uri, transcribe_client)

if __name__ == '__main__':
    main()
```

## 串流轉錄

您可以使用串流媒體檔案或即時媒體串流，建立串流轉錄。

請注意，Amazon Transcribe 串流 AWS SDK for Python (Boto3) 不支援此標準。若要使用 Python 開始串流轉錄，請使用此[非同步 Python SDK](#)。Amazon Transcribe

## Java

以下範例是轉錄串流音訊的 Java 程式。

若要執行此範例，請注意以下項目：

- 您必須使用適用於 [Java 2.x 的 AWS SDK](#)。
- 用戶端必須使用 Java 1.8，以與 [Java 2.x 的 AWS SDK](#) 相容。
- 您指定的取樣率必須與音訊串流的實際取樣率相符。

另請參閱：[重試用於 Amazon Transcribe 串流的用戶端 \(Java SDK\)](#)。該程式碼會管理與 Amazon Transcribe 的連線，並在連線發生錯誤時重試資料的傳送。例如，如果發生暫時性網路錯誤，此用戶端會重新傳送失敗的請求。

```
public class TranscribeStreamingDemoApp {
    private static final Region REGION = Region.US_WEST_2;
    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 16,000 Hz, 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 myMediaFileName) {
        try {
            File inputFile = new
File(getClass().getClassLoader().getResource(myMediaFileName).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 static Subscription currentSubscription;

        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);

    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();
}
}
```

## JavaScript

```
const {
    TranscribeStreamingClient,
    StartStreamTranscriptionCommand,
} = require("@aws-sdk/client-transcribe-streaming");
const { createReadStream } = require("fs");
const { join } = require("path");

const audio = createReadStream(join(__dirname, "my-media-file.flac"),
    { highWaterMark: 1024 * 16});
```

```
const LanguageCode = "en-US";
const MediaEncoding = "pcm";
const MediaSampleRateHertz = "16000";
const credentials = {
  "accessKeyId": "",
  "secretAccessKey": "",
};
async function startRequest() {
  const client = new TranscribeStreamingClient({
    region: "us-west-2",
    credentials
  });

  const params = {
    LanguageCode,
    MediaEncoding,
    MediaSampleRateHertz,
    AudioStream: (async function* () {
      for await (const chunk of audio) {
        yield {AudioEvent: {AudioChunk: chunk}};
      }
    })(),
  };
  const command = new StartStreamTranscriptionCommand(params);
  // Send transcription request
  const response = await client.send(command);
  // Start to print response
  try {
    for await (const event of response.TranscriptResultStream) {
      console.log(JSON.stringify(event));
    }
  } catch(err) {
    console.log("error")
    console.log(err)
  }
}
startRequest();
```

## Python

以下範例是轉錄串流音訊的 Python 程式。

若要執行此範例，請注意以下項目：

- 您必須使用適用於 [Python 的 SDK](#)。
- 您指定的取樣率必須與音訊串流的實際取樣率相符。

```
import asyncio
# This example uses aiofile for asynchronous file reads.
# It's not a dependency of the project but can be installed
# with `pip install aiofile`.
import aiofile

from amazon_transcribe.client import TranscribeStreamingClient
from amazon_transcribe.handlers import TranscriptResultStreamHandler
from amazon_transcribe.model import TranscriptEvent

"""
Here's an example of a custom event handler you can extend to
process the returned transcription results as needed. This
handler will simply print the text out to your interpreter.
"""

class MyEventHandler(TranscriptResultStreamHandler):
    async def handle_transcript_event(self, transcript_event: TranscriptEvent):
        # This handler can be implemented to handle transcriptions as needed.
        # Here's an example to get started.
        results = transcript_event.transcript.results
        for result in results:
            for alt in result.alternatives:
                print(alt.transcript)

async def basic_transcribe():
    # Set up our client with your chosen Region
    client = TranscribeStreamingClient(region = "us-west-2")

    # Start transcription to generate async stream
    stream = await client.start_stream_transcription(
        language_code = "en-US",
        media_sample_rate_hz = 16000,
        media_encoding = "pcm",
    )

    async def write_chunks():
        # NOTE: For pre-recorded files longer than 5 minutes, the sent audio
        # chunks should be rate limited to match the real-time bitrate of the
```



```

# audio stream to avoid signing issues.
async with aiofile.AIOFile('filepath/my-media-file.flac', 'rb') as afp:
    reader = aiofile.Reader(afp, chunk_size = 1024 * 16)
    async for chunk in reader:
        await stream.input_stream.send_audio_event(audio_chunk = chunk)
    await stream.input_stream.end_stream()

# Instantiate our handler and start processing events
handler = MyEventHandler(stream.output_stream)
await asyncio.gather(write_chunks(), handler.handle_events())

loop = asyncio.get_event_loop()
loop.run_until_complete(basic_transcribe())
loop.close()

```

## C++

請參閱程式碼範例一章，取得[串流 C++ SDK 範例](#)。

## 搭配 AWS SDK 使用此服務

AWS 軟件開發套件 ( SDK ) 可用於許多流行的編程語言。每個 SDK 都提供 API、程式碼範例和說明文件，讓開發人員能夠更輕鬆地以偏好的語言建置應用程式。

SDK 文件	代碼範例
<a href="#">AWS SDK for C++</a>	<a href="#">AWS SDK for C++ 程式碼範例</a>
<a href="#">AWS CLI</a>	<a href="#">AWS CLI 程式碼範例</a>
<a href="#">AWS SDK for Go</a>	<a href="#">AWS SDK for Go 程式碼範例</a>
<a href="#">AWS SDK for Java</a>	<a href="#">AWS SDK for Java 程式碼範例</a>
<a href="#">AWS SDK for JavaScript</a>	<a href="#">AWS SDK for JavaScript 程式碼範例</a>
<a href="#">適用於 Kotlin 的 AWS SDK</a>	<a href="#">適用於 Kotlin 的 AWS SDK 程式碼範例</a>
<a href="#">AWS SDK for .NET</a>	<a href="#">AWS SDK for .NET 程式碼範例</a>
<a href="#">AWS SDK for PHP</a>	<a href="#">AWS SDK for PHP 程式碼範例</a>

SDK 文件	代碼範例
<a href="#">AWS Tools for PowerShell</a>	<a href="#">PowerShell 程式碼範例工具</a>
<a href="#">AWS SDK for Python (Boto3)</a>	<a href="#">AWS SDK for Python (Boto3) 程式碼範例</a>
<a href="#">AWS SDK for Ruby</a>	<a href="#">AWS SDK for Ruby 程式碼範例</a>
<a href="#">適用於 Rust 的 AWS SDK</a>	<a href="#">適用於 Rust 的 AWS SDK 程式碼範例</a>
<a href="#">適用於 SAP ABAP 的 AWS SDK</a>	<a href="#">適用於 SAP ABAP 的 AWS SDK 程式碼範例</a>
<a href="#">適用於 Swift 的 AWS SDK</a>	<a href="#">適用於 Swift 的 AWS SDK 程式碼範例</a>

如需此服務的特定範例，請參閱 [使用 SDK 進行 Amazon Transcribe 的代碼示例 AWS](#)。

#### 可用性範例

找不到所需的內容嗎？請使用本頁面底部的提供意見回饋連結申請程式碼範例。

## 使用 HTTP 或進行轉錄 WebSockets

Amazon Transcribe 同時支援 HTTP 進行批次處理 (HTTP/1.1) 和串流 (HTTP/2) 轉錄。WebSockets 支持流轉錄。

如果您要轉錄位於 Amazon S3 值區中的媒體檔案，表示您正在執行批次轉錄。如果您要轉錄音訊資料的即時串流，表示您正在執行串流轉錄。

HTTP 和 WebSockets 要求您使用 AWS 簽名版本 4 標頭驗證您的請求。如需詳細資訊，[請參閱簽署 AWS API 要求](#)。

### Batch 轉錄

您可以使用下列標頭發出批次 HTTP 要求：

- host
- x-amz-target
- 內容類型

- x-amz-content-sha256
- x-amz-date
- authorization

範例如StartTranscriptionJob下：

```
POST /transcribe HTTP/1.1
host: transcribe.us-west-2.amazonaws.com
x-amz-target: com.amazonaws.transcribe.Transcribe.StartTranscriptionJob
content-type: application/x-amz-json-1.1
x-amz-content-sha256: string
x-amz-date: YYYYMMDDTHHMMSSZ
authorization: AWS4-HMAC-SHA256 Credential=access-key/YYYYMMSS/us-west-2/transcribe/
aws4_request, SignedHeaders=content-type;host;x-amz-content-sha256;x-amz-date;x-amz-
target;x-amz-security-token, Signature=string

{
  "TranscriptionJobName": "my-first-transcription-job",
  "LanguageCode": "en-US",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
  },
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",
  "OutputKey": "my-output-files/"
}
```

API 參考中列出了其他操作和參數；所有AWS API 操作的通用參數都列在「[通用參數](#)」部分中。其他簽名元素在[AWS簽名版本 4 請求的元素](#)中詳細說明。

## 串流轉錄

使用 HTTP/2 進行串流轉錄，並 WebSockets 且比使用 SDK 更多參與。建議您在設定第一個直播前先查看此[設定串流轉錄](#)區段。

如需這些方法的詳細資訊，請參閱[設定 HTTP/2 串流](#)或[設定串 WebSocket 流](#)。

### Note

我們強烈建議您使用 SDK 進行串流轉錄。如需支援的 SDK 清單，請參閱[支援的程式設計語言](#)。

## 轉錄串流音訊

使用 Amazon Transcribe 流媒體，您可以為媒體內容生成實時轉錄。與包括上傳媒體檔案的批次轉錄不同，串流媒體會即時傳送至 Amazon Transcribe。Amazon Transcribe 然後返回成績單，也是實時的。

串流媒體可以包括預先錄製的媒體 (電影、音樂和播客) 和即時媒體 (實況新聞廣播)。常見的串流使用案例 Amazon Transcribe 包括體育賽事的即時隱藏式字幕，以及對話務中心音訊的即時監控。

串流內容會以一系列連續資料封包或 Amazon Transcribe 即時轉錄的「區塊」形式提供。批次使用串流的優點包括應用程式中的即時 speech-to-text 功能和更快的轉錄時間。但是，在某些情況下，轉錄速度加快可能導致準確性有限。

Amazon Transcribe 提供以下串流選項：

- [軟體開發套件](#) (首選)
- [HTTP/2](#)
- [WebSockets](#)
- [AWS Management Console](#)

若要在中轉錄串流音訊 AWS Management Console，請對著電腦麥克風說話。

### Tip

如需 SDK 程式碼範例，請參閱上的[AWS 範例存放庫](#) GitHub。

串流轉錄支援的音訊格式為：

- FLAC
- Ogg 容器中的 OPUS 編碼音訊
- PCM(僅簽名 16 位小端序音訊格式，不包括 WAV)

建議使用無失真格式 (FLAC 或 PCM)。

**Note**

並非所有語言都支援串流轉錄。請參閱[支援語言表格](#)中的「資料輸入」欄，了解詳細資訊。

若要檢視串流轉錄的 Amazon Transcribe 區域可用性，請參閱：[Amazon Transcribe 端點和配額](#)。

## 最佳實務

以下建議可提高串流轉錄效率：

- 如果可能，請使用 PCM 編碼的音訊。
- 確保您的串流盡可能即時取得。
- 延遲需視音訊塊的大小。如果您可以依據音訊類型 (例如 PCM) 指定塊大小，請將每塊設定為 50 ms 和 200 ms 間。您可以利用下列公式計算音訊區塊大小：

```
chunk_size_in_bytes = chunk_duration_in_millisecond / 1000 * audio_sample_rate * 2
```

- 使用統一的塊大小。
- 確保您正確指定音訊聲道的數量。
- 對於單聲道 PCM 音訊，每個樣本由兩個位元組組成，因此每個塊應由偶數位元組組成。
- 對於雙聲道 PCM 音訊，每個樣本由四個位元組組成，因此每個塊應該是 4 個位元組的倍數。
- 您的音訊串流不包含語音時，請編碼並傳送相同數量的靜音。例如，PCM 的靜音是零位元組的串流。
- 請務必為您的音訊指定正確的取樣率。如果可能，請以 16,000 Hz 的取樣率錄製；在品質和透過網路傳送的資料量間取得最佳平衡。請注意，多數高階麥克風的錄音速度為 44,100 Hz 或 48,000 Hz。

## 串流和部分結果

由於流媒體是實時工作的，因此成績單是在部分結果中生成的。Amazon Transcribe 根據自然語音片段 (例如揚聲器的變化或音頻中的暫停) 分解傳入的音頻流。系統會傳回轉錄事件串流中您的應用程式，每次回應皆會包含更多轉錄的語音，直到整個區段轉錄完成。

如下列程式碼區塊所示的近似值。您可以登入 [AWS Management Console](#)，選擇即時轉錄並對著麥克風說話，以檢視執行中的流程。說話時，看著轉錄輸出窗格。

在此範例中，每一行都是音訊區段的部分結果。

```
The  
The Amazon.  
The Amazon is  
The Amazon is the law.  
The Amazon is the largest  
The Amazon is the largest ray  
The Amazon is the largest rain for  
The Amazon is the largest rainforest.  
The Amazon is the largest rainforest on the  
The Amazon is the largest rainforest on the planet.
```

部分結果會在 [Results](#) 物件內的轉錄輸出中顯示。此外，在這個對象塊是一個 `IsPartial` 字段。如果此欄位為 `true`，代表您的轉錄區段尚未完成。您可以在下方檢視不完整區段與完整區段之間的差異：

```
"IsPartial": true (incomplete segment)
```

```
"Transcript": "The Amazon is the largest rainforest."
```

```
"EndTime": 4.545,
```

```
"IsPartial": true,
```

```
"ResultId": "12345a67-8bc9-0de1-2f34-a5b678c90d12",
```

```
"StartTime": 0.025
```

```
"IsPartial": false (complete segment)
```

```
"Transcript": "The Amazon is the largest rainforest on the planet."
```

```
"EndTime": 6.025,
```

```
"IsPartial": false,
```

```
"ResultId": "34567e89-0fa1-2bc3-4d56-78e90123456f",
```

```
"StartTime": 0.025
```

完整區段的每個單字都有相關的可信度分數，是介於 0 和 1 間的值。較大的值表示正確轉錄單字的可能性更高。

#### Tip

音訊區段的 `StartTime` 和 `EndTime` 可用於同步轉錄輸出與視訊對話。

如果您執行的應用程式需要低延遲，您可能需要使用[部分結果穩定](#)功能。

## 部分結果穩定

Amazon Transcribe 開始串流音訊後，立即開始傳回轉錄結果。它以遞增方式傳回這些部分結果，直到在自然語音區段層級產生完成的結果。自然語音區段是包含暫停或發言者改變的連續語音。

Amazon Transcribe 繼續輸出部分結果，直到產生語音區段的最終轉錄結果為止。由於語音辨識可能會在了解上下文時修改字詞，因此串流轉錄可能會隨著新的部分結果輸出而變化。

此流程提供每個語音區段兩個選項：

- 等待完成的區段
- 使用區段的部分結果

部分結果穩定化 Amazon Transcribe 會改變產生每個完整區段最終轉錄結果的方式。啟動時，部分結果中僅最後幾個單字可以變更。因此，轉錄的準確性可能會受到影響。不過，傳回文字記錄的速度會在沒有部分結果穩定功能時更快。在為直播加上字幕或產生字幕時，可能會有助於減少延遲。

下列範例顯示部分結果穩定功能未啟動及啟動時，如何處理相同的音訊串流。請注意，您可以將穩定性等級設定為低、中或高。低穩定性提供最高的準確性。穩定性的轉錄速度更快，但準確性略低。

「文字記錄」：	"EndTime":	"IsPartial":
部分結果穩定功能未啟用		
The	0.545	true
The	1.045	true
The Amazon.	1.545	true
The Amazon is	2.045	true
The Amazon is the law.	2.545	true
The Amazon is the	3.045	true
largest	3.545	true
The Amazon is the	4.045	true
largest ray	4.545	true
The Amazon is the	5.045	true
largest rain for	5.545	true
The Amazon is the	6.025	true
largest rainforest.	6.025	false

「文字記錄」：	"EndTime":	"IsPartial":
<p>The Amazon is the largest rainforest on the</p> <p>The Amazon is the largest rainforest on the planet.</p> <p>The Amazon is the largest rainforest on the planet.</p> <p>The Amazon is the largest rainforest on the planet.</p>		

啟用局部結果穩定 (高穩定性)



「文字記錄」：	"EndTime":	"IsPartial":
The	0.515	true
The	1.015	true
The Amazon.	1.515	true
The Amazon is	2.015	true
The Amazon is the large	2.515	true
The Amazon is the	3.015	true
largest	3.515	true
The Amazon is the	4.015	true
largest rainfall.	4.515	true
The Amazon is the	5.015	true
largest rain forest.	5.515	true
The Amazon is the	6.015	true
largest rain forest on	6.335	true
The Amazon is the	6.335	false
largest rain forest on		
the planet.		
The Amazon is the		
largest rain forest on		
the planet.		
The Amazon is the		
largest rain forest on		
the planet.		
The Amazon is the		
largest rain forest on		
the planet.		
The Amazon is the		
largest rain forest on		
the planet.		
The Amazon is the		
largest rain forest on		
the planet.		
The Amazon is the		
largest rain forest on		
the planet.		
The Amazon is the		
largest rain forest on		
the planet.		

當您啟動部分結果穩定功能時，Amazon Transcribe 會使用Stable欄位來指出項目是否穩定，其中「item」指的是轉錄文字或標點符號。Stable 的值是 true 或 false。標記為 false(不穩定)的項目在轉錄區段時更有可能改變。相反地，標記為 true(穩定)的項目不會改變。

您可以選擇顯示不穩定的單字，讓字幕與語音保持一致。即使字幕隨著內容新增而稍有變更，這也是比週期性文字連發(可能會或不會與語音一致)更好的使用者體驗。

您也可以選擇以不同的格式，顯示不穩定的文字，例如斜體，讓觀眾知道這些單字可能會變更。顯示部分結果會限制指定時間內顯示的文字數量。您處理影片字幕，面臨空間限制問題時，這可能很重要。

### 透過 Machine L AWS earning 部落格深入瞭解

若要了解透過即時轉錄提升準確性的更多資訊，請參閱：

- [透過 Amazon Transcribe 部分結果穩定功能改善串流轉錄體驗](#)
- [「那是什麼？」使用 Amazon Transcribe 提高直播的字幕準確性](#)

## 部分結果穩定範例輸出

下列範例輸出顯示不完整區段 ("IsPartial": true) 的 Stable 標記。您可以看到，單字「到」和「Amazon」是不穩定的，因此可能會在區段完成前變更。

```
"Transcript": {
  "Results": [
    {
      "Alternatives": [
        {
          "Items": [
            {
              "Content": "Welcome",
              "EndTime": 2.4225,
              "Stable": true,
              "StartTime": 1.65,
              "Type": "pronunciation",
              "VocabularyFilterMatch": false
            },
            {
              "Content": "to",
              "EndTime": 2.8325,
              "Stable": false,
              "StartTime": 2.4225,
              "Type": "pronunciation",
              "VocabularyFilterMatch": false
            },
            {
              "Content": "Amazon",
              "EndTime": 3.635,
              "Stable": false,
              "StartTime": 2.8325,
              "Type": "pronunciation",
              "VocabularyFilterMatch": false
            }
          ]
        }
      ]
    }
  ]
}
```

```
        },
        {
            "Content": ".",
            "EndTime": 3.635,
            "Stable": false,
            "StartTime": 3.635,
            "Type": "punctuation",
            "VocabularyFilterMatch": false
        }
    ],
    "Transcript": "Welcome to Amazon."
}
],
"EndTime": 4.165,
"IsPartial": true,
"ResultId": "12345a67-8bc9-0de1-2f34-a5b678c90d12",
"StartTime": 1.65
}
]
```

## 設定串流轉錄

本節擴充主要[串流傳輸](#)部分。它旨在為想要使用 HTTP/2 或 WebSockets 直接設置其流而不是 SDK 的用戶提供信息。AWS 本節中的資訊也可用於建置自己的 SDK。

### Important

我們強烈建議您直接使用 SDK 而不是使用 HTTP/2。WebSockets SDK 是轉錄資料串流最簡單且最可靠的方法。若要使用 AWS SDK 開始串流，請參閱[使用 SDK 進行轉錄 AWS](#)。

## 設定 HTTP/2 串流

用於流式傳輸轉錄請求的 [HTTP/2 協議](#)的關鍵組件如下：Amazon Transcribe

- 標頭框架。這包含請求的 HTTP/2 標頭，以及授權標頭中的簽名，該簽名 Amazon Transcribe 用作種子簽名來簽署數據幀。
- [事件串流編碼](#)中包含中繼資料和原始音訊位元組的一或多個訊息框架。
- 結束框架。這是[事件串流編碼](#)中有空內文的簽章訊息。

**Note**

Amazon Transcribe 每個 HTTP/2 會話僅支持一個流。如果您嘗試使用多個串流，轉錄要求會失敗。

1. 將下列原則附加至提出要求的 IAM 角色。如需詳細資訊，請參閱[新增 IAM 原則](#)。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "my-transcribe-http2-policy",
      "Effect": "Allow",
      "Action": "transcribe:StartStreamTranscription",
      "Resource": "*"
    }
  ]
}
```

2. 若要開始工作階段，請將 HTTP/2 請求傳送至 Amazon Transcribe。

```
POST /stream-transcription HTTP/2
host: transcribestreaming.us-west-2.amazonaws.com
X-Amz-Target: com.amazonaws.transcribe.Transcribe.StartStreamTranscription
Content-Type: application/vnd.amazon.eventstream
X-Amz-Content-Sha256: string
X-Amz-Date: YYYYMMDDTHHMMSSZ
Authorization: AWS4-HMAC-SHA256 Credential=access-key/YYYYMMDD/us-west-2/transcribe/aws4_request, SignedHeaders=content-type;host;x-amz-content-sha256;x-amz-date;x-amz-target;x-amz-security-token, Signature=string
x-amzn-transcribe-language-code: en-US
x-amzn-transcribe-media-encoding: flac
x-amzn-transcribe-sample-rate: 16000
transfer-encoding: chunked
```

[《API 參考資料》](#) 中列出其他操作和參數；所有 AWS API 操作的通用參數都會於[通用參數](#)一節列出。

Amazon Transcribe 發送以下響應：

```
HTTP/2.0 200
x-amzn-transcribe-language-code: en-US
x-amzn-transcribe-media-encoding: flac
x-amzn-transcribe-sample-rate: 16000
x-amzn-request-id: 8a08df7d-5998-48bf-a303-484355b4ab4e
x-amzn-transcribe-session-id: b4526fcf-5eee-4361-8192-d1cb9e9d6887
content-type: application/json
```

3. 建立包含音訊資料的音訊事件。將下表中所述的標頭與音訊位元組塊結合，成為事件編碼訊息。若要建立事件訊息的承載，請以原始位元組格式使用緩衝區。

標頭名稱位元組長度	標頭名稱 (字串)	標頭值類型	值字串位元組長度	值字串 (UTF-8)
13	:content-type	7	24	application/octet-stream
11	:event-type	7	10	AudioEvent
13	:message-type	7	5	事件

此範例請求中的二進位資料是以 base64 編碼。在實際請求中，資料是原始位元組。

```
:content-type: "application/vnd.amazon.eventstream"
:event-type: "AudioEvent"
:message-type: "event"
Uk1GRjzxpQBxQVZFZm10IBAAAAABAAEAgD4AAAB9AAACABAAZGF0YVVTwPQAAAAAAAAAAAAAAAAAAD//wIA/f8EAA==
```

4. 建立音訊訊息，其中包含您的音訊資料。
  - a. 您的資訊訊息資料框架包含事件編碼標頭，其中包含目前日期，以及音訊區塊和音訊事件的簽章。

標頭名稱位元組長度	標頭名稱 (字串)	標頭值類型	值字串位元組長度	Value
16	:chunk-signature	6	各有不同	產生的簽章
5	:date	8	8	timestamp

此請求中的二進位資料是以 base64 編碼。在實際請求中，資料是原始位元組。

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

```
AAAA0gAAAIKVoRFcTTcjb250ZW50LXR5cGUHABhhcHBsaWNhdGlvbi9vY3RldC1zdHJlYW0L0mV2ZW50LXR5cGUHAApBdWRpb0V2ZW50DTptZXNzYwd1LXR5cGUHAAV1dmVudAxDb256ZW50LVR5cGUHABphcHBsaWNhdGlvbi94LWFtei1qc29uLTEuMVJJRky88T0AV0FWRWZtdCAQAAAAAQABAIA
+AAAAfQAAAQAQAGRhdGFU8D0AAAAA
AAAAAAAAAAAA//8CAP3/BAC7QLFf
```

- b. 如 [Signature 第 4 版中的建立字串以簽署](#) 所述，建構要簽署的字串。您的字串遵循以下格式：

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

- *DateTime*：建立簽名的日期和時間。格式為 YYYYMMDDTHHMMSSZ，其中 YYYY = 年，MM = 月，DD = 日，HH = 小時，MM = 分鐘，SS = 秒，而「T」和「Z」為固定字元。如需詳細資訊，請參閱 [Signature 第 4 版的處理日期](#)。

- **Keypath**：格式 `date/region/service/aws4_request` 中的簽章範圍。例如 `20220127/us-west-2/transcribe/aws4_request`。
  - **Hex**：此函數會將其輸入編碼為十六進位表示法。
  - **priorSignature**：前一個框架的簽章。對於第一個資料框架，請使用標頭框架的簽章。
  - **HexHash**：一個函數，首先創建其輸入的 SHA-256 哈希，然後使用十六進制函數對哈希進行編碼。
  - **non SignatureHeaders**：編碼為字符串的 `DateTime` 標頭。
  - **payload**：包含音訊事件資料的位元組緩衝區。
- c. 從您的 AWS 密鑰訪問密鑰導出簽名密鑰，並用它來簽署 `stringToSign`。為了提供更大程度的保護，衍生的金鑰有專屬的日期、服務和 AWS 區域。如需詳細資訊，請參閱 [AWS Signature 第 4 版的計算簽章](#)。

確保您實作該 `GetSignatureKey` 函數以衍生出您的簽名密鑰。如果您尚未衍生出簽名密鑰，請參閱 [Signature 第 4 版中如何衍生簽名密鑰的範例](#)。

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

- **HMACSHA256**：此函數會使用 SHA-256 雜湊函數以建立簽章。
- **派生 SigningKey**：簽名版本 4 簽名密鑰。
- **string ToSign**：您為數據框計算的字符串。

在計算資料框架的簽章之後，請建構位元組緩衝區，其中包含日期、簽章和音訊事件承載。將位元組陣列傳送到 Amazon Transcribe 以進行轉錄。

5. 若要指出音訊串流完成，請傳送結束框架 (空資料框架)，其中僅包含日期和簽章。您建構該結束框架的方式與您建構資料框架的方式相同。

Amazon Transcribe 使用發送到您的應用程序的轉錄事件流進行響應。該回應是以事件串流編碼。它包含標準序言和下列標頭。

標頭名稱位元組長度	標頭名稱 (字串)	標頭值類型	值字串位元組長度	值字串 (UTF-8)
13	:content-type	7	16	application/json
11	:event-type	7	15	TranscriptEvent

標頭名稱位元組長度	標頭名稱 (字串)	標頭值類型	值字串位元組長度	值字串 (UTF-8)
13	:message-type	7	5	事件

事件是以原始位元組格式傳送。在這個範例中，位元組是以 base64 編碼的。

```
AAAAUwAAAEP1RHpYBTpkYXRlCAAAAWiXUkMLEDpjaHVuay1zaWduYXR1cmUGACct6Zy+uymwEK2Srlp/
zVBI
5eGn83jdBwCaRUBJA+eaDafqjqI=
```

若要查看轉錄結果，請使用事件串流編碼，將原始位元組解碼。

```
:content-type: "application/vnd.amazon.eventstream"
:event-type: "TranscriptEvent"
:message-type: "event"

{
  "Transcript":
    {
      "Results":
        [
          results
        ]
    }
}
```

- 若要結束串流，請將空白音訊事件傳送至 Amazon Transcribe。建立與任何其他音訊事件完全相同的音訊事件，但具有空承載。簽署事件，並將簽章併入 :chunk-signature 標頭中，如下所示：

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

## 處理 HTTP/2 串流錯誤

如果在處理媒體串流時發生錯誤，請 Amazon Transcribe 傳送例外狀況回應。回應是以事件串流編碼的。



回應包含標準序言和下列標頭：

標頭名稱位元組長度	標頭名稱 (字串)	標頭值類型	值字串位元組長度	值字串 (UTF-8)
13	:content-type	7	16	application/json
11	:event-type	7	19	BadRequest異常
13	:message-type	7	9	例外狀況

當異常回應解碼時，它包含下列資訊：

```
:content-type: "application/vnd.amazon.eventstream"
:event-type: "BadRequestException"
:message-type: "exception"
```

*Exception message*

## 設定串 WebSocket 流

用於流式傳輸轉錄請求的[WebSocket協議](#)的關鍵組件 Amazon Transcribe 是：

- 升級請求。這包含請求的查詢參數，以及 Amazon Transcribe 用作種子簽名來簽署數據框的簽名。
- [事件串流編碼](#)中包含中繼資料和原始音訊位元組的一或多個訊息框架。
- 結束框架。這是[事件串流編碼](#)中有空內文的簽章訊息。

### Note

Amazon Transcribe 每個 WebSocket 工作階段僅支援一個串流。如果您嘗試使用多個串流，轉錄要求會失敗。

1. 將下列原則附加至提出要求的 IAM 角色。如需詳細資訊，請參閱[新增 IAM 原則](#)。

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

```

    {
      "Sid": "my-transcribe-websocket-policy",
      "Effect": "Allow",
      "Action": "transcribe:StartStreamTranscriptionWebSocket",
      "Resource": "*"
    }
  ]
}

```

2. 若要開始工作階段，請以下列格式建立預先簽署的 URL。已加入分行符號以提高可讀性。

```

GET wss://transcribestreaming.us-west-2.amazonaws.com:8443/stream-transcription-
websocket?
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=access-key%2FYYYYMMDD%2Fus-west-2%2Ftranscribe%2Faws4_request
&X-Amz-Date=YYYYMMDDTHHMMSSZ
&X-Amz-Expires=300
&X-Amz-Security-Token=security-token
&X-Amz-Signature=string
&X-Amz-SignedHeaders=content-type%3Bhost%3Bx-amz-date
&language-code=en-US
&media-encoding=flac
&sample-rate=16000

```

#### Note

X-Amz-Expires 的最大值為 300(5 分鐘)。

[《API 參考資料》](#) 中列出其他操作和參數；所有 AWS API 操作的通用參數都會於[通用參數](#)一節列出。

若要為請求建構 URL 及建立 [Signature 第 4 版簽章](#)，請參閱下列步驟。範例是虛擬程式碼。

- a. 建立正式請求。正式請求是其中包括來自您的請求的資訊的字串 (使用標準化格式)。這樣可以確保在 AWS 收到請求時，它可以計算您為 URL 創建的相同簽名。如需詳細資訊，請參閱 [Signature 第 4 版的建立正式請求](#)。

```

# HTTP verb
method = "GET"
# Service name

```

```

service = "transcribe"
# Region
region = "us-west-2"
# Amazon Transcribe streaming endpoint
endpoint = "wss://transcribestreaming.us-west-2.amazonaws.com:8443"
# Host
host = "transcribestreaming.us-west-2.amazonaws.com:8443"
# Date and time of request
amz-date = YYYYMMDDTHHMMSSZ
# Date without time for credential scope
datestamp = YYYYMMDD

```

- b. 建立正式 URI，這是介於網域與查詢字串間的部分 URI。

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

- c. 建立正式標頭和已簽章標頭。請注意正式標頭中的結尾 \n。

- 附加小寫標頭名稱，後面接著冒號 (:)。
- 為該標頭附加逗號分隔的值清單。不要在有多個值的標頭中排序值。
- 附加新的一行 (\n)。

```

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

```

- d. 演算法必須符合雜湊演算法。請使用 SHA-256。

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

- e. 建立憑證的範圍，將衍生的金鑰範圍限定於提出請求的日期、AWS 區域和服務。例如 `20220127/us-west-2/transcribe/aws4_request`。

```

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

```

- f. 建立正式查詢字串。查詢字串值必須是 URL 編碼並以名稱排序。

- 依照字元字碼指標的參數名稱遞增排序。具有重複名稱的參數應依數值排序。例如，以大寫字母 F 開頭的參數名稱，放在以小寫字母 b 開頭的參數名稱之前。

- URI 編碼不能執行 RFC 3986 所定義的任何未預留字元：A-Z、a-z、0-9、連字號 (-)、底線 (\_)、句點 (.) 和波狀符號 (~)。
- 對所有其他含有 %XY 的字元執行百分比編碼，其中 X 和 Y 是十六進位字元 (0-9 和大寫 A-F)。例如，空間字元必須編碼為 %20 (而非像有些編碼結構描述那樣包含「+」)；延伸的 UTF-8 字元必須採用 %XY%ZA%BC 格式。
- 對參數值中的任何等於 (=) 字元進行雙倍編碼。

```
canonical_querystring = "X-Amz-Algorithm=" + algorithm
canonical_querystring += "&X-Amz-Credential="+ URI-encode(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=flac&sample-
rate=16000"
```

- g. 建立承載的雜湊。對於 GET 請求，承載為空字串。

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

- h. 結合下列元素以建立正式請求。

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

3. 建立要簽署的字串，其中包含有關您的請求的中繼資料。您在下一個步驟計算請求簽章時，使用此字串登入。如需詳細資訊，請參閱 [Signature 第 4 版的建立簽署字串](#)。

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

4. 計算簽章。為此，請從您的密鑰訪問 AWS 密鑰導出簽名密鑰。為了提供更大程度的保護，衍生的金鑰有專屬的日期、服務和 AWS 區域。您使用衍生的金鑰以簽署請求。如需詳細資訊，請參閱[計算簽名版本 4 的 AWS 簽名](#)。

確保您實作該 `GetSignatureKey` 函數以衍生出您的簽名密鑰。如果您尚未衍生出簽名密鑰，請參閱 [Signature 第 4 版中如何衍生簽名密鑰的範例](#)。

```
#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
```

函數 `HMAC(key, data)` 代表 HMAC-SHA256 函數，會以二進位格式傳回結果。

5. 將簽署資訊新增至請求並建立請求 URL。

您計算簽章之後，請將簽章新增到查詢字串。如需詳細資訊，請參閱[新增簽章到請求](#)。

首先將身分驗證資訊新增到查詢字串。

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

然後，為請求建立 URL。

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

將請求 URL 與您的 WebSocket 圖書館一起使用，以向其提出請求 Amazon Transcribe。

6. 要求 Amazon Transcribe 必須包含下列標頭。通常，這些標頭由您的 WebSocket 用戶端程式庫管理。

```
Host: transcribestreaming.us-west-2.amazonaws.com:8443
Connection: Upgrade
Upgrade: websocket
Origin: URI-of-WebSocket-client
Sec-WebSocket-Version: 13
Sec-WebSocket-Key: randomly-generated-string
```

7. 當 Amazon Transcribe 收到您的 WebSocket 請求時，它會以 WebSocket 升級回應作出回應。通常，您的 WebSocket 圖書館管理此響應並設置與之通信的套接字 Amazon Transcribe。

以下是來自的回應 Amazon Transcribe。已加入分行符號以提高可讀性。

```
HTTP/1.1 101 WebSocket Protocol Handshake

Connection: upgrade
Upgrade: websocket
websocket-origin: wss://transcribestreaming.us-west-2.amazonaws.com:8443
websocket-location: transcribestreaming.us-west-2.amazonaws.com:8443/stream-
transcription-websocket?
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIAIOSFODNN7EXAMPLE%2F20220208%2Fus-west-2%2Ftranscribe
%2Faws4_request
&X-Amz-Date=20220208T235959Z
&X-Amz-Expires=300
&X-Amz-Signature=Signature Version 4 signature
&X-Amz-SignedHeaders=host
&language-code=en-US
&session-id=String
&media-encoding=flac
&sample-rate=16000
x-amzn-RequestId: RequestId
Strict-Transport-Security: max-age=31536000
sec-websocket-accept: hash-of-the-Sec-WebSocket-Key-header
```

8. 提出您的 WebSocket 串流請求。

建立 WebSocket 連接後，客戶端可以開始發送一系列音頻幀，每個幀都使用[事件流編碼進行編碼](#)。

每個資料框架包含三個標頭並結合原始音訊位元組塊；下表描述這些標頭。

標頭名稱位元組長度	標頭名稱 (字串)	標頭值類型	值字串位元組長度	值字串 (UTF-8)
13	:content-type	7	24	application/octet-stream
11	:event-type	7	10	AudioEvent

標頭名稱位元組長度	標頭名稱 (字串)	標頭值類型	值字串位元組長度	值字串 (UTF-8)
13	:message-type	7	5	事件

9. 若要結束資料串流，請將空的音訊串流區塊傳送到事件串流編碼的訊息中。

回應包含承載中的事件串流編碼的原始位元組。它包含標準序言和下列標頭。

標頭名稱位元組長度	標頭名稱 (字串)	標頭值類型	值字串位元組長度	值字串 (UTF-8)
13	:content-type	7	16	application/json
11	:event-type	7	15	TranscriptEvent
13	:message-type	7	5	事件

您解碼二進位回應時，將獲得包含轉錄結果的 JSON 結構。

### 處理 WebSocket 串流錯誤

如果在處理您的要求時發生例外狀況，Amazon Transcribe 請使用包含事件串流編碼回應的終端機 WebSocket 影格來回應。該回應包含下表中所述的標頭；回應的內文包含描述性錯誤訊息。發送異常響應後，發 Amazon Transcribe 送一個關閉幀。

標頭名稱位元組長度	標頭名稱 (字串)	標頭值類型	值字串位元組長度	值字串 (UTF-8)
13	:content-type	7	16	application/json
15	:exception-type	7	各有不同	各有不同，請參閱以下內容
13	:message-type	7	9	例外狀況

exception-type 標頭包含以下其中一個值。

- **BadRequestException**：建立串流時發生用戶端錯誤，或串流資料時發生錯誤。請確定您的用戶端已準備好接受資料，並再次嘗試請求。
- **InternalFailureException**：Amazon Transcribe 與客戶握手過程中遇到問題。請再次嘗試您的請求。
- **LimitExceededException**：用戶端超過並行串流限制。如需詳細資訊，請參閱[Amazon Transcribe 限制](#)。減少正在轉錄的串流數量。
- **UnrecognizedClientException**：WebSocket 升級要求使用不正確的存取金鑰或秘密金鑰簽署。請確定您正確建立存取金鑰，並再次嘗試請求。

Amazon Transcribe 也可以返回任何常見的服務錯誤。如需清單，請參閱[常見錯誤](#)。

## 事件串流編碼

Amazon Transcribe 使用稱為事件串流編碼的格式來進行串流轉錄。

事件串流編碼會在用戶端與伺服器之間提供雙向通訊。傳送至 Amazon Transcribe 串流服務的資料訊框會以此格式編碼。來自的響應 Amazon Transcribe 也使用此編碼。

每則訊息包含兩個部分：序言和資料。序言包含：

1. 訊息的總位元組長度
2. 所有標頭的合併位元組長度

資料區段包含：

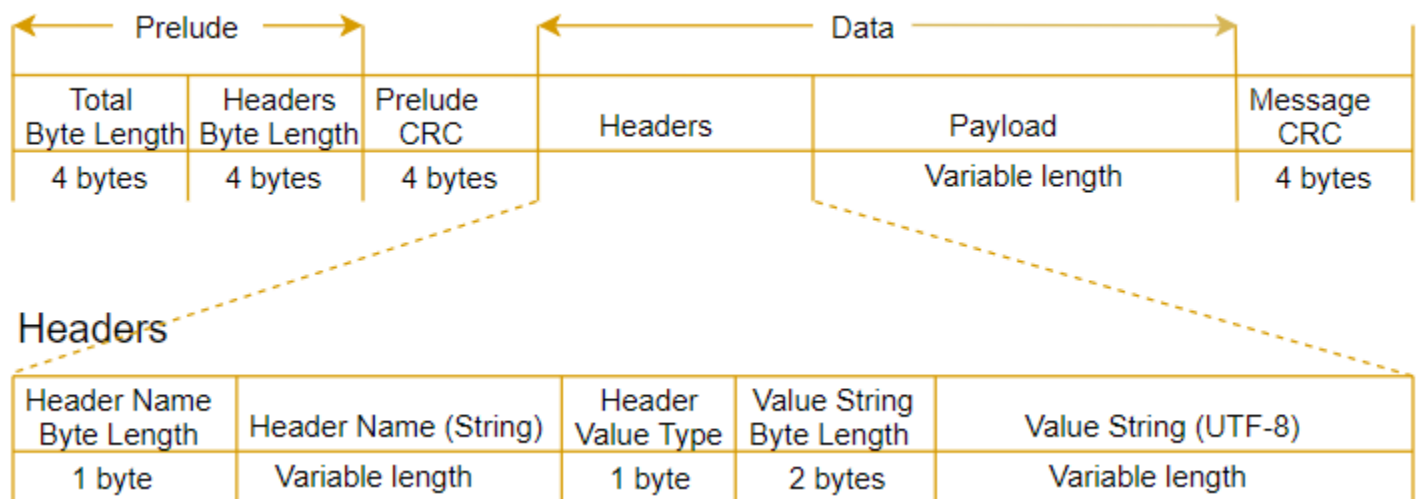
1. 標頭
2. 承載

每個區段以 4 位元組大端序整數循環冗餘校驗 (CRC) 總和結束。訊息 CRC 檢查總和適用於序言區段和資料區段。Amazon Transcribe 使用 CRC32 (通常稱為 GZIP CRC32)，計算兩個 CRC。如需 CRC32 的詳細資訊，請參閱[GZIP 檔案格式規格 4.3 版](#)。

總訊息額外負荷 (包括序言和兩個檢查總和) 為 16 位元組。

下圖顯示構成訊息和標題的元件。每則訊息有多個標頭。





每則訊息都包含下列元件：

- 序言：由兩個 4 位元組欄位組成，總數固定為 8 個位元組。
  - 前 4 個位元組：這是整個訊息的大端序整數位元組長度，包括 4 位元組長度欄位。
  - 次 4 個位元組：這是訊息「標頭」部分的大端序整數位元組長度，不含「標頭」長度欄位本身。
- 序言 CRC：訊息序言部分的 4 位元組 CRC 檢查總和，不含 CRC 本身。序言與訊息 CRC 有獨立的 CRC。這樣 Amazon Transcribe 可以確保可以立即檢測損壞的字節長度信息，而不會導致錯誤，例如緩衝區溢出。
- 標頭：註釋訊息的中繼資料；例如訊息類型、訊息類型和內容類型。訊息有多個標頭，屬於 key:value 配對，其中鍵是 UTF-8 字串。標頭可以任何順序出現在訊息的標頭部分中，而且每個標頭只能出現一次。
- 承載：要轉錄的音訊內容。
- 訊息 CRC：從訊息開頭到檢查總和開頭的 4 位元組 CRC 檢查總和。亦即，訊息中的一切，但 CRC 本身除外。

標題框架是用於流轉錄的授權框架。Amazon Transcribe 使用授權標頭的值作為種子，用於為請求中的數據幀生成授權標題鍵。

每個標頭都包含下列元件；每個框架有多個標頭。

- 標頭名稱位元組長度：標頭名稱的位元組長度。
- 標頭名稱：指出標頭類型的標頭名稱。如需有效值，請參閱下列框架描述。
- 標頭值類型：指出標頭值的數字。下列清單顯示標頭的可能值及其指示內容。
  - 0 – TRUE

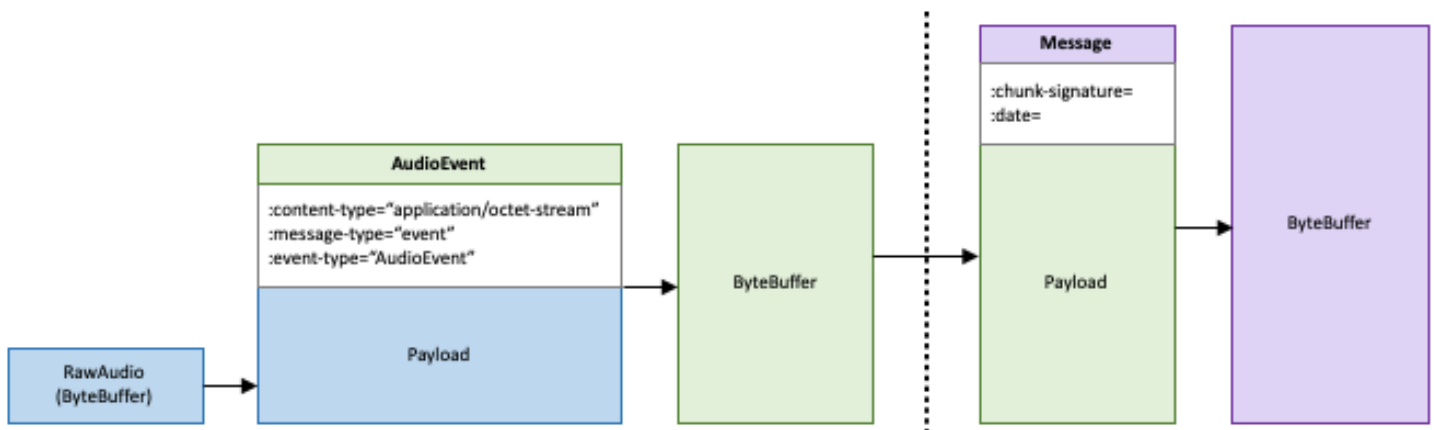
- 1 – FALSE
  - 2 – BYTE
  - 3 – SHORT
  - 4 – INTEGER
  - 5 – LONG
  - 6 – BYTE ARRAY
  - 7 – STRING
  - 8 – TIMESTAMP
  - 9 – UUID
- 數值字串位元組長度：標頭值字串的位元組長度。
  - 標頭值：標頭字串的值。此欄位的有效值取決於標頭類型。請參閱 [設定 HTTP/2 串流](#) 或 [設定串 WebSocket 流](#)，了解詳細資訊。

## 資料框架

每個串流請求都包含一或多個資料框架。建立資料框架的步驟有二：

1. 結合原始音訊資料與中繼資料以建立您的請求的承載。
2. 結合承載與簽章來形成傳送到 Amazon Transcribe 的事件訊息。

下圖顯示此運作方式。



# Job 佇列

使用工作佇列時，您可以提交的轉錄工作要求超過同時處理的數量。如果沒有工作佇列，一旦達到允許的並行請求的配額，您必須等到一個或多個請求完成後，才能提交新的請求。

Job 佇列對於轉錄工作要求而言是選擇性的。通話後分析要求會自動啟用工作佇列。

如果啟用工作佇列，Amazon Transcribe會建立一個佇列，其中包含超出限制的所有要求。請求完成後，系統會立即從您的佇列中提取新的請求並處理。佇列的要求會以 FIFO (先進先出) 順序處理。

您最多可新增 10,000 個任務務務務務務務務務務務務務務務。如果超過此限制，就會收到 `LimitExceededConcurrentJobException` 錯誤訊息。為了維持最佳效能，最多 Amazon Transcribe 只會使用 90% 的配額 (頻寬比率為 0.9) 來處理佇列的工作。請注意，這些是可以在請求時提高。

## Tip

您可以在「[AWS 一般參考](#)」中找到 Amazon Transcribe 資源的預設限制和配額清單。其中一些預設值可以在請求時提高。

如果您啟用工作佇列，但未超過並行要求的配額，則會同時處理所有要求。

## 啟用任務務務務務務

您可以使用 AWS Management Console、AWS CLI、或 AWS SDK 啟用工作佇列；如需範例，請參閱下列範例：

### AWS Management Console

1. 登入 [AWS Management Console](#)。
2. 在功能窗格中，選擇「轉錄工作」，然後選取「建立工作」( 右上角 )。這會開啟 [指定工作詳細資訊] 頁面。
3. 在 [Job 設定] 方塊中，有 [其他設定] 面板。如果展開此面板，您可以選取 [新增至工作佇列] 方塊以啟用工作佇列。

## Specify job details [Info](#)

### Job settings

#### Name

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

#### Language settings

You can transcribe your audio file in a language that you specify or have Amazon Transcribe identify and transcribe it in the predominant language.

**Specific language** [Info](#)

If you know the language spoken in your source audio, choose this option to get the most accurate results. The options available for additional processing vary between languages.

**Automatic language identification** [Info](#)

If you don't know the language spoken in your audio files, choose this option. You have access to fewer options for additional processing than if you choose **Specific language**.

#### Language

Choose the language of the input audio.

#### Model type [Info](#)

Choose the type of model to use for the transcription job.

**General model**

To use a model that is not specialized for a particular use case, choose this option. Configuration options vary between languages.

**Custom language model**

To use a model that you trained for your specific use case, choose this option. This model has fewer configuration options than the general model.

#### ▼ Additional settings

##### Job queue - *optional* [Info](#)

Enables you to submit jobs beyond the limit for concurrent jobs (100). You must specify access permissions to the resources that job queuing uses.

Add to job queue

- 填入您要包含在 [指定工作詳細資訊] 頁面上的任何其他欄位，然後選取 [下一步]。這會帶您前往 [設定工作-選擇性] 頁面。
- 選取 [建立工作] 以執行轉錄工作。

## AWS CLI

此範例使用[start-transcription-job](#)指令和job-execution-settings參數搭配AllowDeferredExecution子參數。請注意，當您AllowDeferredExecution在請求中包含時，您還必須包括DataAccessRoleArn。

如需詳細資訊，請參閱 [StartTranscriptionJob](#) 及 [JobExecutionSettings](#)。

```
aws transcribe start-transcription-job \  
--region us-west-2 \  
--transcription-job-name my-first-transcription-job \  
--media MediaFileUri=s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac \  
--output-bucket-name DOC-EXAMPLE-BUCKET \  
--output-key my-output-files/ \  
--language-code en-US \  
--job-execution-settings  
  AllowDeferredExecution=true,DataAccessRoleArn=arn:aws:iam::111122223333:role/  
ExampleRole
```

以下是使用[start-transcription-job](#)命令的另一個範例，以及啟用佇列的要求主體。

```
aws transcribe start-transcription-job \  
--region us-west-2 \  
--cli-input-json file://my-first-queueing-request.json
```

該文件 my-first-queueing-request.json 包含以下請求主體。

```
{  
  "TranscriptionJobName": "my-first-transcription-job",  
  "Media": {  
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"  
  },  
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",  
  "OutputKey": "my-output-files/",  
  "LanguageCode": "en-US",  
  "JobExecutionSettings": {  
    "AllowDeferredExecution": true,  
    "DataAccessRoleArn": "arn:aws:iam::<111122223333>:role/ExampleRole"  
  }  
}
```

## AWS SDK for Python (Boto3)

這個範例會AWS SDK for Python (Boto3)使用啟用工作佇列，使用AllowDeferredExecution引數來啟用工作佇列。請注意，當您AllowDeferredExecution在請求中包含時，您還必須包括DataAccessRoleArn。如需詳細資訊，請參閱 [StartTranscriptionJob](#) 及 [JobExecutionSettings](#)。

如需使用AWS SDK 的其他範例，包括特定功能、案例和跨服務範例，請參閱本[使用 SDK 進行 Amazon Transcribe 的代碼示例 AWS](#)章。

```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
job_name = "my-first-queueing-request"
job_uri = "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
transcribe.start_transcription_job(
    TranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'DOC-EXAMPLE-BUCKET',
    OutputKey = 'my-output-files/',
    LanguageCode = 'en-US',
    JobExecutionSettings = {
        'AllowDeferredExecution': True,
        'DataAccessRoleArn': 'arn:aws:iam::111122223333:role/ExampleRole'
    }
)

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)
```

您可以透過AWS Management Console或提交[GetTranscriptionJob](#)請求來檢視佇列工作的進度。當工作排入佇列時，就Status是QUEUED。工作開始處理後，狀態會變更為COMPLETED或處理完成FAILED時，狀態會變更為。IN\_PROGRESS

## 標記 資源

標籤是一種自訂中繼資料標籤，您可以將其新增至資源，以便於在搜尋中識別、組織和尋找。標籤由兩個單獨的部分組成：標籤索引鍵和標籤值。這被稱為一個鍵：值對。

標籤鍵通常表示較大的品類，而標籤值則表示該品類的子集。例如，您可以有標籤鍵 = 顏色和標籤值 = 藍色，這將產生鍵：值對Color:Blue。請注意，您可以將標籤的值設為空白字串，但您無法將標籤的值設為 null。忽略標籤值基本上等同於使用空字串。

### Tip

AWS Billing and Cost Management可以使用標籤將您的帳單分為動態類別。例如，如果您新增標籤來代表公司內的不同部門，例如Department:Legal，Department:Sales或AWS可以為您提供每個部門的成本分配。

在 Amazon Transcribe 中，您可以標記下列資源：

- 轉錄工作
- 醫療轉錄工作
- 通話分析通話後轉錄工作
- Custom 自訂
- 自定義醫學詞彙
- 自訂詞彙篩選器
- 自訂語言模型

標籤金鑰最長可包含 128 個字元；標籤值最長可包含 256 個字元；兩者均區分大小寫。Amazon Transcribe每個資源最多可支援 50 個標籤。對於給定資源，每個標籤金鑰必須是唯一的，且只有一個值。請注意，您的標籤不能以開頭，aws:因為會為系統產生的標籤AWS保留此前置詞。您無法新增、修改或刪除aws:\*標籤，也不會計入您的 tags-per-resource 限制。

### 特定於資源標記的 API 操作

[ListTagsForResource](#)、[TagResource](#)、[UntagResource](#)

若要使用標記 API，您必須在請求中包含 Amazon Resource Name (ARN)。ARN 具有這種格式arn:partition:service:region:account-id:resource-type/resource-

id。例如，與轉錄工作相關聯的 ARN 可能如下所示：`arn:aws:transcribe:us-west-2:111122223333:transcription-job/my-transcription-job-name`。

若要進一步了解標記 (包括最佳做法)，請參閱[標記AWS資源](#)。

## 標籤型存取控制

您可以使用標籤來控制您的AWS帳戶。對於以標籤為基礎的存取控制，則在IAM政策的條件元素中提供標籤資訊。您可以使用標籤與其關聯的標籤條件索引鍵，以控制下列項目的存取權：

- Resource: 根據您指派給這些Amazon Transcribe資源的標籤，控制對資源的存取。
  - 使用`aws:ResourceTag/key-name`條件索引鍵，以指定必須連接到該資源的標籤鍵/值組。
- 請求：控制您可以在請求中傳遞哪些標籤。
  - 使用`aws:RequestTag/key-name`條件索引鍵，以指定可以從使用IAM者或角色新增、修改或移除哪些標籤。
- 授權程序：針對授權程序的任何部分，控制以標籤為基礎的存取。
  - 使用`aws:TagKeys/條件索引鍵`，以控制是否可在資源或請求中使用特定標籤鍵，或是否可由委託人使用特定標籤鍵。在這種情況下，此索引鍵值無關緊要。

如需以標籤為基礎的存取控制政策的範例，請參閱[根據標籤檢視轉錄任務](#)。

如需有關以標籤為基礎的存取控制的詳細資訊，請參閱[使用標籤控制對AWS資源的存取](#)。

## 將標籤新增到您的Amazon Transcribe資源

您可以在執行Amazon Transcribe工作之前或之後新增標籤。使用現有的建立 \* 和 Start\* API，您可以在轉錄要求中新增標籤。

您可以使用、或 AWSSDK 新增AWS Management ConsoleAWS CLI、修改或刪除標籤；如需範例，請參閱下列內容：

### AWS Management Console

1. 登入 [AWS Management Console](#)。
2. 在功能窗格中，選擇「轉錄工作」，然後選取「建立工作」（右上角）。這會開啟 [指定工作詳細資訊] 頁面。



- 捲動至 [指定工作詳細資訊] 頁面底部，尋找 [標籤-選擇性] 方塊，然後選取 [新增標籤]。

**Tags - optional**

A tag is a label that you can add to a resource as metadata to help you organize, search, or filter your data. Each tag consists of a key and an option value.

No tags associated with the resource.

You can add up to 50 more tags.

- 輸入「主鍵」欄位的資訊，並選擇性地輸入「值」欄位。

**Tags - optional**

A tag is a label that you can add to a resource as metadata to help you organize, search, or filter your data. Each tag consists of a key and an option value.

Key

Value - optional

You can add up to 49 more tags.

- 填入您要包含在 [指定工作詳細資訊] 頁面上的任何其他欄位，然後選取 [下一步]。這會帶您前往 [設定工作-選擇性] 頁面。

選取 [建立工作] 以執行轉錄工作。

- 您可以瀏覽至 [轉錄工作] 頁面、選取轉錄工作，然後捲動至該工作資訊頁底部，檢視與轉錄工作相關聯的標籤。如果您想要編輯標籤，可以選取 [管理標籤]。

**Tags (2)**

Key	Value
color	blue

## AWS CLI

此範例使用指[start-transcription-job](#)令和Tags參數。如需詳細資訊，請參閱[StartTranscriptionJob](#)及[Tag](#)。

```
aws transcribe start-transcription-job \  
--region us-west-2 \  
--transcription-job-name my-first-transcription-job \  
--media MediaFileUri=s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac \  
--output-bucket-name DOC-EXAMPLE-BUCKET \  
--output-key my-output-files/ \  
--language-code en-US \  
--tags Key=color,Value=blue Key=shape,Value=square
```

這是另一個使用[start-transcription-job](#)命令的示例，以及向該作業添加標籤的請求主體。

```
aws transcribe start-transcription-job \  
--region us-west-2 \  
--cli-input-json file://filepath/my-first-tagging-job.json
```

該文件 *my-first-tagging-job.json* 包含以下請求主體。

```
{  
  "TranscriptionJobName": "my-first-transcription-job",  
  "Media": {  
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"  
  },  
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",  
  "OutputKey": "my-output-files/",  
  "LanguageCode": "en-US",  
  "Tags": [  
    {  
      "Key": "color",  
      "Value": "blue"  
    },  
    {  
      "Key": "shape",  
      "Value": "square"  
    }  
  ]  
}
```

## AWS SDK for Python (Boto3)

下列範例會AWS SDK for Python (Boto3)使用[啟動方法的Tags引數來新增](#)標籤。如需詳細資訊，請參閱 [StartTranscriptionJob](#) 及 [Tag](#)。

如需使用AWS SDK 的其他範例，包括特定功能、案例和跨服務範例，請參閱本[使用 SDK 進行 Amazon Transcribe 的代碼示例 AWS](#)章。

```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
job_name = "my-first-transcription-job"
job_uri = "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
transcribe.start_transcription_job(
    TranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'DOC-EXAMPLE-BUCKET',
    OutputKey = 'my-output-files/',
    LanguageCode = 'en-US',
    Tags = [
        {
            'Key': 'color',
            'Value': 'blue'
        }
    ]
)

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)
```

## 分隔發言者 (日記)

使用揚聲器日記，您可以區分轉錄輸出中的不同揚聲器。Amazon Transcribe 可以區分最多 30 個獨特的揚聲器，並以獨特的價值 ( spk\_0 通過 spk\_9 ) 標記每個獨特揚聲器的文本。

除了 [標準文字記錄區段](#) (transcripts 和 items) 之外，啟用發言者分隔的請求還包括一個 speaker\_labels 區段。本節按發言者分組，包含每個表達方式的資訊，包括發言者標籤和時間戳記。

```
"speaker_labels": {
  "channel_label": "ch_0",
  "speakers": 2,
  "segments": [
    {
      "start_time": "4.87",
      "speaker_label": "spk_0",
      "end_time": "6.88",
      "items": [
        {
          "start_time": "4.87",
          "speaker_label": "spk_0",
          "end_time": "5.02"
        },
        ...
      ],
      {
        "start_time": "8.49",
        "speaker_label": "spk_1",
        "end_time": "9.24",
        "items": [
          {
            "start_time": "8.49",
            "speaker_label": "spk_1",
            "end_time": "8.88"
          },
        ],
      },
    ],
  },
}
```

要檢視具有發言者分隔功能的完整範例文字記錄 (適用於兩位發言者)，請參閱 [例如日記輸出 \(批次\)](#)。

## 在批次轉錄中對發言者進行分隔

若要在批次轉錄中分隔發言者，請參閱下列範例：

## AWS Management Console

1. 登入 [AWS Management Console](#)。
2. 在導覽窗格中，選擇轉錄作業，然後選擇建立作業(右上角)。這會開啟指定作業詳細資訊頁面。

**Specify job details** [Info](#)

**Job settings**

**Name**

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

**Model type** [Info](#)

Choose the type of model to use for the transcription job.

**General model**  
To use a model that is not specialized for a particular use case, choose this option. Configuration options vary between languages.

**Custom language model**  
To use a model that you trained for your specific use case, choose this option. This model has fewer configuration options than the general model.

**Language settings**

You can transcribe your audio file in a language that you specify or have Amazon Transcribe identify and transcribe it in the predominant language.

**Specific language** [Info](#)  
If you know the language spoken in your source audio, choose this option to get the most accurate results. The options available for additional processing vary between languages.

**Automatic language identification** [Info](#)  
If you don't know the language spoken in your audio files, choose this option. You have access to fewer options for additional processing than if you choose **Specific language**.

**Language**

Choose the language of the input audio.

▶ **Additional settings**

3. 填寫您要包含在指定作業詳細資訊頁面上的任何欄位，然後選擇下一步。這會引導您前往設定工作-選擇性頁面。

在音訊設定面板中，選擇發音者分隔(在「音訊識別類型」標題下)。您可以選擇性指定要分隔的發音者數量，最多 10 個。

### Audio settings

**Audio identification** [Info](#)  
Choose to split multi-channel audio into separate channels for transcription, or partition speakers in the input audio.

Audio identification type

Channel identification

Speaker partitioning

Maximum number of speakers  
Providing the number of speakers can increase the accuracy of your results.

The maximum number of speakers is 10.

---

**Alternative results** [Info](#)  
Enable to view more transcription results

4. 選擇建立作業以執行轉錄作業。

## AWS CLI

此範例使用 [start-transcription-job](#)。如需詳細資訊，請參閱 [StartTranscriptionJob](#)。

```
aws transcribe start-transcription-job \  
--region us-west-2 \  
--transcription-job-name my-first-transcription-job \  
--media MediaFileUri=s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac \  
--output-bucket-name DOC-EXAMPLE-BUCKET \  
--output-key my-output-files/ \  
--language-code en-US \  
--show-speaker-labels TRUE \  
--max-speaker-labels 3
```

下面是使用 [start-transcription-job](#) 命令的另一個示例，以及一個請求主體，可以使用該作業進行發言者分區。

```
aws transcribe start-transcription-job \  
--region us-west-2 \  

```

```
--cli-input-json file://my-first-transcription-job.json
```

該文件 my-first-transcription-job.json 包含以下請求主體。

```
{
  "TranscriptionJobName": "my-first-transcription-job",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
  },
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",
  "OutputKey": "my-output-files/",
  "LanguageCode": "en-US",
  "ShowSpeakerLabels": 'TRUE',
  "MaxSpeakerLabels": 3
}
```

## AWS SDK for Python (Boto3)

此範例會使用 AWS SDK for Python (Boto3) 來識別使用[開始轉譯](#)工作方法的通道。如需詳細資訊，請參閱[StartTranscriptionJob](#)。

```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
job_name = "my-first-transcription-job"
job_uri = "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
transcribe.start_transcription_job(
    TranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'DOC-EXAMPLE-BUCKET',
    OutputKey = 'my-output-files/',
    LanguageCode = 'en-US',
    Settings = {
        'ShowSpeakerLabels': True,
        'MaxSpeakerLabels': 3
    }
)

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)
```

## 在串流轉錄中分隔發言者

若要在串流轉錄中分隔發言者，請參閱下列範例：

### 串流轉錄

1. 登入 [AWS Management Console](#)。
2. 在導覽窗格中，選擇即時轉錄。向下捲動至音訊設定，如果此欄位已最小化，請展開此欄位。

**Real-time transcription** [Info](#)

See how Amazon Transcribe creates a text copy of speech in real time. Choose **Start streaming** and talk.

**Transcription** [Download full transcript](#) [Start streaming](#)

Transcription output Current language: English, US

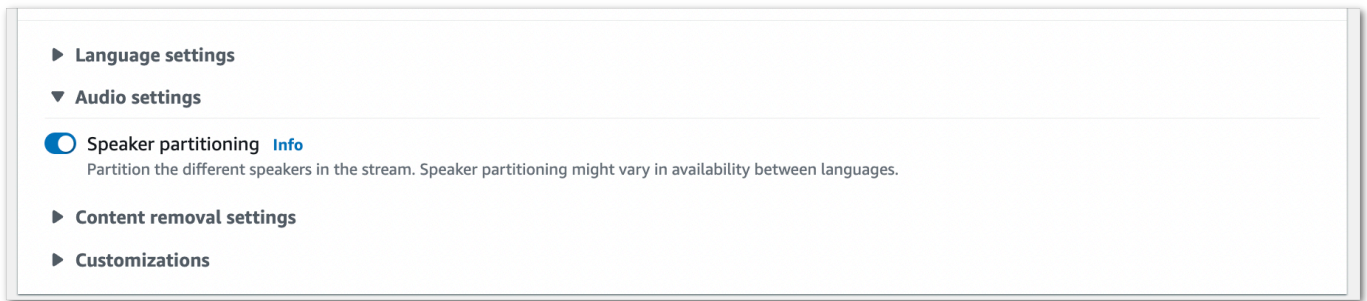
Choose **Start streaming** to begin a real-time transcription of what you speak into your microphone

00:00 of 15:00 min audio stream

- ▶ **Language settings**
- ▼ **Audio settings**
- Speaker partitioning** [Info](#)  
Partition the different speakers in the stream. Speaker partitioning might vary in availability between languages.
- ▶ **Content removal settings**
- ▶ **Customizations**

3. 開啟 發言者分隔。





4. 您現在已準備好轉錄串流。選擇開始串流並開始說話。若要結束聽寫，選擇停止串流。

## HTTP/2 串流

此範例會建立 HTTP/2 要求，以分隔轉錄輸出中的發言者。如需搭配使用 HTTP/2 串流的詳細資訊 Amazon Transcribe，請參閱 [設定 HTTP/2 串流](#)。如需有關特定參數和標題的更多詳細資訊 Amazon Transcribe，請參閱 [StartStreamTranscription](#)。

```
POST /stream-transcription HTTP/2
host: transcribestreaming.us-west-2.amazonaws.com
X-Amz-Target: com.amazonaws.transcribe.Transcribe.StartStreamTranscription
Content-Type: application/vnd.amazon.eventstream
X-Amz-Content-Sha256: string
X-Amz-Date: 20220208T235959Z
Authorization: AWS4-HMAC-SHA256 Credential=access-key/20220208/us-west-2/transcribe/
aws4_request, SignedHeaders=content-type;host;x-amz-content-sha256;x-amz-date;x-amz-
target;x-amz-security-token, Signature=string
x-amzn-transcribe-language-code: en-US
x-amzn-transcribe-media-encoding: flac
x-amzn-transcribe-sample-rate: 16000
x-amzn-transcribe-show-speaker-label: true
transfer-encoding: chunked
```

您可以在 API [參考](#)中找到參數定義；所有 [AWS API](#) 作業的共用參數會列在「[一般參數](#)」區段中。

## WebSocket 流

此範例會建立預先簽署的 URL，以分隔轉錄輸出中的發言者。已加入分行符號以提高可讀性。如需搭配使用 WebSocket 串流的詳細資訊 Amazon Transcribe，請參閱 [設定串流 WebSocket 流](#)。如需參數詳細資訊，請參閱 [StartStreamTranscription](#)。

```
GET wss://transcribestreaming.us-west-2.amazonaws.com:8443/stream-transcription-
websocket?
```

```
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIAIOSFODNN7EXAMPLE%2F20220208%2Fus-
west-2%2Ftranscribe%2Faws4_request
&X-Amz-Date=20220208T235959Z
&X-Amz-Expires=300
&X-Amz-Security-Token=security-token
&X-Amz-Signature=string
&X-Amz-SignedHeaders=content-type%3Bhost%3Bx-amz-date
&language-code=en-US
&specialty=PRIMARYCARE
&type=DICTATION
&media-encoding=flac
&sample-rate=16000
&show-speaker-label=true
```

您可以在 [API 參考](#) 中找到參數定義；所有 [AWS API](#) 作業的共用參數會列在「[一般參數](#)」區段中。

## 例如日記輸出 ( 批次 )

以下是啟用日記的批量轉錄的輸出示例。

```
{
  "jobName": "my-first-transcription-job",
  "accountId": "111122223333",
  "results": {
    "transcripts": [
      {
        "transcript": "I've been on hold for an hour. Sorry about that."
      }
    ],
    "speaker_labels": {
      "channel_label": "ch_0",
      "speakers": 2,
      "segments": [
        {
          "start_time": "4.87",
          "speaker_label": "spk_0",
          "end_time": "6.88",
          "items": [
            {
              "start_time": "4.87",
              "speaker_label": "spk_0",
              "end_time": "5.02"
            }
          ]
        }
      ]
    }
  }
}
```

```
    },
    {
      "start_time": "5.02",
      "speaker_label": "spk_0",
      "end_time": "5.17"
    },
    {
      "start_time": "5.17",
      "speaker_label": "spk_0",
      "end_time": "5.29"
    },
    {
      "start_time": "5.29",
      "speaker_label": "spk_0",
      "end_time": "5.64"
    },
    {
      "start_time": "5.64",
      "speaker_label": "spk_0",
      "end_time": "5.84"
    },
    {
      "start_time": "6.11",
      "speaker_label": "spk_0",
      "end_time": "6.26"
    },
    {
      "start_time": "6.26",
      "speaker_label": "spk_0",
      "end_time": "6.88"
    }
  ]
},
{
  "start_time": "8.49",
  "speaker_label": "spk_1",
  "end_time": "9.24",
  "items": [
    {
      "start_time": "8.49",
      "speaker_label": "spk_1",
      "end_time": "8.88"
    },
  ],
}
```

```
        "start_time": "8.88",
        "speaker_label": "spk_1",
        "end_time": "9.05"
    },
    {
        "start_time": "9.05",
        "speaker_label": "spk_1",
        "end_time": "9.24"
    }
]
}
]
},
"items": [
    {
        "start_time": "4.87",
        "speaker_label": "spk_0",
        "end_time": "5.02",
        "alternatives": [
            {
                "confidence": "1.0",
                "content": "I've"
            }
        ],
        "type": "pronunciation"
    },
    {
        "start_time": "5.02",
        "speaker_label": "spk_0",
        "end_time": "5.17",
        "alternatives": [
            {
                "confidence": "1.0",
                "content": "been"
            }
        ],
        "type": "pronunciation"
    },
    {
        "start_time": "5.17",
        "speaker_label": "spk_0",
        "end_time": "5.29",
        "alternatives": [
            {
```

```
        "confidence": "1.0",
        "content": "on"
    }
],
"type": "pronunciation"
},
{
    "start_time": "5.29",
    "speaker_label": "spk_0",
    "end_time": "5.64",
    "alternatives": [
        {
            "confidence": "1.0",
            "content": "hold"
        }
    ],
    "type": "pronunciation"
},
{
    "start_time": "5.64",
    "speaker_label": "spk_0",
    "end_time": "5.84",
    "alternatives": [
        {
            "confidence": "1.0",
            "content": "for"
        }
    ],
    "type": "pronunciation"
},
{
    "start_time": "6.11",
    "speaker_label": "spk_0",
    "end_time": "6.26",
    "alternatives": [
        {
            "confidence": "1.0",
            "content": "an"
        }
    ],
    "type": "pronunciation"
},
{
    "start_time": "6.26",
```

```
    "speaker_label": "spk_0",
    "end_time": "6.88",
    "alternatives": [
      {
        "confidence": "1.0",
        "content": "hour"
      }
    ],
    "type": "pronunciation"
  },
  {
    "speaker_label": "spk_0",
    "alternatives": [
      {
        "confidence": "0.0",
        "content": "."
      }
    ],
    "type": "punctuation"
  },
  {
    "start_time": "8.49",
    "speaker_label": "spk_1",
    "end_time": "8.88",
    "alternatives": [
      {
        "confidence": "1.0",
        "content": "Sorry"
      }
    ],
    "type": "pronunciation"
  },
  {
    "start_time": "8.88",
    "speaker_label": "spk_1",
    "end_time": "9.05",
    "alternatives": [
      {
        "confidence": "0.902",
        "content": "about"
      }
    ],
    "type": "pronunciation"
  },
}
```

```
    {
      "start_time": "9.05",
      "speaker_label": "spk_1",
      "end_time": "9.24",
      "alternatives": [
        {
          "confidence": "1.0",
          "content": "that"
        }
      ],
      "type": "pronunciation"
    },
    {
      "speaker_label": "spk_1",
      "alternatives": [
        {
          "confidence": "0.0",
          "content": "."
        }
      ],
      "type": "punctuation"
    }
  ]
},
"status": "COMPLETED"
}
```

## 轉錄多聲道音訊

如果您的音訊有兩個聲道，您可以使用頻道識別來分別從每個頻道轉錄語音。Amazon Transcribe目前不支援超過兩個頻道的音訊。

在您的成績單中，頻道會分配標籤ch\_0和ch\_1。

除了[標準成績單區段](#) (transcripts和items) 之外，啟用頻道識別的請求還包括一個channel\_labels區段。此區段包含每個語音符號或標點符號，並依頻道分組，以及其相關的頻道標籤、時間戳記和可信度分數。

```
"channel_labels": {
  "channels": [
    {
      "channel_label": "ch_0",
      "items": [
        {
          "channel_label": "ch_0",
          "start_time": "4.86",
          "end_time": "5.01",
          "alternatives": [
            {
              "confidence": "1.0",
              "content": "I've"
            }
          ],
          "type": "pronunciation"
        },
        ...
      ],
      "channel_label": "ch_1",
      "items": [
        {
          "channel_label": "ch_1",
          "start_time": "8.5",
          "end_time": "8.89",
          "alternatives": [
            {
              "confidence": "1.0",
              "content": "Sorry"
            }
          ],
          "type": "pronunciation"
        }
      ]
    }
  ]
}
```



```
    },  
    ...  
    "number_of_channels": 2  
  },
```

請注意，如果某個頻道上的使用者與不同頻道上的使用者同時說話，則每個頻道的時間戳記會重疊，而個人彼此交談。

若要檢視具有通道識別功能的完整範例記錄，請參閱[示例通道識別輸出 \(批處理\)](#)。

## 在批次轉錄中使用頻道識別

若要識別批次轉錄中的頻道，您可以使用AWS Management Console、AWS CLI、或 AWSSDK；請參閱下列範例：

### AWS Management Console

1. 登入 [AWS Management Console](#)。
2. 在功能窗格中，選擇「轉錄工作」，然後選取「建立工作」（右上角）。這會開啟 [指定工作詳細資訊] 頁面。

## Specify job details [Info](#)

### Job settings

**Name**

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

**Model type [Info](#)**  
Choose the type of model to use for the transcription job.

**General model**  
To use a model that is not specialized for a particular use case, choose this option. Configuration options vary between languages.

**Custom language model**  
To use a model that you trained for your specific use case, choose this option. This model has fewer configuration options than the general model.

**Language settings**  
You can transcribe your audio file in a language that you specify or have Amazon Transcribe Identify and transcribe it in the predominant language.

**Specific language [Info](#)**  
If you know the language spoken in your source audio, choose this option to get the most accurate results. The options available for additional processing vary between languages.

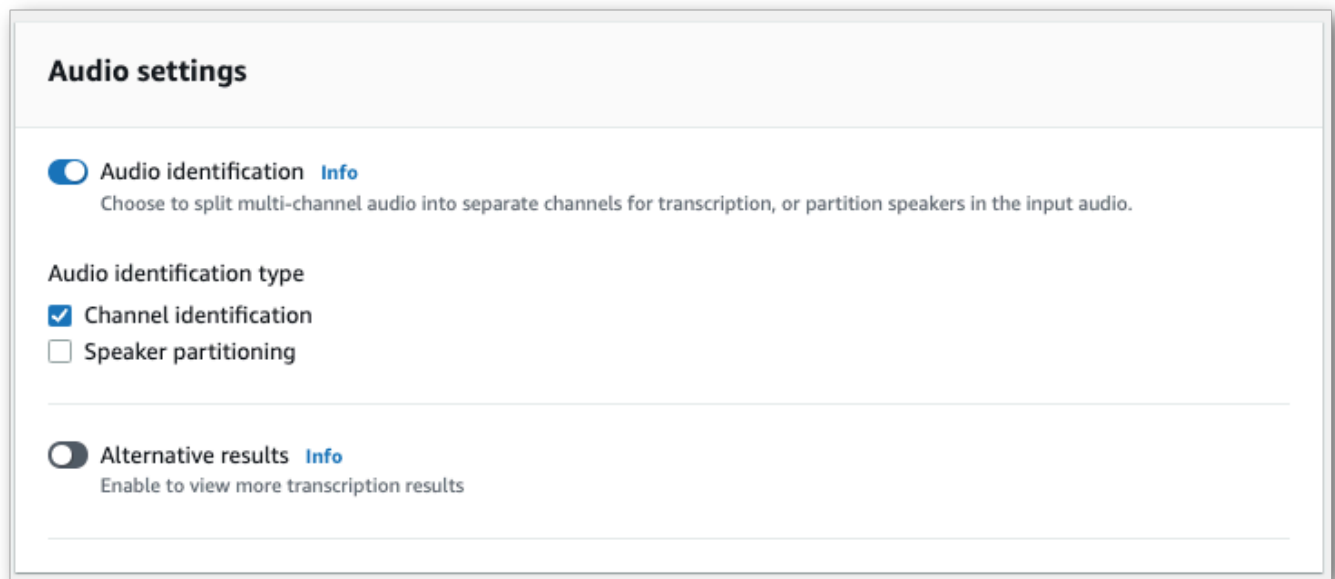
**Automatic language identification [Info](#)**  
If you don't know the language spoken in your audio files, choose this option. You have access to fewer options for additional processing than if you choose **Specific language**.

**Language**  
Choose the language of the input audio.

▶ **Additional settings**

3. 填寫您要包含在 [指定工作詳細資訊] 頁面上的任何欄位，然後選取 [下一步]。這會帶您前往 [設定工作-選擇性] 頁面。

在「音訊設定」面板中，選取「聲道識別」(在「音訊識別類型」標題下)。



4. 選取 [建立工作] 以執行轉錄工作。

## AWS CLI

此範例使用 [start-transcription-job](#)。如需詳細資訊，請參閱[StartTranscriptionJob](#)。

```
aws transcribe start-transcription-job \  
--region us-west-2 \  
--transcription-job-name my-first-transcription-job \  
--media MediaFileUri=s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac \  
--output-bucket-name DOC-EXAMPLE-BUCKET \  
--output-key my-output-files/ \  
--language-code en-US \  
--settings ChannelIdentification=true
```

以下是使用[start-transcription-job](#)命令的另一個範例，以及可透過該工作啟用通道識別的要求主體。

```
aws transcribe start-transcription-job \  
--region us-west-2 \  
--cli-input-json file://my-first-transcription-job.json
```

該文件 *my-first-transcription-job.json* 包含以下請求主體。

```
{
```

```
"TranscriptionJobName": "my-first-transcription-job",
"Media": {
  "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
},
"OutputBucketName": "DOC-EXAMPLE-BUCKET",
"OutputKey": "my-output-files/",
"LanguageCode": "en-US",
"Settings": {
  "ChannelIdentification": true
}
}
```

## AWS SDK for Python (Boto3)

此範例會使用AWS SDK for Python (Boto3)來識別使用[開始轉譯工作](#)方法的通道。如需詳細資訊，請參閱[StartTranscriptionJob](#)。

```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
job_name = "my-first-transcription-job"
job_uri = "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
transcribe.start_transcription_job(
    TranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'DOC-EXAMPLE-BUCKET',
    OutputKey = 'my-output-files/',
    LanguageCode = 'en-US',
    Settings = {
        'ChannelIdentification': True
    }
)

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)
```

## 在串流轉錄中使用頻道識別

若要識別串流轉錄中的頻道，您可以使用 HTTP/2 或 WebSockets；如需範例，請參閱下列內容：

### HTTP/2 流 P/2

此範例會建立 HTTP/2 要求，以分隔轉錄輸出中的通道。如需搭配使用 HTTP/2 串流的詳細資訊 Amazon Transcribe，請參閱[設定 HTTP/2 串流](#)。如需有關特定參數和標題的更多詳細資訊 Amazon Transcribe，請參閱[StartStreamTranscription](#)。

```
POST /stream-transcription HTTP/2
host: transcribestreaming.us-west-2.amazonaws.com
X-Amz-Target: com.amazonaws.transcribe.Transcribe.StartStreamTranscription
Content-Type: application/vnd.amazon.eventstream
X-Amz-Content-Sha256: string
X-Amz-Date: 20220208T235959Z
Authorization: AWS4-HMAC-SHA256 Credential=access-key/20220208/us-west-2/transcribe/
aws4_request, SignedHeaders=content-type;host;x-amz-content-sha256;x-amz-date;x-amz-
target;x-amz-security-token, Signature=string
x-amzn-transcribe-language-code: en-US
x-amzn-transcribe-media-encoding: flac
x-amzn-transcribe-sample-rate: 16000
x-amzn-channel-identification: TRUE
transfer-encoding: chunked
```

您可以在 API [參考中找到參數定義](#)；所有 AWS API 作業的共用參數會列在「[一般參數](#)」區段中。

### WebSocket 流

此範例會建立預先簽署的 URL，以分隔轉錄輸出中的頻道。已加入分行符號以提高可讀性。如需搭配使用 WebSocket 串流的詳細資訊 Amazon Transcribe，請參閱[設定串 WebSocket 流](#)。如需參數的詳細資訊，請參閱[StartStreamTranscription](#)。

```
GET wss://transcribestreaming.us-west-2.amazonaws.com:8443/stream-transcription-
websocket?
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIAIOSFODNN7EXAMPLE%2F20220208%2Fus-
west-2%2Ftranscribe%2Faws4_request
&X-Amz-Date=20220208T235959Z
&X-Amz-Expires=300
&X-Amz-Security-Token=security-token
&X-Amz-Signature=string
```

```
&X-Amz-SignedHeaders=content-type%3Bhost%3Bx-amz-date
&language-code=en-US
&specialty=PRIMARYCARE
&type=DICTIONARY
&media-encoding=flac
&sample-rate=16000
&channel-identification=TRUE
```

您可以在 [API 參考中找到參數定義](#)；所有 AWS API 作業的共用參數會列在「[一般參數](#)」區段中。

## 示例通道識別輸出 ( 批處理 )

以下是啟用通道識別的批次轉錄的輸出範例。

```
{
  "jobName": "my-first-transcription-job",
  "accountId": "111122223333",
  "results": {
    "transcripts": [
      {
        "transcript": "I've been on hold for an hour. Sorry about that."
      }
    ],
    "channel_labels": {
      "channels": [
        {
          "channel_label": "ch_0",
          "items": [
            {
              "channel_label": "ch_0",
              "start_time": "4.86",
              "end_time": "5.01",
              "alternatives": [
                {
                  "confidence": "1.0",
                  "content": "I've"
                }
              ],
              "type": "pronunciation"
            },
            {
              "channel_label": "ch_0",
              "start_time": "5.01",
```

```
        "end_time": "5.16",
        "alternatives": [
            {
                "confidence": "1.0",
                "content": "been"
            }
        ],
        "type": "pronunciation"
    },
    {
        "channel_label": "ch_0",
        "start_time": "5.16",
        "end_time": "5.28",
        "alternatives": [
            {
                "confidence": "1.0",
                "content": "on"
            }
        ],
        "type": "pronunciation"
    },
    {
        "channel_label": "ch_0",
        "start_time": "5.28",
        "end_time": "5.62",
        "alternatives": [
            {
                "confidence": "1.0",
                "content": "hold"
            }
        ],
        "type": "pronunciation"
    },
    {
        "channel_label": "ch_0",
        "start_time": "5.62",
        "end_time": "5.83",
        "alternatives": [
            {
                "confidence": "1.0",
                "content": "for"
            }
        ],
        "type": "pronunciation"
    }
```

```
    },
    {
      "channel_label": "ch_0",
      "start_time": "6.1",
      "end_time": "6.25",
      "alternatives": [
        {
          "confidence": "1.0",
          "content": "an"
        }
      ],
      "type": "pronunciation"
    },
    {
      "channel_label": "ch_0",
      "start_time": "6.25",
      "end_time": "6.87",
      "alternatives": [
        {
          "confidence": "1.0",
          "content": "hour"
        }
      ],
      "type": "pronunciation"
    },
    {
      "channel_label": "ch_0",
      "language_code": "en-US",
      "alternatives": [
        {
          "confidence": "0.0",
          "content": "."
        }
      ],
      "type": "punctuation"
    }
  ]
},
{
  "channel_label": "ch_1",
  "items": [
    {
      "channel_label": "ch_1",
      "start_time": "8.5",
```



```
        "end_time": "8.89",
        "alternatives": [
            {
                "confidence": "1.0",
                "content": "Sorry"
            }
        ],
        "type": "pronunciation"
    },
    {
        "channel_label": "ch_1",
        "start_time": "8.89",
        "end_time": "9.06",
        "alternatives": [
            {
                "confidence": "0.9176",
                "content": "about"
            }
        ],
        "type": "pronunciation"
    },
    {
        "channel_label": "ch_1",
        "start_time": "9.06",
        "end_time": "9.25",
        "alternatives": [
            {
                "confidence": "1.0",
                "content": "that"
            }
        ],
        "type": "pronunciation"
    },
    {
        "channel_label": "ch_1",
        "alternatives": [
            {
                "confidence": "0.0",
                "content": "."
            }
        ],
        "type": "punctuation"
    }
]
```

```
    }
  ],
  "number_of_channels": 2
},
"items": [
  {
    "channel_label": "ch_0",
    "start_time": "4.86",
    "end_time": "5.01",
    "alternatives": [
      {
        "confidence": "1.0",
        "content": "I've"
      }
    ],
    "type": "pronunciation"
  },
  {
    "channel_label": "ch_0",
    "start_time": "5.01",
    "end_time": "5.16",
    "alternatives": [
      {
        "confidence": "1.0",
        "content": "been"
      }
    ],
    "type": "pronunciation"
  },
  {
    "channel_label": "ch_0",
    "start_time": "5.16",
    "end_time": "5.28",
    "alternatives": [
      {
        "confidence": "1.0",
        "content": "on"
      }
    ],
    "type": "pronunciation"
  },
  {
    "channel_label": "ch_0",
    "start_time": "5.28",
```

```
        "end_time": "5.62",
        "alternatives": [
            {
                "confidence": "1.0",
                "content": "hold"
            }
        ],
        "type": "pronunciation"
    },
    {
        "channel_label": "ch_0",
        "start_time": "5.62",
        "end_time": "5.83",
        "alternatives": [
            {
                "confidence": "1.0",
                "content": "for"
            }
        ],
        "type": "pronunciation"
    },
    {
        "channel_label": "ch_0",
        "start_time": "6.1",
        "end_time": "6.25",
        "alternatives": [
            {
                "confidence": "1.0",
                "content": "an"
            }
        ],
        "type": "pronunciation"
    },
    {
        "channel_label": "ch_0",
        "start_time": "6.25",
        "end_time": "6.87",
        "alternatives": [
            {
                "confidence": "1.0",
                "content": "hour"
            }
        ],
        "type": "pronunciation"
    }
```

```
    },
    {
      "channel_label": "ch_0",
      "alternatives": [
        {
          "confidence": "0.0",
          "content": "."
        }
      ],
      "type": "punctuation"
    },
    {
      "channel_label": "ch_1",
      "start_time": "8.5",
      "end_time": "8.89",
      "alternatives": [
        {
          "confidence": "1.0",
          "content": "Sorry"
        }
      ],
      "type": "pronunciation"
    },
    {
      "channel_label": "ch_1",
      "start_time": "8.89",
      "end_time": "9.06",
      "alternatives": [
        {
          "confidence": "0.9176",
          "content": "about"
        }
      ],
      "type": "pronunciation"
    },
    {
      "channel_label": "ch_1",
      "start_time": "9.06",
      "end_time": "9.25",
      "alternatives": [
        {
          "confidence": "1.0",
          "content": "that"
        }
      ]
    }
  ]
}
```

```
    ],
    "type": "pronunciation"
  },
  {
    "channel_label": "ch_1",
    "alternatives": [
      {
        "confidence": "0.0",
        "content": "."
      }
    ],
    "type": "punctuation"
  }
]
},
"status": "COMPLETED"
}
```

## 識別媒體中的優勢語言

Amazon Transcribe 可以自動識別媒體中所使用的語言，而無需指定語言代碼。

[批次語言識別](#) 可以識別媒體檔案中使用的優勢語言，或者，如果您的媒體包含多種語言，則可以識別所有使用的語言。為了提高語言識別的準確性，您可以選擇性提供您認為可能存在於媒體中的兩種或多種語言的清單。

[串流語言識別](#) 可在每個聲道識別一種語言 (最多支援兩個聲道)，或者，如果您的串流包含多種語言，則可以識別所有使用的語言。串流請求必須至少在請求中包含兩個額外的語言選項。提供語言選項允許更快的語言識別。Amazon Transcribe 識別語言的速度越快，串流的前幾秒鐘內資料遺失的變化就越少。

### Important

批次和串流轉錄支援不同的語言。請參閱[支援語言表格](#)中的資料輸入欄，了解詳細資訊。請注意，目前不支援瑞典文和越南文的語言識別。

若要了解使用語言識別的監控和事件的詳細資訊，請參閱[語言識別事件](#)。

## 使用批次轉錄作業進行語言識別

使用批次語言識別功能，自動識別媒體檔案中的語言。

如果您的媒體僅包含一種語言，您可以啟用[單一語言識別](#)功能，識別媒體檔案中所使用的優勢語言，並僅使用此語言建立文字記錄。

如果您的媒體包含一種以上的語言，您可以啟用[多語言識別](#)功能，識別媒體檔案中所使用的所有語言，並使用每種識別的語言建立您的文字記錄。請注意，會產生多語言文字記錄。您可以使用其他服務 Amazon Translate，例如翻譯成績單。

請參閱[支援的語言](#)表，了解支援語言和相關語言代碼的完整清單。

為了獲得最佳結果，請確保您的媒體檔案至少包含 30 秒的語音。

如需有關 AWS Management Console、AWS CLI 和 AWS Python SDK 的使用範例，請參閱[使用語言識別與批次轉錄](#)。

## 識別多語言音訊中的語言

多語言識別功能適用於多語言媒體檔案，並提供可反映媒體中所有[支援語言](#)的文字記錄。這代表如果發言者在對話中改變語言，或每個參與者說的是不同的語言，則您的轉錄輸出會正確偵測並轉錄每種語言。例如，如果您的媒體包含在美國英文 (en-US) 和印地文 (hi-IN) 之間交替使用的雙語使用者，則多語言識別可以識別並將美國英文轉錄為 en-US，並將口語印地文轉錄為 hi-IN。

這與單一語言識別不同，只使用一種優勢語言以建立轉錄。在這種情況下，任何不是優勢語言的口語語言都會被錯誤轉錄。

### Note

多語言識別目前不支援修訂和自訂語言模型。

### Note

以下語言目前支援多種語言識別：en-AB、en-au、en-GB、EN-IE、恩-英、恩-紐西蘭、en-WL、恩 ZA、ES-ES、ES-US、FR-FR、zh-CN、日-tw、日-TW、PT-PT、德-DE、AF-ZA、日-AE、日-tw、日-tw、日-TW、PT-PT、德-DE、AF-ZA、阿聯、高-DK、日-DK、日-英、日、阿聯、阿聯、阿聯、阿聯、阿聯、阿爾-DK、阿聯、阿爾-TW、PT-PT、德-DE-IR，它，日本，日本，KO-KR，MS-我的，NL-NL，汝汝，他在，德英，TH-日，TR-TR

多語言文字記錄提供偵測到的語言摘要，以及媒體中每種語言的總使用時間。範例如下：

```
"results": {
  "transcripts": [
    {
      "transcript": "welcome to Amazon transcribe. ## ## ##### ### #### ####
## #### ### #####"
    }
  ],
  ...
  "language_codes": [
    {
      "language_code": "en-US",
```

```
        "duration_in_seconds": 2.45
    },
    {
        "language_code": "hi-IN",
        "duration_in_seconds": 5.325
    },
    {
        "language_code": "ja-JP",
        "duration_in_seconds": 4.15
    }
]
}
```

## 提高語言識別準確性

使用語言識別功能，您可以選擇包含您認為可能存在於媒體中的語言清單。包含語言選項 (LanguageOptions) 限制只能 Amazon Transcribe 使用您在將音訊與正確語言進行比對時所指定的語言，這樣可以加快語言識別速度，並提高與指定正確語言方言相關的準確性。

如果您選擇包含語言代碼，則必須包含至少兩個。您可以包含的語言代碼數量沒有限制，但我們建議您使用二到五個，以達到最佳的效率和準確性。

### Note

如果您在要求中包含語言代碼，且您提供的語言代碼都不符合您音訊中識別的語言，請從您指定的語言代碼中 Amazon Transcribe 選取最接近的語言相符的語言。然後，它會產生該語言的文字記錄。例如，如果您的媒體使用美式英文 (en-US)，而且您提供 Amazon Transcribe 的語言代碼 zh-CNfr-FRde-DE、和 Amazon Transcribe 很可能會將您的媒體與德文 (de-DE) 相符，並產生德文轉錄。語言代碼和口語語言不符可能會導致文字記錄不正確，因此我們建議您在包含語言代碼時要小心。

## 結合語言識別與其他 Amazon Transcribe 功能

您可以將批次語言識別與任何其他 Amazon Transcribe 功能結合使用。如果將語言識別與其他功能結合使用，則僅限於這些功能支援的語言。例如，如果將語言識別與內容密文搭配使用，您只能使用美式英文 (en-US) 或美國西班牙文 (es-US)，因為這只是可用於密文的語言。請參閱 [支援的語言和特定語言功能](#)，了解詳細資訊。



### Important

如果您在啟用內容密文的情況下使用自動語言識別功能，且您的音訊包含美式英文 (en-US) 或美國西班牙文 (es-US) 以外的語言，您的文字記錄中只會編輯美式英文或美式西班牙文內容。其他語言無法修訂，也且不回出現警告或工作失敗。

## 自訂語言模型，自訂詞彙和自訂詞彙篩選

如果您要在語言識別請求中新增一或多個自訂語言模型、自訂詞彙或自訂詞彙篩選，您必須加入 [LanguageIdSettings](#) 參數。然後，您可以使用對應的自訂語言模型、自訂詞彙和自訂詞彙篩選，指定語言代碼。請注意，多語言識別不支援自訂語言模型。

建議您在使用 [LanguageIdSettings](#) 時加入 `LanguageOptions`，以確保識別正確的語言方言。例如，如果您指定了 en-US 自訂字彙，但 Amazon Transcribe 決定媒體中所使用的語言為 en-AU，則您的自訂字彙不會套用至您的轉錄。如果您加入 `LanguageOptions` 並指定 en-US 為唯一的英文方言，則您的自訂詞彙會套用至您的轉錄。

如需請求中的 [LanguageIdSettings](#) 範例，請參閱 AWS CLI 和 [使用語言識別與批次轉錄](#) 一節中 AWS SDK 下拉式面板中的選項 2。

## 使用語言識別與批次轉錄

您可以使用 AWS Management Console、AWS CLI 或 AWS SDK，在批次轉錄作業中使用自動語言識別；請參閱下列範例：

### AWS Management Console

1. 登入 [AWS Management Console](#)。
2. 在導覽窗格中，選擇轉錄作業，然後選擇建立作業(右上角)。這會開啟指定作業詳細資訊頁面。
3. 在工作設定面板中，找到語言設定區段，然後選擇自動語言識別或自動多語言識別。

如果您知道音訊檔案中有哪些語言，您可以選擇多種語言選項 (從選擇語言下拉式方塊中)。提供語言選項可以提高準確性，但不是必需。

## Specify job details [Info](#)

### Job settings

#### Name

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

#### Language settings

You can transcribe your audio file in a language that you specify or have Amazon Transcribe identify and transcribe it in the predominant language.

- Specific language** [Info](#)  
If you know the language spoken in your source audio, choose this option to get the most accurate results. The options available for additional processing vary between languages.
- Automatic language identification** [Info](#)  
If you don't know the language spoken in your audio files, choose this option. You have access to fewer options for additional processing than if you choose **Specific language**.
- Automatic multiple languages identification** [Info](#)  
If there are multiple languages spoken in your audio files and you're not sure what these languages are, choose this option. This selection provides limited additional processing options compared to **Specific language**.

#### Language options for automatic language identification - *optional*

To improve accuracy, choose at least two languages spoken the most often in your audio library. Amazon Transcribe chooses from one of the languages you've specified to transcribe each audio file. Leave this field empty if you're unsure about which languages to select.

- English, US (en-US)
- English, AU (en-AU)
- English, UK (en-GB)
- Hindi, IN (hi-IN)
- Spanish, US (es-US)

4. 填入您要包含在指定作業詳細資訊頁面上的任何其他欄位，然後選擇下一步。這會引導您前往設定工作 - 選擇性頁面。

## Configure job - *optional* Info

### Audio settings

- Audio identification** Info  
Choose to split multi-channel audio into separate channels for transcription, or identify speakers in the input audio.
- Alternative results** Info  
Enable to view more transcription results

### Content removal

Content removal conceals information in the resulting transcript from your source audio file. Amazon Transcribe changes items in the transcript and does not modify the source audio.

- PII redaction** Info  
Label the type of PII and also mask the content with the PII entity type in the transcription output. For example, (123) 456-7890 will be masked as [PHONE].
- Vocabulary filtering** Info  
Vocabulary filtering can remove, mask or tag specified words in the final transcript.

### Customization

- Custom vocabulary** Info  
A custom vocabulary improves the accuracy of recognizing words and phrases specific to your use case.

Cancel Previous **Create Job**

5. 選擇建立作業以執行轉錄作業。

## AWS CLI

此範例使用指 [start-transcription-job](#) 令和 `IdentifyLanguage` 參數。如需詳細資訊，請參閱 [StartTranscriptionJob](#) 及 [LanguageIdSettings](#)。

選項 1：沒有 `language-id-settings` 參數。如果您未在請求中包含自訂語言模型、自訂詞彙或自訂詞彙篩選，請使用此選項。 `language-options` 為選用操作，但建議您採用。

```
aws transcribe start-transcription-job \
--region us-west-2 \
--transcription-job-name my-first-transcription-job \
--media MediaFileUri=s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac \
--output-bucket-name DOC-EXAMPLE-BUCKET \
--output-key my-output-files/ \
--identify-language \ (or --identify-multiple-languages) \
--language-options "en-US" "hi-IN"
```

選項 2：使用 `language-id-settings` 參數。如果您要在請求中包含自訂語言模型、自訂詞彙或自訂詞彙篩選，請使用此選項。

```
aws transcribe start-transcription-job \
--region us-west-2 \
--transcription-job-name my-first-transcription-job \
--media MediaFileUri=s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac \
--output-bucket-name DOC-EXAMPLE-BUCKET \
--output-key my-output-files/ \
--identify-language \ (or --identify-multiple-languages) \
--language-options "en-US" "hi-IN" \
--language-id-settings en-US=VocabularyName=my-en-US-vocabulary,en-US=VocabularyFilterName=my-en-US-vocabulary-filter,en-US=LanguageModelName=my-en-US-language-model,hi-IN=VocabularyName=my-hi-IN-vocabulary,hi-IN=VocabularyFilterName=my-hi-IN-vocabulary-filter
```

以下是使用 [start-transcription-job](#) 命令的另一個範例，以及識別語言的要求主體。

```
aws transcribe start-transcription-job \
--region us-west-2 \
--cli-input-json file://filepath/my-first-language-id-job.json
```

該文件 `my-first-language-id-job.json` 包含以下請求主體。

選項 1：沒有 `LanguageIdSettings` 參數。如果您未在請求中包含自訂語言模型、自訂詞彙或自訂詞彙篩選，請使用此選項。 `LanguageOptions` 為選用操作，但建議您採用。

```
{
  "TranscriptionJobName": "my-first-transcription-job",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
  },
}
```

```

"OutputBucketName": "DOC-EXAMPLE-BUCKET",
"OutputKey": "my-output-files/",
"IdentifyLanguage": true, (or "IdentifyMultipleLanguages": true),
"LanguageOptions": [
    "en-US", "hi-IN"
]
}

```

選項 2：使用 `LanguageIdSettings` 參數。如果您要在請求中包含自訂語言模型、自訂詞彙或自訂詞彙篩選，請使用此選項。

```

{
  "TranscriptionJobName": "my-first-transcription-job",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
  },
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",
  "OutputKey": "my-output-files/",
  "IdentifyLanguage": true, (or "IdentifyMultipleLanguages": true)
  "LanguageOptions": [
    "en-US", "hi-IN"
  ],
  "LanguageIdSettings": {
    "en-US" : {
      "LanguageModelName": "my-en-US-language-model",
      "VocabularyFilterName": "my-en-US-vocabulary-filter",
      "VocabularyName": "my-en-US-vocabulary"
    },
    "hi-IN": {
      "VocabularyName": "my-hi-IN-vocabulary",
      "VocabularyFilterName": "my-hi-IN-vocabulary-filter"
    }
  }
}

```

## AWS SDK for Python (Boto3)

這個範例會使用 [開始工作](#) 方法的 `IdentifyLanguage` 引數 AWS SDK for Python (Boto3) 來識別檔案的語言。如需詳細資訊，請參閱 [StartTranscriptionJob](#) 及 [LanguageIdSettings](#)。

如需使用 AWS SDK 的其他範例，包括特定功能、案例和跨服務範例，請參閱本章。 [使用 SDK 進行 Amazon Transcribe 的代碼示例 AWS](#)

選項 1：沒有 `LanguageIdSettings` 參數。如果您未在請求中包含自訂語言模型、自訂詞彙或自訂詞彙篩選，請使用此選項。 `LanguageOptions` 為選用操作，但建議您採用。

```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
job_name = "my-first-transcription-job"
job_uri = "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
transcribe.start_transcription_job(
    TranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'DOC-EXAMPLE-BUCKET',
    OutputKey = 'my-output-files/',
    MediaFormat = 'flac',
    IdentifyLanguage = True, (or IdentifyMultipleLanguages = True),
    LanguageOptions = [
        'en-US', 'hi-IN'
    ]
)

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)
```

選項 2：使用 `LanguageIdSettings` 參數。如果您要在請求中包含自訂語言模型、自訂詞彙或自訂詞彙篩選，請使用此選項。

```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe')
job_name = "my-first-transcription-job"
job_uri = "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
transcribe.start_transcription_job(
    TranscriptionJobName = job_name,
    Media = {
```

```
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'DOC-EXAMPLE-BUCKET',
    OutputKey = 'my-output-files/',
    MediaFormat='flac',
    IdentifyLanguage=True, (or IdentifyMultipleLanguages=True)
    LanguageOptions = [
        'en-US', 'hi-IN'
    ],
    LanguageIdSettings={
        'en-US': {
            'VocabularyName': 'my-en-US-vocabulary',
            'VocabularyFilterName': 'my-en-US-vocabulary-filter',
            'LanguageModelName': 'my-en-US-language-model'
        },
        'hi-IN': {
            'VocabularyName': 'my-hi-IN-vocabulary',
            'VocabularyFilterName': 'my-hi-IN-vocabulary-filter'
        }
    }
)

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)
```

## 使用串流轉錄進行語言識別

串流語言識別可以識別媒體串流中使用的優勢語言。Amazon Transcribe 至少需要三秒鐘的語音以識別語言。

如果您的串流僅包含一種語言，您可以啟用單一語言識別功能，識別媒體檔案中所使用的優勢語言，並僅使用此語言建立文字記錄。

如果您的串流包含一種以上的語言，您可以啟用多語言識別功能，識別串流中所使用的所有語言，並使用每種識別的語言建立您的文字記錄。請注意，會產生多語言文字記錄。您可以使用其他服務，例如 Amazon Transcribe，翻譯您的文字記錄。

若要使用串流語言識別，您必須提供至少兩個語言代碼，而且每個串流只能選擇一種語言方言。這表示您無法選擇 en-US 和 en-AU 作為相同轉錄的語言選項。

您也可以選擇從您提供的語言代碼集中，選擇偏好的語言。新增偏好語言可加速語言識別流程，這對於短音訊片段很實用。

### Important

如果您提供的語言代碼都不符合語言或您音訊中識別的語言，Amazon Transcribe 從您指定的語言代碼選擇最接近的語言相符的語言。然後，它會產生該語言的文字記錄。例如，如果您的媒體使用美式英文 (en-US)，而且您提供 Amazon Transcribe 的語言代碼是 zh-CN、fr-FR 和 de-DE，則 Amazon Transcribe 很可能會將您的媒體與德文 (de-DE) 配對，並產生德文轉錄。語言代碼和口語語言不符可能會導致文字記錄不正確，因此我們建議您在包含語言代碼時要小心。

如果您的媒體包含兩個聲道，則 Amazon Transcribe 可以識別每個聲道所使用的優勢語言。在這種情況下，請將 [ChannelIdentification](#) 參數設定為 true，並分別轉錄每個聲道。請注意，此參數的預設值為 false。如果不變更，則只會轉錄第一個聲道，而且只會識別一種語言。

串流語言識別無法與自訂語言模型或修訂結合使用。如果將語言識別與其他功能結合使用，則僅限於這些功能和串流轉錄中支援的語言。請參閱[支援的語言](#)。

### Note

PCM 和 FLAC 是唯一支援串流語言識別的音訊格式。

## 識別多語言音訊中的語言

多語言識別功能適用於多語言串流，並提供可反映串流中所有支援語言的文字記錄。這代表如果發言者在對話中改變語言，或每個參與者說的是不同的語言，則您的轉錄輸出會正確偵測並轉錄每種語言。

例如，如果您的串流包含在美國英文 (en-US) 和印地文 (hi-IN) 之間交替使用的雙語使用者，則多語言識別可以識別並將美國英文轉錄為 en-US，並將口語印地文轉錄為 hi-IN。這與單一語言識別不同，只使用一種優勢語言以建立轉錄。在這種情況下，任何不是優勢語言的口語語言都會被錯誤轉錄。



**Note**

多語言識別目前不支援修訂和自訂語言模型。

## 透過串流媒體使用語言識別

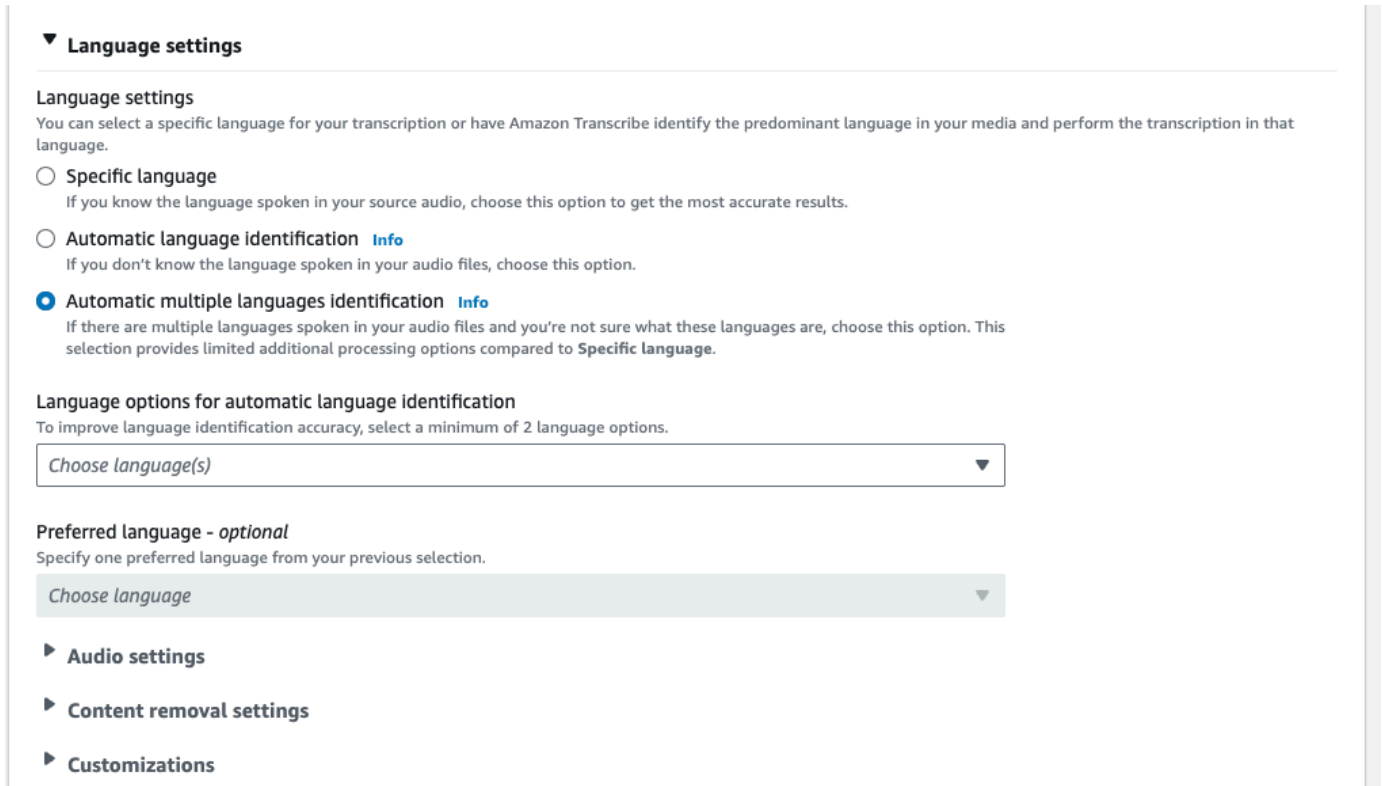
您可以使用 AWS Management Console、HTTP/2 或 WebSockets，在串流轉錄中使用自動語言識別；請參閱下列，取得範例：

### AWS Management Console

1. 登入 [AWS Management Console](#)。
2. 在導覽窗格中，選擇即時轉錄。向下捲動至語言設定，如果此欄位已最小化，請展開此欄位。

The screenshot shows the 'Real-time transcription' interface in the AWS Management Console. At the top, it says 'Real-time transcription' with an 'Info' link. Below that, a subtitle reads: 'See how Amazon Transcribe creates a text copy of speech in real time. Choose **Start streaming** and talk.' The main area is titled 'Transcription' and contains two buttons: 'Download full transcript' and 'Start streaming'. Below the buttons, there is a section for 'Transcription output' with the text 'Choose Start streaming to begin a real-time transcription of what you speak into your microphone'. To the right of this section, it says 'Current language: Automatic, Confidence: N/A'. At the bottom of the main area, there is a progress indicator '00:00 of 15:00 min audio stream'. Below the main area, there are four expandable menu items: 'Language settings', 'Audio settings', 'Content removal settings', and 'Customizations'.

### 3. 選取自動語言識別或自動多語言識別。



The screenshot displays the 'Language settings' section of the Amazon Transcribe console. It includes three radio button options for language identification: 'Specific language', 'Automatic language identification', and 'Automatic multiple languages identification'. The 'Automatic multiple languages identification' option is selected. Below these options is a dropdown menu for 'Language options for automatic language identification' with the text 'Choose language(s)'. Further down is another dropdown menu for 'Preferred language - optional' with the text 'Choose language'. At the bottom, there are three expandable sections: 'Audio settings', 'Content removal settings', and 'Customizations'.

▼ **Language settings**

**Language settings**  
You can select a specific language for your transcription or have Amazon Transcribe identify the predominant language in your media and perform the transcription in that language.

**Specific language**  
If you know the language spoken in your source audio, choose this option to get the most accurate results.

**Automatic language identification** [Info](#)  
If you don't know the language spoken in your audio files, choose this option.

**Automatic multiple languages identification** [Info](#)  
If there are multiple languages spoken in your audio files and you're not sure what these languages are, choose this option. This selection provides limited additional processing options compared to **Specific language**.

**Language options for automatic language identification**  
To improve language identification accuracy, select a minimum of 2 language options.

Choose language(s) ▼

**Preferred language - optional**  
Specify one preferred language from your previous selection.

Choose language ▼

▶ **Audio settings**

▶ **Content removal settings**

▶ **Customizations**

4. 為您的轉錄提供至少兩個語言代碼。請注意，每種語言只能提供一種方言。例如，您無法同時選擇 en-US 和 fr-CA 作為相同轉錄的語言選項。

▼ **Language settings**

**Language settings**  
You can select a specific language for your transcription or have Amazon Transcribe identify the predominant language in your media and perform the transcription in that language.

**Specific language**  
If you know the language spoken in your source audio, choose this option to get the most accurate results.

**Automatic language identification** [Info](#)  
If you don't know the language spoken in your audio files, choose this option.

**Automatic multiple languages identification** [Info](#)  
If there are multiple languages spoken in your audio files and you're not sure what these languages are, choose this option. This selection provides limited additional processing options compared to **Specific language**.

**Language options for automatic language identification**  
To improve language identification accuracy, select a minimum of 2 language options.

Choose language(s) ▼

English, US (en-US) × French, CA (fr-CA) ×

**Preferred language - optional**  
Specify one preferred language from your previous selection.

Choose language ▲

Q

None

English, US (en-US)

French, CA (fr-CA)

5. (選擇性) 從您在上一步中選擇的語言子集中，您可以為文字記錄選擇偏好的語言。

▼ **Language settings**

**Language settings**  
You can select a specific language for your transcription or have Amazon Transcribe identify the predominant language in your media and perform the transcription in that language.

**Specific language**  
If you know the language spoken in your source audio, choose this option to get the most accurate results.

**Automatic language identification** [Info](#)  
If you don't know the language spoken in your audio files, choose this option.

**Language options for automatic language identification**  
To improve language identification accuracy, select a minimum of 2 language options.

Choose language(s) ▼

English, US (en-US) × French, CA (fr-CA) ×

**Preferred language - optional**  
Specify one preferred language from your previous selection.

Choose language ▲

None

English, US (en-US)

French, CA (fr-CA)

► **Customizations**

6. 您現在已準備好轉錄串流。選擇開始串流並開始說話。若要結束聽寫，選擇停止串流。

## HTTP/2 串流

此範例會建立啟用語言識別的 HTTP/2 請求。如需有關搭配 Amazon Transcribe 使用 HTTP/2 串流的詳細資訊，請參閱 [設定 HTTP/2 串流](#)。如需 Amazon Transcribe 特定參數和標頭的詳細資訊，請參閱 [StartStreamTranscription](#)。

```
POST /stream-transcription HTTP/2
host: transcribestreaming.us-west-2.amazonaws.com
X-Amz-Target: com.amazonaws.transcribe.Transcribe.StartStreamTranscription
Content-Type: application/vnd.amazon.eventstream
X-Amz-Content-Sha256: string
X-Amz-Date: 20220208T235959Z
Authorization: AWS4-HMAC-SHA256 Credential=access-key/20220208/us-west-2/transcribe/
aws4_request, SignedHeaders=content-type;host;x-amz-content-sha256;x-amz-date;x-amz-
target;x-amz-security-token, Signature=string
x-amzn-transcribe-media-encoding: flac
x-amzn-transcribe-sample-rate: 16000
x-amzn-transcribe-identify-language: true
x-amzn-transcribe-language-options: en-US,de-DE
x-amzn-transcribe-preferred-language: en-US
transfer-encoding: chunked
```

此範例會建立啟用多語言識別的 HTTP/2 請求。如需有關搭配 Amazon Transcribe 使用 HTTP/2 串流的詳細資訊，請參閱 [設定 HTTP/2 串流](#)。如需 Amazon Transcribe 特定參數和標頭的詳細資訊，請參閱 [StartStreamTranscription](#)。

```
POST /stream-transcription HTTP/2
host: transcribestreaming.us-west-2.amazonaws.com
X-Amz-Target: com.amazonaws.transcribe.Transcribe.StartStreamTranscription
Content-Type: application/vnd.amazon.eventstream
X-Amz-Content-Sha256: string
X-Amz-Date: 20220208T235959Z
Authorization: AWS4-HMAC-SHA256 Credential=access-key/20220208/us-west-2/transcribe/
aws4_request, SignedHeaders=content-type;host;x-amz-content-sha256;x-amz-date;x-amz-
target;x-amz-security-token, Signature=string
x-amzn-transcribe-media-encoding: flac
x-amzn-transcribe-sample-rate: 16000
x-amzn-transcribe-identify-multiple-languages: true
x-amzn-transcribe-language-options: en-US,de-DE
x-amzn-transcribe-preferred-language: en-US
```

```
transfer-encoding: chunked
```

如果您的請求中使用 `identify-language` 或 `identify-multiple-languages`，則也必須納入 `language-options`。您不能在相同的請求中同時使用 `language-code` 和 `identify-language`。

您可以在 [《API 參考資料》](#) 中找到參數定義；所有 AWS API 操作的通用參數都會於 [通用參數](#) 一節列出。

## WebSocket 串流

此範例會建立預先簽署 URL，並在 WebSocket 串流中使用語言識別。已加入分行符號以提高可讀性。如需有關搭配 Amazon Transcribe 使用 WebSocket 串流的詳細資訊，請參閱 [設定串 WebSocket 流](#)。如需參數詳細資訊，請參閱 [StartStreamTranscription](#)。

```
GET wss://transcribestreaming.us-west-2.amazonaws.com:8443/stream-transcription-  
websocket?  
&X-Amz-Algorithm=AWS4-HMAC-SHA256  
&X-Amz-Credential=AKIAIOSFODNN7EXAMPLE%2F20220208%2Fus-  
west-2%2Ftranscribe%2Faws4_request  
&X-Amz-Date=20220208T235959Z  
&X-Amz-Expires=300  
&X-Amz-Security-Token=security-token  
&X-Amz-Signature=string  
&X-Amz-SignedHeaders=content-type%3Bhost%3Bx-amz-date  
&media-encoding=flac  
&sample-rate=16000  
&identify-language=true  
&language-options=en-US,de-DE  
&preferred-language=en-US
```

此範例會建立預先簽署 URL，並在 WebSocket 串流中使用多語言識別。已加入分行符號以提高可讀性。如需有關搭配 Amazon Transcribe 使用 WebSocket 串流的詳細資訊，請參閱 [設定串 WebSocket 流](#)。如需參數詳細資訊，請參閱 [StartStreamTranscription](#)。

```
GET wss://transcribestreaming.us-west-2.amazonaws.com:8443/stream-transcription-  
websocket?  
&X-Amz-Algorithm=AWS4-HMAC-SHA256  
&X-Amz-Credential=AKIAIOSFODNN7EXAMPLE%2F20220208%2Fus-  
west-2%2Ftranscribe%2Faws4_request  
&X-Amz-Date=20220208T235959Z
```

```
&X-Amz-Expires=300
&X-Amz-Security-Token=security-token
&X-Amz-Signature=string
&X-Amz-SignedHeaders=content-type%3Bhost%3Bx-amz-date
&media-encoding=flac
&sample-rate=16000
&identify-multiple-languages=true
&language-options=en-US,de-DE
&preferred-language=en-US
```

如果您的請求中使用 `identify-language` 或 `identify-multiple-languages`，則也必須納入 `language-options`。您不能在相同的請求中同時使用 `language-code` 和 `identify-language`。

您可以在 [《API 參考資料》](#) 中找到參數定義；所有 AWS API 操作的通用參數都會於 [通用參數](#) 一節列出。

## 替代轉錄

Amazon Transcribe 轉錄音訊時，會建立相同文字稿的不同版本，並為每個版本指派置信度分數。在典型的轉錄中，您只能獲得具有最高可信度分數的版本。

如果您開啟替代轉錄，則會 Amazon Transcribe 傳回其他可信度較低的成績單版本。您最多可以選擇返回 10 個替代轉錄。如果您指定的替代數量多於 Amazon Transcribe 標識的數量，則只會返回實際的替代數量。

所有替代方案都位於相同的轉錄輸出檔案中，並在區段層級顯示。區段是語音的自然暫停，例如變更喇叭或音訊暫停。

替代轉錄僅適用於批次轉錄。

您的轉錄輸出的結構如下所示。省略號 (...) 在程式碼範例中指出為簡潔起見而移除內容的位置。

### 1. 給定段的完整最終轉錄。

```
"results": {
  "language_code": "en-US",
  "transcripts": [
    {
      "transcript": "The amazon is the largest rainforest on the planet."
    }
  ],
  ...
}
```

### 2. 前一-transcript 節中每個單字的置信度分數。

```
"items": [
  {
    "start_time": "1.15",
    "end_time": "1.35",
    "alternatives": [
      {
        "confidence": "1.0",
        "content": "The"
      }
    ],
    "type": "pronunciation"
  },
  {
    ...
  }
]
```

```

    "start_time": "1.35",
    "end_time": "2.05",
    "alternatives": [
      {
        "confidence": "1.0",
        "content": "amazon"
      }
    ],
    "type": "pronunciation"
  },

```

3. 您的替代轉錄位於轉錄輸出的segments部分中。每個區段的替代方案是依信賴度分數遞減排序。

```

"segments": [
  {
    "start_time": "1.04",
    "end_time": "5.065",
    "alternatives": [
      {
        ...
        "transcript": "The amazon is the largest rain forest on the
planet.",
        "items": [
          {
            "start_time": "1.15",
            "confidence": "1.0",
            "end_time": "1.35",
            "type": "pronunciation",
            "content": "The"
          },
          {
            "start_time": "3.06",
            "confidence": "0.0037",
            "end_time": "3.38",
            "type": "pronunciation",
            "content": "rain"
          },
          {
            "start_time": "3.38",
            "confidence": "0.0037",
            "end_time": "3.96",
            "type": "pronunciation",
            "content": "forest"
          }
        ]
      }
    ]
  }
]

```



```
},
```

#### 4. 轉錄輸出結尾處的狀態。

```
"status": "COMPLETED"  
}
```

## 請求替代轉錄

您可以使用AWS Management Console、或 AWSSDK 要求替代轉錄；如需範例 AWS CLI，請參閱下列內容：

### AWS Management Console

1. 登入 [AWS Management Console](#)。
2. 在功能窗格中，選擇「轉錄工作」，然後選取「建立工作」（右上角）。這會開啟 [指定工作詳細資訊] 頁面。

## Specify job details [Info](#)

### Job settings

**Name**

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

**Model type [Info](#)**  
Choose the type of model to use for the transcription job.

**General model**  
To use a model that is not specialized for a particular use case, choose this option. Configuration options vary between languages.

**Custom language model**  
To use a model that you trained for your specific use case, choose this option. This model has fewer configuration options than the general model.

**Language settings**  
You can transcribe your audio file in a language that you specify or have Amazon Transcribe identify and transcribe it in the predominant language.

**Specific language [Info](#)**  
If you know the language spoken in your source audio, choose this option to get the most accurate results. The options available for additional processing vary between languages.

**Automatic language identification [Info](#)**  
If you don't know the language spoken in your audio files, choose this option. You have access to fewer options for additional processing than if you choose **Specific language**.

**Language**  
Choose the language of the input audio.

► **Additional settings**

3. 填寫您要包含在 [指定工作詳細資訊] 頁面上的任何欄位，然後選取 [下一步]。這會帶您前往 [設定工作-選擇性] 頁面。

選取 [替代結果]，然後指定您想要在成績單中輸入的替代轉錄結果數目上限。

## Configure job - optional Info

### Audio settings

**Audio identification** Info  
Choose to split multi-channel audio into separate channels for transcription, or identify speakers in the input audio.

---

**Alternative results** Info  
Enable to view more transcription results

**Maximum alternatives**  
Provide the number of alternative transcription to provide in the text output.

The maximum number of alternative results is 10.

4. 選取 [建立工作] 以執行轉錄工作。

## AWS CLI

此範例使用指 [start-transcription-job](#) 令和 `ShowAlternatives` 參數。如需詳細資訊，請參閱 [StartTranscriptionJob](#) 及 [ShowAlternatives](#)。

請注意，如果您 `ShowAlternatives=true` 在請求中包含，則還必須包括 `MaxAlternatives`。

```
aws transcribe start-transcription-job \
--region us-west-2 \
--transcription-job-name my-first-transcription-job \
--media MediaFileUri=s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac \
--output-bucket-name DOC-EXAMPLE-BUCKET \
--output-key my-output-files/ \
--language-code en-US \
--settings ShowAlternatives=true,MaxAlternatives=4
```

這是使用 [start-transcription-job](#) 命令的另一個示例，以及包含替代轉錄的請求主體。

```
aws transcribe start-transcription-job \
--region us-west-2 \
```

```
--cli-input-json file://filepath/my-first-alt-transcription-job.json
```

該文件 my-first-alt-transcription-job.json 包含以下請求主體。

```
{
  "TranscriptionJobName": "my-first-transcription-job",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
  },
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",
  "OutputKey": "my-output-files/",
  "LanguageCode": "en-US",
  "Settings": {
    "ShowAlternatives": true,
    "MaxAlternatives": 4
  }
}
```

## AWS SDK for Python (Boto3)

下列範例會使用使用 [start\\_transcription\\_job](#) 方法的 ShowAlternatives 引數 AWS SDK for Python (Boto3) 來要求替代轉錄。如需詳細資訊，請參閱 [StartTranscriptionJob](#) 及 [ShowAlternatives](#)。

如需使用 AWS SDK 的其他範例，包括特定功能、案例和跨服務範例，請參閱本 [使用 SDK 進行 Amazon Transcribe 的代碼示例 AWS](#) 章。

請注意，如果您 'ShowAlternatives': True 在請求中包含，則還必須包括 MaxAlternatives。

```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
job_name = "my-first-transcription-job"
job_uri = "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
transcribe.start_transcription_job(
    TranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'DOC-EXAMPLE-BUCKET',
    OutputKey = 'my-output-files/',
```

```
LanguageCode = 'en-US',
Settings = {
    'ShowAlternatives':True,
    'MaxAlternatives':4
}
)

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)
```

## 通過自定義詞彙和自定義語言模型提高轉錄準確性

如果您的媒體包含特定網域或非標準字詞 (例如品牌名稱、縮寫字、技術文字和行話), Amazon Transcribe 可能無法在轉錄輸出中正確擷取這些字詞。

若要更正轉錄不準確性並針對特定使用案例自訂輸出, 您可以建立 [自訂詞彙](#) 和 [自訂語言模型](#)。

- [自訂詞彙](#) 旨在調整和提高所有情況下特定單詞的識別和格式。這涉及到提供 Amazon Transcribe 單詞和, 可選, 發音和顯示形式。

如果未 Amazon Transcribe 正確轉譯成績單中的特定詞彙, 您可以建立自訂字彙檔案, 告訴 Amazon Transcribe 您希望這些字詞的顯示方式。這種特定於單詞的方法最適合用於糾正品牌名稱和縮略詞之類的術語。

- [自訂語言模型](#) 旨在捕獲與術語相關聯的上下文。這涉及到提供 Amazon Transcribe 大量特定於域的文本數據。

如果無法 Amazon Transcribe 正確呈現技術術語, 或在成績單中使用不正確的同音字詞, 您可以建立自訂語言模型來教導 Amazon Transcribe 您的網域特定語言。例如, 自定義語言模型可以了解何時使用「浮冰」(浮冰) 與「流」(線性流)。

這種情境感知方法最適合轉錄大量網域特定語音。自定義語言模型可以產生顯著的準確性提高單獨自定義詞彙。使用批次轉錄時, 您可以在請求中同時包含自訂語言模型和自訂字彙。

### Tip

若要達到最高的轉錄準確度, 請將自訂字彙與您的自訂語言模型搭配使用。

如需如何使用建立自訂字彙的影片示範 AWS Management Console, 請參閱 [使用自訂字彙](#)。

如需如何建立及使用自訂語言模型的影片示範, 請參閱 [使用自訂語言模型 \(CLM\) 提升轉錄準確度](#)。

### 透過 Machine LAWS earning 部落格深入瞭解

Custom 自訂

- [使用 F1 比賽的現場轉錄 Amazon Transcribe](#)

自訂語言模型：

- [建立自訂語言模型以提 speech-to-text 升效能Amazon Transcribe](#)
- [使用自定義語言模型提高課堂講座的轉錄準確性Amazon Transcribe](#)

## 自訂詞彙

使用自訂詞彙以提高一或多個特定單字的轉錄準確性。這些通常是特定領域的術語，例如品牌名稱和縮寫、適當名詞以及 Amazon Transcribe 無法正確呈現的單字。

自訂詞彙可以與所有支援的語言搭配使用。請注意，只有您語言的[字元集](#)中列出的字元才可在自訂詞彙使用。

### Important

您使用 Amazon Transcribe 時，您必須對自己資料的完整性負責。請勿在自訂詞彙中輸入機密資訊、個人資訊 (PII) 或受保護的健康資訊 (PHI)。

建立自訂詞彙時的注意事項：

- 每個最多可以有 100 個自訂字彙檔 AWS 帳戶
- 每個自訂詞彙的大小限制為 50 Kb。
- 如果使用 API 建立自訂詞彙，您的詞彙檔案必須是文字 (\*.txt) 格式。如果使用 AWS Management Console，您的字彙檔案可以是文字 (\*.txt) 格式或逗號分隔值 (\*.csv) 格式。
- 自訂詞彙的每個項目不能超過 256 個字元
- 若要使用自訂字彙，該詞彙的建立方式必須與您的轉錄 AWS 區域 相同。

### Tip

您可以使用測試您的自訂詞彙 AWS Management Console。一旦您的自定義詞彙準備好使用，請登錄到 AWS Management Console，選擇實時轉錄，滾動到自定義，打開自定義詞彙，然後從下拉列表中選擇您的自定義詞彙。然後選擇開始串流。在麥克風中朗讀自訂詞彙中的一些單字，看看是否正確顯示。

## 自訂詞彙表與清單

### ⚠ Important

清單格式的自訂詞彙已不再使用。如果您要建立新的自訂詞彙，請使用[表格格式](#)。

表格提供更多選項，以及更適當控制自訂詞彙中單字的輸入和輸出。使用表格時，您必須指定多個類別 (Phrase and DisplayAs)，以便微調輸出。

清單沒有其他選項，因此您只能輸入您希望在文字記錄中出現的項目，取代所有空格為連字號。

AWS Management Console AWS CLI、和 AWS SDK 都以相同的方式使用自訂詞彙表；每種方法的使用方式都不同，因此可能需要額外的格式才能在方法之間成功使用。

如需詳細資訊，請參閱 [使用資料表建立自訂詞彙](#) 及 [使用清單建立自訂字彙](#)。

若要了解如何將 Amazon Augmented AI 與自訂詞彙搭配使用的詳細資訊，請參閱[開始與 Amazon Transcribe 建構人工評論](#)

### 📘 自訂詞彙特定的 API 操作

[CreateVocabulary](#), [DeleteVocabulary](#), [GetVocabulary](#), [ListVocabularies](#), [UpdateVocabulary](#)

## 使用資料表建立自訂詞彙

使用表格格式是建立自訂詞彙的首選方式。詞彙表必須由四個欄 (Phrase, SoundsLike, IPA, and DisplayAs) 組成，可依任何順序包含：

片語	SoundsLike	IPA	DisplayAs
必要。表格中的每一列都必須包含此欄中的項目。  請勿在此欄中使用空格。	SoundsLike 不再支援「自訂字彙」。請將欄留空。此欄中的任何值都將被忽略。我們將在 future 移除對此專欄的支援。	IPA不再支援「自訂字彙」。請將欄留空。此欄中的任何值都將被忽略。我們將在 future 移除對此專欄的支援。	選用。此欄中的列可以保留空白。  您可以在此欄中使用空格。



片語	SoundsLike	IPA	DisplayAs
<p>如果您輸入的內容包含多個單字，請以連字號 (-) 分隔每個單字。例如 <b>Andorra-la-Vella</b> 或 <b>Los-Angeles</b>。</p> <p>對於縮寫詞，任何發音的字母都必須使用句點分隔。之後的句號也需要發音。如果您有複數個縮寫，您必須在縮寫和「s」間使用連字號。例如，「CLI」是 <b>C.L.I.</b> (非 <b>C.L.I</b>) 和「ABCs」是 <b>A.B.C.-s</b> (非 <b>A.B.C-s</b>)。</p> <p>如果您的片語同時包含一個單字和一個縮寫，則這兩個元件必須用連字號分隔。例如，「DynamoDB」是 <b>Dynamo-D.B.</b></p> <p>請勿在此欄中包含數字；必須拼出數字。例如，「VX02Q」是 <b>V.X.-zero-two-Q.</b></p>			<p>定義您的轉錄輸出中項目的外觀。例如，Phrase 欄中的 <b>Andorra-la-Vella</b> 是在 DisplayAs 欄中的 <b>Andorra la Vella</b>。</p> <p>如果此欄中的列是空的，Amazon Transcribe 會使用 Phrase 欄的內容來決定輸出。</p> <p>您可以在此欄中包含數字 (0-9)。</p>

建立表格時要注意的事項：

- 您的表必須包含所有四個列標題(Phrase, SoundsLike, IPA, and DisplayAs)。Phrase欄必須包含每一列上的項目。通過IPA和提供發音輸入的功能SoundsLike不再受支持，您可以將該列保留空白。這些欄中的任何值都將被忽略。
- 每一欄必須以 TAB 或逗號 (,) 分隔；這適用於自訂詞彙檔案中的每一列。如果列包含空白欄，您仍必須為每個欄納入分隔符號 (TAB 或逗號)。
- 只有 IPA 和 DisplayAs 欄才允許使用空格。請勿使用空格以分隔欄。
- IPA而SoundsLike且不再支援「自訂字彙」。請將欄留空。這些欄中的任何值都將被忽略。我們將在 future 移除對此專欄的支援。
- DisplayAs 欄支援符號和特殊字元 (例如 C++)。所有其他欄支援您語言的[字元集](#)頁面上列出的字元。
- 如果要在 Phrase 欄中包含數字，則必須拼出數字。DisplayAs 欄僅支援數字 (0-9)。
- 您必須將表格儲存為 LF 格式的純文字 (\*.txt) 檔案。如果您使用任何其他格式，例如 CRLF，將無法處理您的自訂詞彙。
- 您必須將自訂字彙檔案上傳至 Amazon S3 值區並使用處理，[CreateVocabulary](#)然後才能將其納入轉錄請求中。請參閱 [建立自訂詞彙表](#)，了解指示。

#### Note

縮寫或其他字組的字母如須單獨發音，請在單一字母後面輸入句號 (**A.B.C.**)。若要輸入縮寫的複數形式，例如「ABCs」，請以連字號 (**A.B.C.-s**) 分隔縮寫中的「s」。您可以使用大寫或小寫字母以定義縮寫。並非所有語言都支援縮寫；請參閱 [支援的語言和特定語言功能](#)。

以下是樣本自訂詞彙表 (其中 **[TAB]** 代表一個 tab 字元)：

```
Phrase[TAB]SoundsLike[TAB]IPA[TAB]DisplayAs
Los-Angeles[TAB][TAB][TAB]Los Angeles
Eva-Maria[TAB][TAB][TAB]
A.B.C.-s[TAB][TAB][TAB]ABCs
Amazon-dot-com[TAB][TAB][TAB]Amazon.com
C.L.I.[TAB][TAB][TAB]CLI
Andorra-la-Vella[TAB][TAB][TAB]Andorra la Vella
Dynamo-D.B.[TAB][TAB][TAB]DynamoDB
V.X.-zero-two[TAB][TAB][TAB]VX02
V.X.-zero-two-Q.[TAB][TAB][TAB]VX02Q
```

為了釐清，此處是與欄對齊的相同表格。請勿在自訂詞彙表中的欄間加入空格；您的表格看似應該與前面的範例一樣未對齊。

Phrase	<b>[TAB]</b> SoundsLike	<b>[TAB]</b> IPA	<b>[TAB]</b> DisplayAs
Los-Angeles	<b>[TAB]</b>	<b>[TAB]</b>	<b>[TAB]</b> Los Angeles
Eva-Maria	<b>[TAB]</b>	<b>[TAB]</b>	<b>[TAB]</b>
A.B.C.-s	<b>[TAB]</b>	<b>[TAB]</b>	<b>[TAB]</b> ABCs
amazon-dot-com	<b>[TAB]</b>	<b>[TAB]</b>	<b>[TAB]</b> amazon.com
C.L.I.	<b>[TAB]</b>	<b>[TAB]</b>	<b>[TAB]</b> CLI
Andorra-la-Vella	<b>[TAB]</b>	<b>[TAB]</b>	<b>[TAB]</b> Andorra la Vella
Dynamo-D.B.	<b>[TAB]</b>	<b>[TAB]</b>	<b>[TAB]</b> DynamoDB
V.X.-zero-two	<b>[TAB]</b>	<b>[TAB]</b>	<b>[TAB]</b> VX02
V.X.-zero-two-Q.	<b>[TAB]</b>	<b>[TAB]</b>	<b>[TAB]</b> VX02Q

## 建立自訂詞彙表

若要處理自訂字彙表以搭配使用 Amazon Transcribe，請參閱下列範例：

### AWS Management Console

1. 登入 [AWS Management Console](#)。
2. 在導覽窗格中，選擇自訂詞彙。這會開啟自訂詞彙頁面，您可以在其中檢視現有的詞彙或建立新詞彙。
3. 選擇建立詞彙。

The screenshot shows the 'Custom vocabulary' page in the Amazon Transcribe console. At the top, there's a breadcrumb 'Amazon Transcribe > Custom vocabulary' and a title 'Custom vocabulary' with an 'Info' link. Below the title is a sub-header 'Use custom vocabularies to improve transcription accuracy. Learn more' with a link. The main content is divided into two columns under an 'Overview' section. The left column is titled '1. Create custom vocabulary' and describes creating a vocabulary by uploading a file or adding phrases, mentioning templates (.csv, .txt). The right column is titled '2. Apply to Real-time or batch transcription' and explains that the created vocabulary can be applied to real-time or batch transcription jobs. Below this is a 'Manage vocabularies' section with a search bar, a filter dropdown set to 'All', and a table with columns for Name, Language, Last modified, and Status. The table is currently empty, showing 'Empty resources' and 'No resources to display', with a 'Create vocabulary' button at the bottom.

這會引導您前往建立詞彙頁面。輸入新自訂詞彙的名稱。

此處您有三種選擇：

- a. 從您的電腦上傳 txt 或 csv 檔案。

您可以從頭開始建立自訂詞彙，也可以下載範本以幫助您開始使用。然後，您的詞彙會自動填入檢視和編輯詞彙窗格中。

## Create vocabulary [Info](#)

### Vocabulary settings

**Name**

Vocabulary names can be up to 200 characters in length. Allowed characters: a-z, A-Z, 0-9, periods (.), dashes (-), and underscores (\_).

**Language**

- b. 從一個 Amazon S3 位置導入一個 txt 或 csv 文件。

您可以從頭開始建立自訂詞彙，也可以下載範本以幫助您開始使用。將完成的詞彙檔案上傳至 Amazon S3 儲存貯體，並在請求中指定其 URI。然後，您的詞彙會自動填入檢視和編輯詞彙窗格中。

### Create and import vocabulary [Info](#)

**Vocabulary input source**

File upload  
Upload a vocabulary table from your computer.

S3 location  
Import a vocabulary table from an S3 location.

Create vocabulary on console  
Manually create a vocabulary table on the console.

**Download vocabulary template – Optional**  
Download and complete a custom vocabulary template in your preferred format.

[Download template](#)

**Import from S3**  
Provide a path to the S3 location where your vocabulary file is stored. To find a path, go to [Amazon S3](#).

Resource URI

[View](#) [Browse S3](#)

- c. 在主控台中手動建立詞彙。

捲動至檢視和編輯詞彙窗格，然後選擇新增 10 列。您現在可以手動輸入術語。

### Create and import vocabulary [Info](#)

**Vocabulary input source**

File upload  
Upload a vocabulary table from your computer.

S3 location  
Import a vocabulary table from an S3 location.

Create vocabulary on console  
Manually create a vocabulary table on the console.

---

**View and edit vocabulary (0)** [Reset vocabulary](#) [Delete](#) [Download latest vocabulary](#)

[Show all](#) [< 1 >](#)

Phrase <a href="#">↗</a>	SoundsLike (optional) <a href="#">↗</a>	IPA (optional) <a href="#">↗</a>	DisplayAs (optional) <a href="#">↗</a>
No rows added yet			
<a href="#">Add 10 rows</a>			

4. 您可以在檢視和編輯詞彙窗格中編輯詞彙。按一下您想要修改的項目，進行變更。

**View and edit vocabulary - new (10)** [Info](#) Reset vocabulary Delete Download latest ▼

Q Filter Phrase, SoundsLike, IPA or DisplayAs Show all ▼ < 1 >

<input type="checkbox"/>	Phrase <a href="#">✎</a> ▼	SoundsLike - optional <a href="#">✎</a> ▼	IPA - optional <a href="#">✎</a> ▼	DisplayAs - optional <a href="#">✎</a> ▼
<input type="checkbox"/>	Amazon-E.-C.-two	-	-	Amazon EC2
<input type="checkbox"/>	Amazon-S.-three	-	-	Amazon S3
<input type="checkbox"/>	Amazon-elasticashe	-	-	Amazon ElastiCache
<input type="checkbox"/>	Amazon-sagemaker	-	-	Amazon SageMaker
<input type="checkbox"/>	A.-W.-S.-iam	-	-	AWS IAM
<input type="checkbox"/>	A.-W.-S.-I.-o.-T.	-	-	AWS IoT
<input type="checkbox"/>	A.-W.-S.-W.-A.-F.	-	-	AWS WAF
<input type="checkbox"/>	c.-plus-plus	-	-	C++
<input type="checkbox"/>	nice-d.-c.-v.	-	-	NICE DCV
<input type="checkbox"/>	w.-w.-w.-dot-amazon-dot-com	-	-	www.amazon.com

Add row

如果您執行時發生錯誤，您會收到詳細的錯誤訊息，以在處理詞彙前修正任何問題。請注意，如果您選擇建立詞彙前未更正所有錯誤，詞彙請求就會失敗。

**View and edit vocabulary - new (4)** [Info](#) Reset vocabulary Delete Download latest ▼

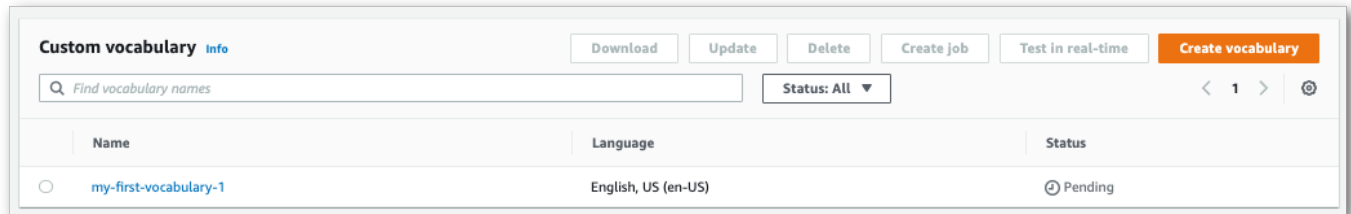
Q Filter Phrase, SoundsLike, IPA or DisplayAs Show all ▼ < 1 >

<input type="checkbox"/>	Phrase <a href="#">✎</a> ▼	SoundsLike - optional <a href="#">✎</a> ▼	IPA - optional <a href="#">✎</a> ▼	DisplayAs - optional <a href="#">✎</a> ▼
<input type="checkbox"/>	<input type="text" value="Amazon-E.-C. two"/> <span>✕</span> <span>✓</span> ⚠ Phrase contains unsupported characters (" "). Phrase contains a formatting error.	-	-	Amazon EC2
<input type="checkbox"/>	Amazon-S.-three	-	-	Amazon S3
<input type="checkbox"/>	c.-plus-plus	-	-	C++
<input type="checkbox"/>	w.-w.-w.-dot-amazon-dot-com	-	-	www.amazon.com

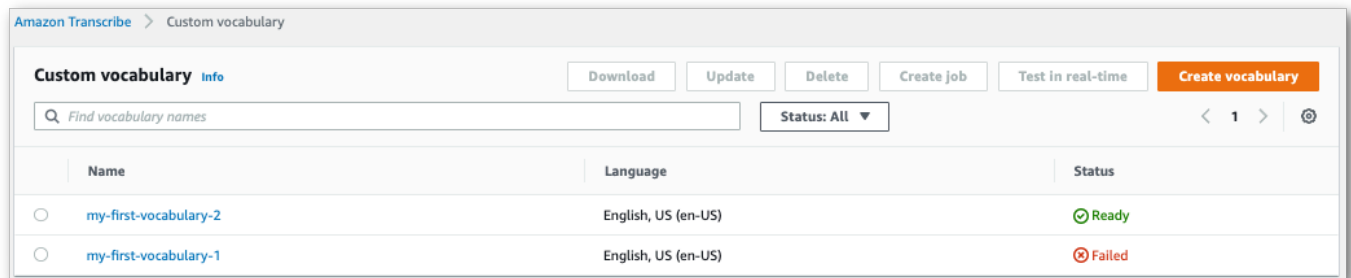
Add row

選擇核取記號 (✓) 以儲存變更，或選擇「X」以捨棄變更。

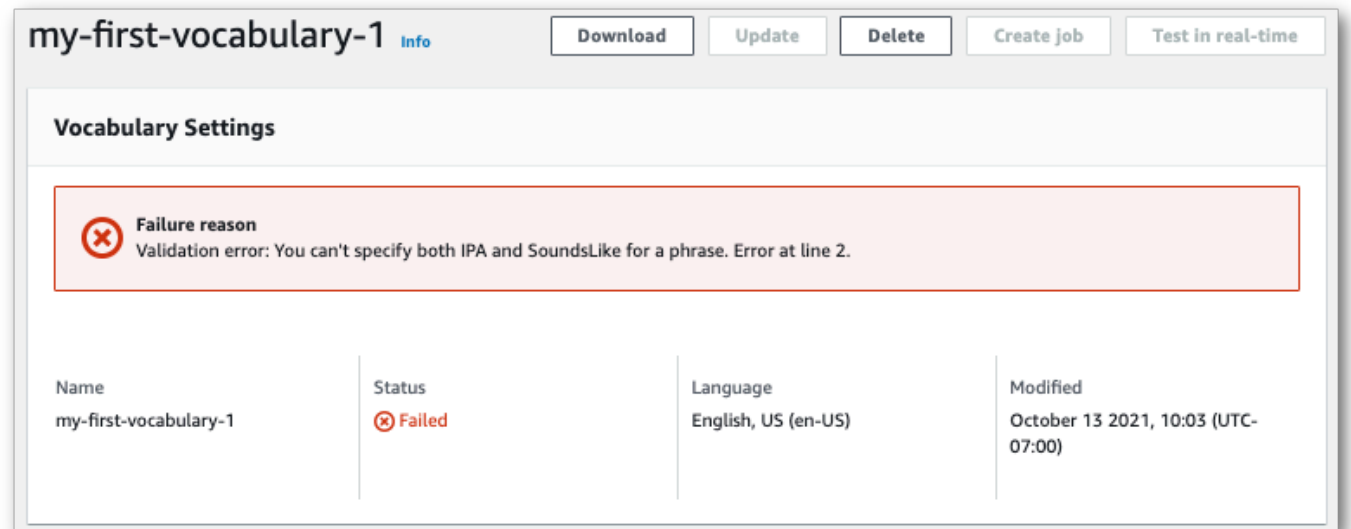
5. 或者，將標籤新增至您的自訂詞彙。完成所有欄位並滿意您的詞彙後，選擇頁面底部的建立詞彙。這會引導您返回自訂詞彙頁面，您可以在此檢視自訂詞彙的狀態。狀態從「待處理」變更為「就緒」時，您的自訂詞彙可以與轉錄搭配使用。



6. 如果狀態變更為「失敗」，選擇自訂詞彙的名稱以前往其資訊頁面。



此頁面頂端有失敗原因橫幅，提供自訂詞彙失敗原因的相關資訊。請更正文字檔案中的錯誤，然後再試一次。



## AWS CLI

此範例使用 [create-vocabulary](#) 指令，搭配表格式的詞彙檔案。如需詳細資訊，請參閱 [CreateVocabulary](#)。



若要在轉錄工作中使用現有的自訂字彙，請VocabularyName在呼叫[StartTranscriptionJob](#)作業時在[Settings](#)欄位中設定，或從下 AWS Management Console拉式清單中選擇自訂字彙。

```
aws transcribe create-vocabulary \  
--vocabulary-name my-first-vocabulary \  
--vocabulary-file-uri s3://DOC-EXAMPLE-BUCKET/my-vocabularies/my-vocabulary-file.txt \  
--language-code en-US
```

這是使用 [create-vocabulary](#) 指令的另一個範例，以及建立自訂詞彙的請求內文。

```
aws transcribe create-vocabulary \  
--cli-input-json file://filepath/my-first-vocab-table.json
```

該文件 *my-first-vocab-table.json* 包含以下請求主體。

```
{  
  "VocabularyName": "my-first-vocabulary",  
  "VocabularyFileUri": "s3://DOC-EXAMPLE-BUCKET/my-vocabularies/my-vocabulary-  
table.txt",  
  "LanguageCode": "en-US"  
}
```

VocabularyState 從 PENDING 變更為READY 後，您的自訂詞彙即可與轉錄搭配使用。執行下列指令可檢視自訂詞彙的目前狀態：

```
aws transcribe get-vocabulary \  
--vocabulary-name my-first-vocabulary
```

## AWS SDK for Python (Boto3)

此範例使用，AWS SDK for Python (Boto3) 使用 [create\\_](#) 詞彙方法從表格建立自訂字彙。如需詳細資訊，請參閱 [CreateVocabulary](#)。

若要在轉錄工作中使用現有的自訂字彙，請VocabularyName在呼叫[StartTranscriptionJob](#)作業時在[Settings](#)欄位中設定，或從下 AWS Management Console拉式清單中選擇自訂字彙。

如需使用 AWS SDK 的其他範例，包括特定功能、案例和跨服務範例，請參閱本章。[使用 SDK 進行 Amazon Transcribe 的代碼示例 AWS](#)

```
from __future__ import print_function
```

```
import time
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
vocab_name = "my-first-vocabulary"
response = transcribe.create_vocabulary(
    LanguageCode = 'en-US',
    VocabularyName = vocab_name,
    VocabularyFileUri = 's3://DOC-EXAMPLE-BUCKET/my-vocabularies/my-vocabulary-
table.txt'
)

while True:
    status = transcribe.get_vocabulary(VocabularyName = vocab_name)
    if status['VocabularyState'] in ['READY', 'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
print(status)
```

### Note

如果您為自訂字彙檔案建立新 Amazon S3 值區，請確定提出 [CreateVocabulary](#) 請求的 IAM 角色具有存取此值區的權限。如果角色沒有正確的授權，您的請求將失敗。您可以選擇性地在請求中加入 `DataAccessRoleArn` 參數來指定 IAM 角色。如需中 IAM 角色和原則的詳細資訊 Amazon Transcribe，請參閱 [Amazon Transcribe 身分型政策範例](#)。

## 使用清單建立自訂字彙

### Important

[清單格式的自訂字彙已被取代](#)，因此，如果您要建立新的自訂字彙，我們強烈建議您使用 [表格格式](#)。

您可以使用 AWS Management Console、AWS CLI 或 SDK 從清單建立自訂字彙。AWS

- **AWS Management Console**：您必須創建並上傳包含自定義詞彙的文本文件。您可以使用以行分隔或逗號分隔的項目。請注意，您的清單必須以 LF 格式儲存為文字 (\*.txt) 檔案。如果您使用任何其他格式，例如 CRLF，您的自訂字彙將不被接受 Amazon Transcribe。

- AWS CLI和 AWSSDK：您必須使用旗標在 API 呼叫中以逗號分隔的條目包含自訂詞彙。[Phrases](#)

如果一個項目包含多個單字，您必須以連字符連接每個單字。例如，您將「洛杉磯」包括為**Los-Angeles**和「安道爾城」作為**Andorra-la-Vella**

以下是兩種有效清單格式的範例。如需有關特定[建立自訂詞彙表的自訂方法](#)的範例，請參閱。

- 以逗號分隔的項目：

```
Los-Angeles,CLI,Eva-Maria,ABCs,Andorra-la-Vella
```

- 以行分隔的項目：

```
Los-Angeles
CLI
Eva-Maria
ABCs
Andorra-la-Vella
```

### Important

您只能使用您的語言支援的字元。如需詳細資訊，請參閱您語言的[字元集](#)。

此作業不支援自訂字彙清單 [CreateMedicalVocabulary](#)。如果要 [使用資料表建立自訂詞彙](#) 建立自訂醫療字彙，您必須使用表格格式；如需說明，請參閱。

## 建立自訂詞彙表的自訂

若要處理自訂字彙清單以搭配使用 Amazon Transcribe，請參閱下列範例：

### AWS CLI

此範例使用「[建立詞彙](#)」指令搭配清單格式的自訂字彙檔案。如需詳細資訊，請參閱 [CreateVocabulary](#)。

```
aws transcribe create-vocabulary \  
--vocabulary-name my-first-vocabulary \  
--language-code en-US \  

```

```
--phrases {CLI,Eva-Maria,ABCs}
```

這是另一個使用「[創建詞彙](#)」命令的示例，以及創建自定義詞彙的請求主體。

```
aws transcribe create-vocabulary \  
--cli-input-json file://filepath/my-first-vocab-list.json
```

該文件 `my-first-vocab-list.json` 包含以下請求主體。

```
{  
  "VocabularyName": "my-first-vocabulary",  
  "LanguageCode": "en-US",  
  "Phrases": [  
    "CLI", "Eva-Maria", "ABCs"  
  ]  
}
```

從 `VocabularyState` 更改 `PENDING` 為後 `READY`，您的自定義詞彙即可與轉錄一起使用。使用命令可檢視自訂字彙的目前狀態：

```
aws transcribe get-vocabulary \  
--vocabulary-name my-first-vocabulary
```

## AWS SDK for Python (Boto3)

此範例使用，AWS SDK for Python (Boto3) 使用 [create\\_word](#) 方法從清單建立自訂字彙。如需詳細資訊，請參閱 [CreateVocabulary](#)。

如需使用 AWS SDK 的其他範例，包括特定功能、案例和跨服務範例，請參閱本章。 [使用 SDK 進行 Amazon Transcribe 的代碼示例 AWS](#)

```
from __future__ import print_function  
import time  
import boto3  
transcribe = boto3.client('transcribe', 'us-west-2')  
vocab_name = "my-first-vocabulary"  
response = transcribe.create_vocabulary(  
    LanguageCode = 'en-US',  
    VocabularyName = vocab_name,  
    Phrases = [  

```

```
        'CLI', 'Eva-Maria', 'ABCs'  
    ]  
)  
  
while True:  
    status = transcribe.get_vocabulary(VocabularyName = vocab_name)  
    if status['VocabularyState'] in ['READY', 'FAILED']:  
        break  
    print("Not ready yet...")  
    time.sleep(5)  
print(status)
```

### Note

如果您為自訂字彙檔案建立新Amazon S3值區，請確定提出[CreateVocabulary](#)請求的IAM角色具有存取此值區的權限。如果角色沒有正確的權限，您的請求將失敗。您可以選擇性地在請求中加入DataAccessRoleArn參數來指定IAM角色。如需中IAM角色和原則的詳細資訊 Amazon Transcribe，請參閱[Amazon Transcribe 身分型政策範例](#)。

## 使用自訂字彙

建立自訂字彙後，您可以將其包含在轉錄請求中；請參閱以下章節以取得範例。

您要包含在請求中的自訂字彙的語言，必須與您為媒體指定的語言代碼相符。如果語言不相符，您的自訂字彙就不會套用至您的轉錄，也沒有警告或錯誤。

### 在批次轉錄中使用自訂詞彙

若要將自訂字彙與批次轉錄搭配使用，請參閱下列範例：

#### AWS Management Console

1. 登入 [AWS Management Console](#)。
2. 在功能窗格中，選擇「轉錄工作」，然後選取「建立工作」（右上角）。這會開啟 [指定工作詳細資訊] 頁面。

## Specify job details [Info](#)

### Job settings

**Name**

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

**Model type** [Info](#)

Choose the type of model to use for the transcription job.

**General model**

To use a model that is not specialized for a particular use case, choose this option. Configuration options vary between languages.

**Custom language model**

To use a model that you trained for your specific use case, choose this option. This model has fewer configuration options than the general model.

**Language settings**

You can transcribe your audio file in a language that you specify or have Amazon Transcribe identify and transcribe it in the predominant language.

**Specific language** [Info](#)

If you know the language spoken in your source audio, choose this option to get the most accurate results. The options available for additional processing vary between languages.

**Automatic language identification** [Info](#)

If you don't know the language spoken in your audio files, choose this option. You have access to fewer options for additional processing than if you choose **Specific language**.

**Language**

Choose the language of the input audio.

命名您的工作並指定輸入媒體。選擇性地包含任何其他欄位，然後選擇下一步。

3. 在 [設定工作] 頁面底部的 [自訂] 面板中，開啟 [自訂字彙]。

## Configure job - optional [Info](#)

### Audio settings

**Audio identification** [Info](#)  
Choose to split multi-channel audio into separate channels for transcription, or identify speakers in the input audio.

---

**Alternative results** [Info](#)  
Enable to view more transcription results

---

### Content removal

Content removal conceals information in the resulting transcript from your source audio file. Amazon Transcribe changes items in the transcript and does not modify the source audio.

**PII redaction** [Info](#)  
Label the type of PII and also mask the content with the PII entity type in the transcription output. For example, (123) 456-7890 will be masked as [PHONE].

---

**Vocabulary filtering** [Info](#)  
Vocabulary filtering can remove, mask or tag specified words in the final transcript.

---

### Customization

**Custom vocabulary** [Info](#)  
A custom vocabulary improves the accuracy of recognizing words and phrases specific to your use case.

**Vocabulary selection**  
The vocabularies shown here are based on your language settings. You can choose up to one vocabulary per language. You can also [create a new vocabulary](#). [↗](#)

Choose a vocabulary ▼

Cancel Previous Create job

4. 從下拉菜單中選擇您的自定義詞彙。

選取 [建立工作] 以執行轉錄工作。

## AWS CLI

此範例使用[start-transcription-job](#)指令和Settings參數搭配VocabularyName子參數。如需詳細資訊，請參閱 [StartTranscriptionJob](#) 及 [Settings](#)。

```
aws transcribe start-transcription-job \  
--region us-west-2 \  
--transcription-job-name my-first-transcription-job \  
--media MediaFileUri=s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac \  
--output-bucket-name DOC-EXAMPLE-BUCKET \  
--output-key my-output-files/ \  
--language-code en-US \  
--settings VocabularyName=my-first-vocabulary
```

以下是使用[start-transcription-job](#)命令的另一個範例，以及包含該工作的自訂詞彙的要求主體。

```
aws transcribe start-transcription-job \  
--region us-west-2 \  
--cli-input-json file://my-first-vocabulary-job.json
```

該文件 *my-first-vocabulary-job.json* 包含以下請求主體。

```
{  
  "TranscriptionJobName": "my-first-transcription-job",  
  "Media": {  
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"  
  },  
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",  
  "OutputKey": "my-output-files/",  
  "LanguageCode": "en-US",  
  "Settings": {  
    "VocabularyName": "my-first-vocabulary"  
  }  
}
```

## AWS SDK for Python (Boto3)

這個範例會使用Settings引數AWS SDK for Python (Boto3)來包含自訂字彙，並使用[啟動工作方法的引數](#)。如需詳細資訊，請參閱 [StartTranscriptionJob](#) 及 [Settings](#)。

如需使用AWS SDK 的其他範例，包括特定功能、案例和跨服務範例，請參閱本[使用 SDK 進行 Amazon Transcribe 的代碼示例](#) [AWS](#)章。



```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
job_name = "my-first-transcription-job"
job_uri = "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
transcribe.start_transcription_job(
    TranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'DOC-EXAMPLE-BUCKET',
    OutputKey = 'my-output-files/',
    LanguageCode = 'en-US',
    Settings = {
        'VocabularyName': 'my-first-vocabulary'
    }
)

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)
```

## 在串流轉錄中使用自訂詞彙

若要將自訂字彙與串流轉錄搭配使用，請參閱下列範例：

### AWS Management Console

1. 登入[AWS Management Console](#)。
2. 在瀏覽窗格中，選擇 Real-time transcription (即時轉錄)。向下捲動至「自訂」，如果此欄位已最小化，請展開此欄位。

## Real-time transcription [Info](#)

See how Amazon Transcribe creates a text copy of speech in real time. Choose **Start streaming** and talk.

### Transcription

[Download full transcript](#) [Start streaming](#)

Transcription output Current language: English, US

Choose **Start streaming** to begin a real-time transcription of what you speak into your microphone

00:00 of 15:00 min audio stream

- ▶ Language settings
- ▶ Audio settings
- ▶ Content removal settings
- ▶ Customizations

3. 切換自定義詞彙，然後從下拉菜單中選擇一個自定義詞彙。

▼ Customizations

Custom vocabulary [Info](#)  
A custom vocabulary improves the accuracy of recognizing words and phrases specific to your use case.

**Vocabulary selection**  
The vocabularies shown here are based on your language settings. You can choose up to one vocabulary per language. You can also [create a new vocabulary](#). [↗](#)

Choose a vocabulary ▼

請加入您要套用至串流的其他設定。

4. 您現在已準備好轉錄串流的準備好使用。選取 [開始串流] 並開始說話。若要結束聽寫，請選取 [停止串流]。

## HTTP/2 串流

此範例會建立包含您自訂字彙的 HTTP/2 請求。如需搭配使用 HTTP/2 串流的詳細資訊 Amazon Transcribe，請參閱[設定 HTTP/2 串流](#)。如需有關特定參數和標題的更多詳細資訊 Amazon Transcribe，請參閱[StartStreamTranscription](#)。

```
POST /stream-transcription HTTP/2
host: transcribestreaming.us-west-2.amazonaws.com
X-Amz-Target: com.amazonaws.transcribe.Transcribe.StartStreamTranscription
Content-Type: application/vnd.amazon.eventstream
X-Amz-Content-Sha256: string
X-Amz-Date: 20220208T235959Z
Authorization: AWS4-HMAC-SHA256 Credential=access-key/20220208/us-west-2/transcribe/
aws4_request, SignedHeaders=content-type;host;x-amz-content-sha256;x-amz-date;x-amz-
target;x-amz-security-token, Signature=string
x-amzn-transcribe-language-code: en-US
x-amzn-transcribe-media-encoding: flac
x-amzn-transcribe-sample-rate: 16000
x-amzn-transcribe-vocabulary-name: my-first-vocabulary
transfer-encoding: chunked
```

您可以在 [API 參考中找到參數定義](#)；所有 AWS API 作業的共用參數會列在「[一般參數](#)」區段中。

## WebSocket 流

此範例會建立預先簽署的 URL，將您的自訂字彙套用至 WebSocket 串流。已加入分行符號以提高可讀性。如需搭配使用 WebSocket 串流的詳細資訊 Amazon Transcribe，請參閱[設定串 WebSocket 流](#)。如需參數的詳細資訊，請參閱[StartStreamTranscription](#)。

```
GET wss://transcribestreaming.us-west-2.amazonaws.com:8443/stream-transcription-
websocket?
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIAIOSFODNN7EXAMPLE%2F20220208%2Fus-
west-2%2Ftranscribe%2Faws4_request
&X-Amz-Date=20220208T235959Z
&X-Amz-Expires=300
&X-Amz-Security-Token=security-token
&X-Amz-Signature=string
&X-Amz-SignedHeaders=content-type%3Bhost%3Bx-amz-date
&language-code=en-US
&media-encoding=flac
&sample-rate=16000
```

```
&vocabulary-name=my-first-vocabulary
```

您可以在 [API 參考中找到參數定義](#)；所有 AWS API 作業的共用參數會列在「[一般參數](#)」區段中。

## 自訂語言模型

自訂語言模型旨在提高領域特定語音的轉錄準確性。這包括您在正常日常對話中聽到的內容以外的任何內容。例如，如果您正在轉錄科學會議的論文集，則標準轉錄不太可能識別主持人使用的許多科學術語。在這種情況下，您可以訓練自訂語言模型，以辨識您學科中使用的專業術語。

與自訂字彙不同，它會透過提供提示 (例如發音) 來增加字詞的辨識度，自訂語言模型會學習與特定字詞相關聯的上下文。這包括單詞的使用方式和時間，以及單詞與其他詞語的關係。例如，如果您使用氣候科學研究論文訓練模型，您的模型可能會了解到「浮冰」比「冰流」更可能是單詞對。

若要檢視自訂語言模型支援的語言，請參閱[支援的語言和特定語言功能](#)。請注意，如果您在要求中包含自訂語言模型，就無法啟用語言識別 (您必須指定語言代碼)。

### 特定於自訂語言模型的 API 作業

[CreateLanguageModel](#), [DeleteLanguageModel](#), [DescribeLanguageModel](#),  
[ListLanguageModels](#)

## 資料來源

您可以使用任何類型的文字資料來訓練模型。但是，您的文本內容越接近音頻內容，您的模型就越準確。因此，選擇在與音訊相同環境中使用相同術語的文字資料非常重要。

訓練模型的最佳資料是準確的成績單。這被認為是域內數據。網域內文字資料與您要轉錄的音訊具有完全相同的術語、用法和上下文。

如果您沒有準確的成績單，請使用期刊文章、技術報告、白皮書、會議論文集、說明手冊、新聞文章、網站內容，以及任何其他文字，這些文字包含與您音訊相似的內容中使用的所需詞彙。這被視為與網域相關的資料。

建立健全的自訂語言模型可能需要大量的文字資料，其中必須包含音訊中所說的術語。您最多可以 Amazon Transcribe 提供 2 GB 的文字資料來訓練模型 — 這稱為訓練資料。或者，當您沒有 (或很少) 網域內的文字記錄時，您可以提供 Amazon Transcribe 最多 200 MB 的文字資料來調整模型 — 這稱為調整資料。

## 訓練與調整資料

訓練資料的目的在於教導如Amazon Transcribe何辨識新術語，並學習使用這些術語的上下文。為了創建一個強大的模型，Amazon Transcribe可能需要大量的相關文本數據。強烈建議提供盡可能多的訓練資料，最多可達 2 GB 的限制。

調整資料的目的在於協助精簡和最佳化從訓練資料中學到的情境關係。許可，以建立自訂語言模型，不需要調整資料。

您可以決定如何最佳選擇訓練，以及選擇性地調整資料的方式。每種情況都是獨一無二的，取決於您擁有的數據類型和數量。如果缺少網域內訓練資料，建議您調整資料。

如果您選擇同時包含這兩種資料類型，請勿重疊訓練和調整資料；訓練和調整資料應該是唯一的。重疊的資料可能會偏差和歪斜您的自訂語言模型，從而影響其準確性。

作為一般指導，我們建議盡可能使用準確的網域內文字作為訓練資料。以下是一些一般案例，依偏好設定順序列出：

- 如果您有超過 10,000 個字的準確網域內成績單文字，請將其用作訓練資料。在這種情況下，不需要包括調整數據。這是許可，以訓練自訂語言模型。
- 如果您的網域內部文字成績單文字包含少於 10,000 個字，但沒有得到理想的結果，請考慮使用與網域相關的書面文字 (例如技術報告) 來增強訓練資料。在這種情況下，請保留一小部分 (10-25%) 的網域內成績單資料作為調整資料使用。
- 如果您沒有網域內的成績單文字，請將所有與網域相關的文字上傳為訓練資料。在這種情況下，轉錄文本樣式的文本比書面文本更好。這是訓練自訂語言模型最不有效的案例。

當您準備好建立模型時，請參閱[建立自訂語言模型](#)。

## 建立自訂語言模型

在建立自訂語言模型之前，您必須：

- 準備資料。數據必須以純文本格式保存，並且不能包含任何特殊字符。
- 將您的資料上傳至Amazon S3值區。建議為訓練和調整資料建立個別的資料夾。
- 確保Amazon Transcribe可以訪問您的存Amazon S3儲桶。您必須指定具有存取權限的IAM角色才能使用您的資料。

## 準備資料

您可以將所有數據編譯在一個文件中，也可以將其另存為多個文件。請注意，如果您選擇包括調整資料，則必須將其儲存在與訓練資料不同的檔案中。

無論您使用多少文本文件進行培訓或調整數據都沒關係。上傳一個包含 100,000 個字詞的文件，結果與上傳 10 個文件（含 10,000 個字）相同的結果。以最方便的方式準備文字資料。

請確定您的所有資料檔符合以下條件：

- 它們都使用與您要創建的模型相同的語言。例如，如果您想要建立以美式英文 (en-US) 轉錄音訊的自訂語言模型，則所有文字資料都必須使用美式英文。
- 它們採用 UTF-8 編碼的純文本格式。
- 它們不包含任何特殊字符或格式，例如 HTML 標記。
- 訓練資料的合併大小上限為 2 GB，而調整資料的最大大小為 200 MB。

若未符合這些條件，您的模型將失敗。

## 上傳您的資料

上傳資料之前，請先為訓練資料建立新資料夾。如果使用調整資料，請建立另一個獨立的資料夾

您的存儲桶的 URI 可能如下所示：

- `s3://DOC-EXAMPLE-BUCKET/my-model-training-data/`
- `s3://DOC-EXAMPLE-BUCKET/my-model-tuning-data/`

將您的訓練和調整資料上傳至適當的值區。

您可以稍後將更多資料新增至這些值區。但是，如果這樣做，則需要使用新資料重新建立模型。現有模型無法使用新數據進行更新。

## 允許存取您的資料

若要建立自訂語言模型，您必須指定具有存取 Amazon S3 值區之權限的 IAM 角色。如果您尚未擁有可存取放置訓練資料之 Amazon S3 值區的角色，則必須建立一個角色。建立角色之後，您可以附加原則以授與該角色權限。請勿將政策連接至使用者。

如需取得範例政策，請參閱 [Amazon Transcribe 身分型政策範例](#)。

若要瞭解如何建立新IAM身分，請參閱 [IAM身分識別 \(使用者、使用者群組和角色\)](#)。

若要進一步了解政策，請參閱：

- [中的策略和權限 IAM](#)
- [建立IAM策略](#)
- [AWS 資源的存取權管理](#)

## 建立您的自訂語言模型

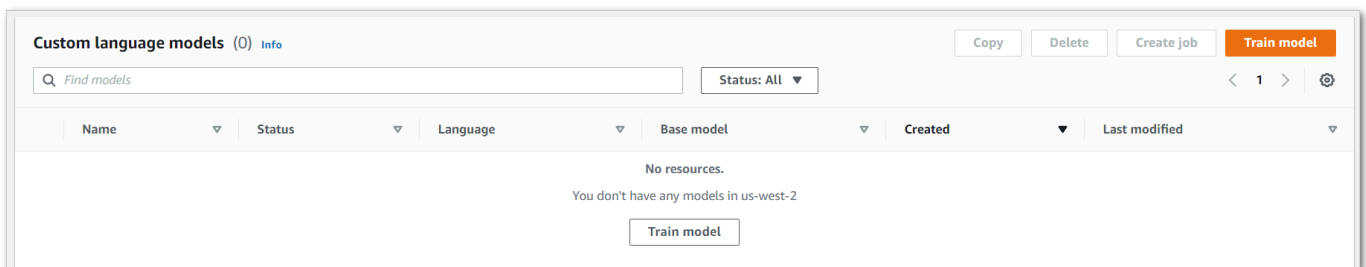
建立自訂語言模型時，您必須選擇基礎模型。有兩種基本模型選項：

- **NarrowBand**：針對取樣率小於 16,000 Hz 的音訊使用此選項。此模型類型通常用於以 8,000 赫茲記錄的電話對話。
- **WideBand**：針對取樣率大於或等於 16,000 Hz 的音訊使用此選項。

您可以使用AWS Management Console、AWS CLI或 AWS SDK 建立自訂語言模型。請參閱下列範例：

### AWS Management Console

1. 登入 [AWS Management Console](#)。
2. 在導覽窗格中，選擇自訂語言模型。這會開啟 [自訂語言模型] 頁面，您可以在其中檢視現有的自訂語言模型或訓練新的自訂語言模型。
3. 若要訓練新模型，請選取訓練模型。



這將帶您進入「火車模型」頁面。新增名稱、指定語言，然後選擇您要用於模型的基礎模型。然後，將路徑新增至訓練，並加入您的調整資料 (選擇性)。您必須包含具有存取資料之權限的IAM角色。

## Train model [Info](#)

### Model settings

**Name**

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

**Language**

Choose the language of your model.

 ▼

**Base model [Info](#)**

Choose the base model that you want to use to create your custom language model. Choose the model based on the sample rate of your source audio.

**Narrow band**

For audio that has a sample rate less than 16 KHz. Typically, this is 8 KHz audio from telephone conversations.

**Wide band**

For audio that has a sample rate of 16 KHz or greater. Typically, this is 16 KHz audio from media sources.

### Training data [Info](#)

**Training data location on S3**

Type or paste the S3 prefix for the text files that you want to use as training data, or browse to find the files that have matching S3 prefixes.

The file format must be plain text in the language that you have selected for the model. The maximum file size is 2 GB.

### Tuning data - optional [Info](#)

**Tuning data location on S3**

Type or paste the S3 prefix for the text files that you want to use as tuning data, or browse to find the files that have matching S3 prefixes.

The file format must be plain text in the language that you have selected for the model. The maximum file size is 200 MB.

### Access permissions

**IAM role [Info](#)**

Use an existing IAM role

Create an IAM role

By choosing **Train model** you are authorizing creation of this role.

**Role name**

A role that grants access to the S3 input locations.

 ▼

4. 完成所有欄位後，請選取頁面底部的 [訓練模型]。



## AWS CLI

此範例使用[create-language-model](#)指令。如需詳細資訊，請參閱 [CreateLanguageModel](#) 及 [LanguageModel](#)。

```
aws transcribe create-language-model \  
--base-model-name NarrowBand \  
--model-name my-first-language-model \  
--input-data-config S3Uri=s3://DOC-EXAMPLE-BUCKET/my-clm-training-  
data/,TuningDataS3Uri=s3://DOC-EXAMPLE-BUCKET/my-clm-tuning-  
data/,DataAccessRoleArn=arn:aws:iam::111122223333:role/ExampleRole \  
--language-code en-US
```

下面是另一個使用[create-language-model](#)命令的示例，以及創建自定義語言模型的請求主體。

```
aws transcribe create-language-model \  
--cli-input-json file://filepath/my-first-language-model.json
```

該文件 `my-first-language-model.json` 包含以下請求主體。

```
{  
  "BaseModelName": "NarrowBand",  
  "ModelName": "my-first-language-model",  
  "InputDataConfig": {  
    "S3Uri": "s3://DOC-EXAMPLE-BUCKET/my-clm-training-data/",  
    "TuningDataS3Uri": "s3://DOC-EXAMPLE-BUCKET/my-clm-tuning-data/",  
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/ExampleRole"  
  },  
  "LanguageCode": "en-US"  
}
```

## AWS SDK for Python (Boto3)

這個範例會使用建立AWS SDK for Python (Boto3)的 CLM，使用建立[語言模型](#)方法。如需詳細資訊，請參閱 [CreateLanguageModel](#) 及 [LanguageModel](#)。

如需使用 AWS SDK 的其他範例，包括特定功能、案例和跨服務範例，請參閱本章。[使用 SDK 進行 Amazon Transcribe 的代碼示例](#) [AWS](#)

```
from __future__ import print_function  
import time
```

```
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
model_name = 'my-first-language-model',
transcribe.create_language_model(
    LanguageCode = 'en-US',
    BaseModelName = 'NarrowBand',
    ModelName = model_name,
    InputDataConfig = {
        'S3Uri': 's3://DOC-EXAMPLE-BUCKET/my-clm-training-data/',
        'TuningDataS3Uri': 's3://DOC-EXAMPLE-BUCKET/my-clm-tuning-data/',
        'DataAccessRoleArn': 'arn:aws:iam::111122223333:role/ExampleRole'
    }
)

while True:
    status = transcribe.get_language_model(ModelName = model_name)
    if status['LanguageModel']['ModelStatus'] in ['COMPLETED', 'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
print(status)
```

## 更新您的自訂語言模型

Amazon Transcribe 不斷更新可用於自訂語言模型的基礎模型。為了從這些更新中受益，我們建議每 6 到 12 個月訓練一次新的自訂語言模型。

若要查看您的自訂語言模型是否使用最新的基礎模型，[DescribeLanguageModel](#) 請使用 AWS CLI 或 AWS SDK 執行要求，然後在回應中尋找 UpgradeAvailability 欄位。

如果 UpgradeAvailability 是 true，則表示您的模型未執行最新版本的基礎模型。若要在自訂語言模型中使用最新的基礎模型，您必須建立新的自訂語言模型。自訂語言模型無法升級。

## 使用自訂語言模型

建立自訂語言模型後，您可以將其包含在轉錄請求中；請參閱以下各節以取得範例。

您要包含在請求中的模型語言必須與您為媒體指定的語言代碼相符。如果語言不匹配，則您的自定義語言模型不會應用於轉錄，並且沒有警告或錯誤。

### 在批次轉錄中使用自訂語言模型

若要搭配批次轉譯使用自訂語言模型，請參閱下列範例：

## AWS Management Console

1. 登入 [AWS Management Console](#)。
2. 在功能窗格中，選擇「轉錄工作」，然後選取「建立工作」（右上角）。這會開啟 [指定工作詳細資訊] 頁面。
3. 在「Job 設定」面板中的「模型類型」下，選取「自訂語言模型」方塊。

**Job settings**

**Name**

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

**Model type** [Info](#)

Choose the type of model to use for the transcription job.

**General model**  
To use a model that is not specialized for a particular use case, choose this option. Configuration options vary between languages.

**Custom language model**  
To use a model that you trained for your specific use case, choose this option. This model has fewer configuration options than the general model.

**Language**

Choose the language of the input audio.

**Custom model selection**

Choose an existing model or [create a new one.](#)

▶ **Additional settings**

您還必須從下拉菜單中選擇輸入語言。

### Job settings

**Name**

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

**Model type** [Info](#)

Choose the type of model to use for the transcription job.

**General model**

To use a model that is not specialized for a particular use case, choose this option. Configuration options vary between languages.

**Custom language model**

To use a model that you trained for your specific use case, choose this option. This model has fewer configuration options than the general model.

**Language**

Choose the language of the input audio.

English, US (en-US) ▲

English, US (en-US)

English, AU (en-AU)

English, UK (en-GB)

Hindi, IN (hi-IN)

Spanish, US (es-US)

- 在 [自訂模型選擇] 下，從下拉式功能表中選取現有的自訂語言模型或建立新的語言模型。

在「輸入資料」面板中新增輸入檔案的Amazon S3位置。

- 選取下一步以取得其他組態選項。

選取 [建立工作] 以執行轉錄工作。

## AWS CLI

此範例使用[start-transcription-job](#)指令和ModelSettings參數搭配VocabularyName子參數。如需詳細資訊，請參閱[StartTranscriptionJob](#)及[ModelSettings](#)。

```
aws transcribe start-transcription-job \
--region us-west-2 \
--transcription-job-name my-first-transcription-job \
```

```
--media MediaFileUri=s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac \  
--output-bucket-name DOC-EXAMPLE-BUCKET \  
--output-key my-output-files/ \  
--language-code en-US \  
--model-settings LanguageModelName=my-first-language-model
```

以下是使用[start-transcription-job](#)命令的另一個範例，以及包含該工作的自訂語言模型的要求主體。

```
aws transcribe start-transcription-job \  
--region us-west-2 \  
--cli-input-json file://my-first-model-job.json
```

檔案 `my-first-model-job.json` 包含下列要求主體。

```
{  
  "TranscriptionJobName": "my-first-transcription-job",  
  "Media": {  
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"  
  },  
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",  
  "OutputKey": "my-output-files/",  
  "LanguageCode": "en-US",  
  "ModelSettings": {  
    "LanguageModelName": "my-first-language-model"  
  }  
}
```

## AWS SDK for Python (Boto3)

這個範例會使用 `ModelSettings` 引數 AWS SDK for Python (Boto3) 來包含自訂語言模型，並使用[開始工作方法的引數](#)。如需詳細資訊，請參閱 [StartTranscriptionJob](#) 及 [ModelSettings](#)。

如需使用 AWS SDK 的其他範例，包括特定功能、案例和跨服務範例，請參閱本[使用 SDK 進行 Amazon Transcribe 的代碼示例 AWS](#)章。

```
from __future__ import print_function  
import time  
import boto3  
transcribe = boto3.client('transcribe', 'us-west-2')  
job_name = "my-first-transcription-job"  
job_uri = "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
```

```
transcribe.start_transcription_job(  
    TranscriptionJobName = job_name,  
    Media = {  
        'MediaFileUri': job_uri  
    },  
    OutputBucketName = 'DOC-EXAMPLE-BUCKET',  
    OutputKey = 'my-output-files/',  
    LanguageCode = 'en-US',  
    ModelSettings = {  
        'LanguageModelName': 'my-first-language-model'  
    }  
)  
  
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)
```

## 在串流轉錄中使用自訂語言模型

若要搭配串流轉譯使用自訂語言模型，請參閱下列範例：

### AWS Management Console

1. 登入[AWS Management Console](#)。
2. 在瀏覽窗格中，選擇 Real-time transcription (即時轉錄)。向下捲動至「自訂」，如果此欄位已最小化，請展開此欄位。

## Real-time transcription [Info](#)

See how Amazon Transcribe creates a text copy of speech in real time. Choose **Start streaming** and talk.

### Transcription

[Download full transcript](#) [Start streaming](#)

Transcription output Current language: English, US

Choose **Start streaming** to begin a real-time transcription of what you speak into your microphone

00:00 of 15:00 min audio stream

- ▶ **Language settings**
- ▶ **Audio settings**
- ▶ **Content removal settings**
- ▶ **Customizations**

3. 開啟 [自訂語言模型]，然後從下拉式功能表中選取模型。

▼ **Customizations**

- Custom vocabulary** [Info](#)  
A custom vocabulary improves the accuracy of recognizing words and phrases specific to your use case.
- Partial results stabilization** [Info](#)  
Configure Amazon Transcribe to present results that don't change as it processes the transcription output from your stream.
- Custom language model** [Info](#)  
Select the model you want to use with this streaming session.

**Custom model selection** [Info](#)  
The models shown here are based on your language settings. You can choose up to one model per language. You can also [create a new model](#).

Choose model ▼

新增您要套用至串流的設定。

4. 您現在已準備好轉錄串流。選取 [開始串流] 並開始說話。若要結束聽寫，請選取 [停止串流]。

## HTTP/2 串流

此範例會建立包含您自訂語言模型的 HTTP/2 要求。如需搭配使用 HTTP/2 串流的詳細資訊 Amazon Transcribe，請參閱[設定 HTTP/2 串流](#)。如需特定參數和標頭的詳細資訊 Amazon Transcribe，請參閱[StartStreamTranscription](#)。

```
POST /stream-transcription HTTP/2
host: transcribestreaming.us-west-2.amazonaws.com
X-Amz-Target: com.amazonaws.transcribe.Transcribe.StartStreamTranscription
Content-Type: application/vnd.amazon.eventstream
X-Amz-Content-Sha256: string
X-Amz-Date: 20220208T235959Z
Authorization: AWS4-HMAC-SHA256 Credential=access-key/20220208/us-west-2/transcribe/
aws4_request, SignedHeaders=content-type;host;x-amz-content-sha256;x-amz-date;x-amz-
target;x-amz-security-token, Signature=string
x-amzn-transcribe-language-code: en-US
x-amzn-transcribe-media-encoding: flac
x-amzn-transcribe-sample-rate: 16000
x-amzn-transcribe-language-model-name: my-first-language-model
transfer-encoding: chunked
```

您可以在 [API 參考中找到參數定義](#)；所有 AWS API 作業的共用參數會列在「[一般參數](#)」區段中。

## WebSocket 流

此範例會建立預先簽署的 URL，將您的自訂語言模型套用至 WebSocket 串流。已加入分行符號以提高可讀性。如需搭配使用 WebSocket 串流的詳細資訊 Amazon Transcribe，請參閱[設定串 WebSocket 流](#)。如需參數的詳細資訊，請參閱[StartStreamTranscription](#)。

```
GET wss://transcribestreaming.us-west-2.amazonaws.com:8443/stream-transcription-
websocket?
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIAIOSFODNN7EXAMPLE%2F20220208%2Fus-
west-2%2Ftranscribe%2Faws4_request
&X-Amz-Date=20220208T235959Z
&X-Amz-Expires=300
&X-Amz-Security-Token=security-token
&X-Amz-Signature=string
&X-Amz-SignedHeaders=content-type%3Bhost%3Bx-amz-date
&language-code=en-US
&media-encoding=flac
&sample-rate=16000
```



```
&language-model-name=my-first-language-model
```

您可以在 [API 參考中找到參數定義](#)；所有 AWS API 作業的共用參數會列在「[一般參數](#)」區段中。

## 使用自訂詞彙篩選器刪除、遮罩或標記單字

自訂字彙篩選器是一個文字檔案，其中包含您要在轉錄輸出中修改的個別單字的自訂清單。

常見的使用案例是移除攻擊性或褻瀆的詞彙；不過，自訂詞彙篩選器是完全自訂的，因此您可以選取任何您想要的字詞。例如，如果您有即將推出的新產品，您可以在會議記錄中遮罩產品名稱。在這種情況下，您可以保持利益相關者，up-to-date 同時將產品名稱保密，直到推出。

字彙篩選有三種顯示方法：maskremove、和tag。請參閱下列範例，瞭解每個範例的運作方式。

- 遮罩：以三個星號 (\*\*\*) 取代指定的文字。

```
"transcript": "You can specify a list of *** or *** words, and *** *** removes them from transcripts automatically."
```

- 刪除：刪除指定的單詞，不留任何東西在它們的位置。

```
"transcript": "You can specify a list of or words, and removes them from transcripts automatically."
```

- Tag: 為每個指定的單字增加一個 tag ("vocabularyFilterMatch": true)，但不會改變單字本身。標記允許快速的成績單替換和編輯。

```
"transcript": "You can specify a list of profane or offensive words, and amazon transcribe removes them from transcripts automatically."
```

```
...
  "alternatives": [
    {
      "confidence": "1.0",
      "content": "profane"
    }
  ],
  "type": "pronunciation",
  "vocabularyFilterMatch": true
```

當您提交轉錄請求時，您可以指定自定義詞彙過濾器和要應用的過濾方法。Amazon Transcribe 然後根據您指定的過濾方法，在文字記錄中出現時修改完全相符的字詞。

自定義詞彙過濾器可以應用於批次和串流轉錄請求。若要了解如何建立自訂字彙篩選，請參閱[建立字彙篩選器](#)。若要瞭解如何套用自訂字彙篩選器，請參閱[使用自訂詞彙篩選器](#)。

**Note**

Amazon Transcribe 自動遮罩種族敏感詞彙，但您可以聯絡 [AWS 技術 Support](#) 部門選擇退出此預設篩選器。

如需詞彙篩選的影片逐步解說，請參閱 [使用字彙篩選](#)。

**詞彙篩選特定的 API 作業**

[CreateVocabularyFilter](#), [DeleteVocabularyFilter](#), [GetVocabularyFilter](#), [ListVocabularyFilters](#), [UpdateVocabularyFilter](#)

## 建立字彙篩選器

有兩個選項可建立 Custom。

1. 使用 UTF-8 編碼將以行分隔的單字清單儲存為純文字檔案。
  - 您可以搭配 AWS Management Console、AWS CLI、或 AWS SDK 使用此方法。
  - 如果使用 AWS Management Console，您可以為自訂字彙檔提供本機路徑或 Amazon S3 URI。
  - 如果使用 AWS CLI 或 AWS SDK，您必須將自訂詞彙檔案上傳到 Amazon S3 值區，並在請求中包含 Amazon S3 URI。
2. 直接在 API 請求中包含逗號分隔文字的清單。
  - 您可以使用 [Words](#) 參數與 AWS CLI 或 AWS SDK 搭配使用此方法。

如需每種方法的範例，請參閱 [建立自訂字彙篩選器](#)

建立自訂字彙篩選器時要注意的事項：

- 單詞不區分。例如，「詛咒」和「詛咒」被處理相同。
- 只會篩選完全相符的字詞。例如，如果您的篩選器包含「宣誓」，但您的媒體包含「發誓」或「咒罵」一詞，則不會過濾這些字眼。只有「發誓」的實例被過濾。因此，您必須包含要篩選的單字的所有變體。
- 篩選條件不適用於包含在其他字詞中的文字。例如，如果自訂詞彙篩選器包含「海洋」而不是「潛艇」，則不會在成績單中變更「潛艇」。

- 每個項目只能包含一個單詞 ( 不可包含空格 )。
- 如果您將自訂詞彙篩選器儲存為文字檔案，則必須使用 UTF-8 編碼的純文字格式。
- 每個最多可以有 100 個自訂詞彙篩選條件AWS 帳戶，每個篩選器的大小最多可達 50 Kb。
- 您只能使用您的語言支援的字元。如需詳細資訊，請參閱您語言的[字元集](#)。

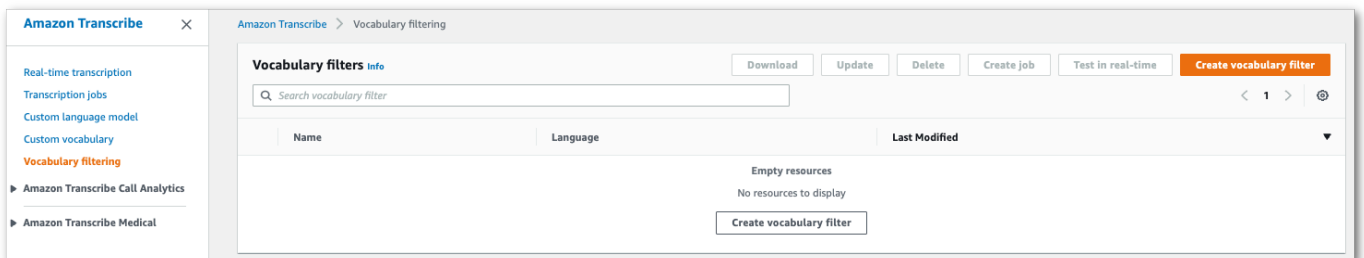
## 建立自訂字彙篩選器

若要處理自訂詞彙篩選以搭配使用Amazon Transcribe，請參閱下列範例：

### AWS Management Console

在繼續之前，請將您的自訂字彙篩選器儲存為文字 (\*.txt) 檔案。您可以選擇性地將檔案上傳至Amazon S3值區。

1. 登入 [AWS Management Console](#)。
2. 在導覽窗格中，選擇 Vocabulary filtering (詞彙篩選)。這會開啟 [字彙篩選] 頁面，您可以在其中檢視現有的自訂字彙篩選條件或建立新的字彙篩選
3. 選取 [建立字彙篩選]。



這會帶您前往 [建立字彙篩選] 頁面。輸入新的自訂字彙。

選取 [字彙輸入來源] 下的 [檔案上傳] 或 [S3 位置] 選項。然後指定自訂字彙檔案的位置。

## Create vocabulary filter Info

### Vocabulary filtering settings

**Name**

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

**Language**

English, US (en-US) ▼

**Vocabulary input source Info**

File upload

S3 location

**Vocabulary filter file location on S3**

Provide a path to the S3 location where your vocabulary filter file is stored. To find a path, go to [Amazon S3](#)

File format: txt, maximum size 50 KB.

### Tags - optional

A tag is a label you can add to a resource as metadata to help you organize, search, or filter your data. Each tag consists of a key and an optional value, in the form 'key:value'.

No tags associated with the resource.

You can add up to 50 more tags.

4. (可選) 將標籤添加到您的自定義詞彙過濾器。完成所有欄位後，請選取頁面底部的 [建立字彙篩選器]。如果處理檔案時沒有錯誤，這會帶您回到「字彙篩選」頁面。

現已準備好使用。

## AWS CLI

此範例使用指 [create-vocabulary-filter](#) 令將單字清單處理為可用的自訂字彙篩選器。如需詳細資訊，請參閱 [CreateVocabularyFilter](#)。

選項 1：您可以使用 `words` 參數將單詞列表包含到請求中。

```
aws transcribe create-vocabulary-filter \  
--vocabulary-filter-name my-first-vocabulary-filter \  

```

```
--language-code en-US \  
--words profane,offensive,Amazon,Transcribe
```

選項 2：您可以將單字清單儲存為文字檔案，並將其上傳至Amazon S3值區，然後使用 `vocabulary-filter-file-uri` 參數在要求中加入檔案的 URI。

```
aws transcribe create-vocabulary-filter \  
--vocabulary-filter-name my-first-vocabulary-filter \  
--language-code en-US \  
--vocabulary-filter-file-uri s3://DOC-EXAMPLE-BUCKET/my-vocabulary-filters/my-vocabulary-filter.txt
```

這是另一個使用 `create-vocabulary-filter` 命令的示例，以及創建自定義詞彙過濾器的請求主體。

```
aws transcribe create-vocabulary-filter \  
--cli-input-json file://filepath/my-first-vocab-filter.json
```

檔案 `my-first-vocab-filter.json` 包含下列要求主體。

選項 1：您可以使用 `Words` 參數將單詞列表包含到請求中。

```
{  
  "VocabularyFilterName": "my-first-vocabulary-filter",  
  "LanguageCode": "en-US",  
  "Words": [  
    "profane", "offensive", "Amazon", "Transcribe"  
  ]  
}
```

選項 2：您可以將單字清單儲存為文字檔案，並將其上傳至Amazon S3值區，然後使用 `VocabularyFilterFileUri` 參數在要求中加入檔案的 URI。

```
{  
  "VocabularyFilterName": "my-first-vocabulary-filter",  
  "LanguageCode": "en-US",  
  "VocabularyFilterFileUri": "s3://DOC-EXAMPLE-BUCKET/my-vocabulary-filters/my-vocabulary-filter.txt"  
}
```

**Note**

如果您VocabularyFilterFileUri在請求中包含，則無法使用Words；您必須選擇其中一個。

## AWS SDK for Python (Boto3)

此範例會使AWS SDK for Python (Boto3)用建立自訂字彙篩選器，使用[建立詞彙篩選](#)方法。如需詳細資訊，請參閱[CreateVocabularyFilter](#)。

如需使用AWS SDK 的其他範例，包括特定功能、案例和跨服務範例，請參閱本[使用 SDK 進行 Amazon Transcribe 的代碼示例 AWS](#)章。

選項 1：您可以使用Words參數將單詞列表包含到請求中。

```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
vocab_name = "my-first-vocabulary-filter"
response = transcribe.create_vocabulary_filter(
    LanguageCode = 'en-US',
    VocabularyFilterName = vocab_name,
    Words = [
        'profane', 'offensive', 'Amazon', 'Transcribe'
    ]
)
```

選項 2：您可以將單字清單儲存為文字檔案，並將其上傳至Amazon S3值區，然後使用VocabularyFilterFileUri參數在要求中加入檔案的 URI。

```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
vocab_name = "my-first-vocabulary-filter"
response = transcribe.create_vocabulary_filter(
    LanguageCode = 'en-US',
    VocabularyFilterName = vocab_name,
    VocabularyFilterFileUri = 's3://DOC-EXAMPLE-BUCKET/my-vocabulary-filters/my-
vocabulary-filter.txt'
```

)

**Note**

如果您VocabularyFilterFileUri在請求中包含，則無法使用Words；您必須選擇其中一個。

**Note**

如果您為自訂字彙篩選檔案建立新Amazon S3值區，請確定提出CreateVocabularyFilter請求的IAM角色具有存取此值區的權限。如果角色沒有正確的權限，您的請求將失敗。您可以選擇性地在請求中加入DataAccessRoleArn參數來指定IAM角色。如需中IAM角色和原則的詳細資訊Amazon Transcribe，請參閱[Amazon Transcribe 身分型政策範例](#)。

## 使用自訂詞彙篩選器

建立自訂字彙篩選條件後，您可以將其包含在轉錄請求中；請參閱下列章節以取得範例。

您要包含在請求中的自訂詞彙篩選器的語言，必須與您為媒體指定的語言代碼相符。如果您使用語言識別並指定多種語言選項，則可以針對指定的語言包含一個自訂詞彙篩選條件。如果自訂字彙篩選器的語言與您的音訊中識別的語言不符，您的篩選條件就不會套用至您的轉錄，也不會出現警告或錯誤訊息。

### 在批次轉錄中使用自訂詞彙篩選

若要將自訂字彙篩選器與批次轉錄搭配使用，請參閱下列範例：

#### AWS Management Console

1. 登入 [AWS Management Console](#)。
2. 在功能窗格中，選擇「轉錄工作」，然後選取「建立工作」（右上角）。這會開啟 [指定工作詳細資訊] 頁面。



## Specify job details [Info](#)

### Job settings

**Name**

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

**Model type** [Info](#)

Choose the type of model to use for the transcription job.

**General model**  
To use a model that is not specialized for a particular use case, choose this option. Configuration options vary between languages.

**Custom language model**  
To use a model that you trained for your specific use case, choose this option. This model has fewer configuration options than the general model.

**Language settings**

You can transcribe your audio file in a language that you specify or have Amazon Transcribe identify and transcribe it in the predominant language.

**Specific language** [Info](#)  
If you know the language spoken in your source audio, choose this option to get the most accurate results. The options available for additional processing vary between languages.

**Automatic language identification** [Info](#)  
If you don't know the language spoken in your audio files, choose this option. You have access to fewer options for additional processing than if you choose **Specific language**.

**Language**

Choose the language of the input audio.

▶ **Additional settings**

為您的工作命名並指定輸入媒體。選擇性地包含任何其他欄位，然後選擇下一步。

3. 在 [設定工作] 頁面的 [內容移除] 面板中，開啟 [字彙篩選]。

## Configure job - optional [Info](#)

### Audio settings

**Audio identification** [Info](#)  
Choose to split multi-channel audio into separate channels for transcription, or identify speakers in the input audio.

**Alternative results** [Info](#)  
Enable to view more transcription results

### Content removal

Content removal conceals information in the resulting transcript from your source audio file. Amazon Transcribe changes items in the transcript and does not modify the source audio.

**PII redaction** [Info](#)  
Label the type of PII and also mask the content with the PII entity type in the transcription output. For example, (123) 456-7890 will be masked as [PHONE].

**Vocabulary filtering** [Info](#)  
Vocabulary filtering can remove, mask or tag specified words in the final transcript.

**Filter selection**  
The vocabulary filters shown here are based on your language settings. You can choose up to one vocabulary filter per language. You can also [create a new vocabulary filter](#).

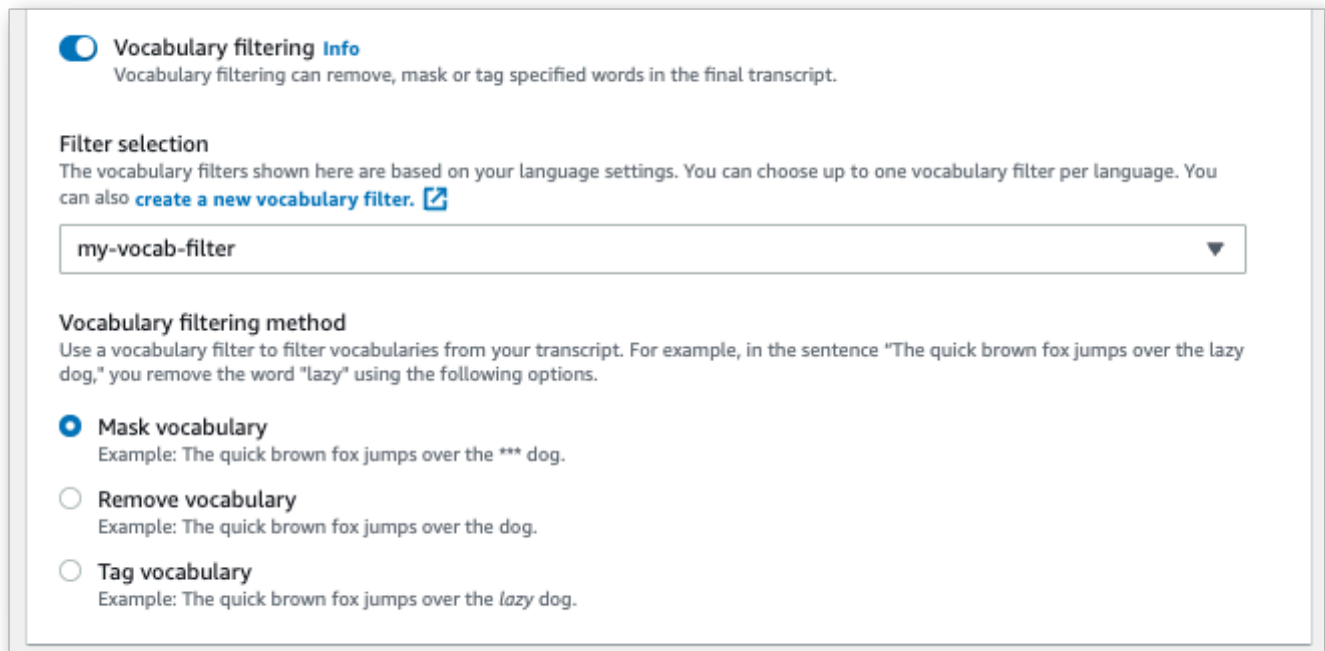
Choose a vocabulary filter ▼

### Customization

**Custom vocabulary** [Info](#)  
A custom vocabulary improves the accuracy of recognizing words and phrases specific to your use case.

Cancel Previous **Create job**

4. 從下拉菜單中選擇自定義詞彙過濾器，然後指定過濾方法。



5. 選取 [建立工作] 以執行轉錄工作。

## AWS CLI

此範例使用具有和VocabularyFilterMethod子Settings參數的[start-transcription-job](#)指令VocabularyFilterName和參數。如需詳細資訊，請參閱 [StartTranscriptionJob](#) 及 [Settings](#)。

```
aws transcribe start-transcription-job \
--region us-west-2 \
--transcription-job-name my-first-transcription-job \
--media MediaFileUri=s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac \
--output-bucket-name DOC-EXAMPLE-BUCKET \
--output-key my-output-files/ \
--language-code en-US \
--settings VocabularyFilterName=my-first-vocabulary-filter,VocabularyFilterMethod=mask
```

以下是使用[start-transcription-job](#)命令的另一個範例，以及包含該工作的自訂詞彙篩選器的要求主體。

```
aws transcribe start-transcription-job \
--region us-west-2 \
--cli-input-json file://my-first-vocabulary-filter-job.json
```

該文件 `my-first-vocabulary-filter-job.json` 包含以下請求主體。

```
{
  "TranscriptionJobName": "my-first-transcription-job",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
  },
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",
  "OutputKey": "my-output-files/",
  "LanguageCode": "en-US",
  "Settings": {
    "VocabularyFilterName": "my-first-vocabulary-filter",
    "VocabularyFilterMethod": "mask"
  }
}
```

### AWS SDK for Python (Boto3)

此範例會使 AWS SDK for Python (Boto3) 用包含自訂字彙篩選器，並使用 [啟動工作](#) 方法的 `Settings` 引數。如需詳細資訊，請參閱 [StartTranscriptionJob](#) 及 [Settings](#)。

如需使用 AWS SDK 的其他範例，包括特定功能、案例和跨服務範例，請參閱本 [使用 SDK 進行 Amazon Transcribe 的代碼示例](#) AWS 章。

```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
job_name = "my-first-transcription-job"
job_uri = "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
transcribe.start_transcription_job(
    TranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'DOC-EXAMPLE-BUCKET',
    OutputKey = 'my-output-files/',
    LanguageCode = 'en-US',
    Settings = {
        'VocabularyFilterName': 'my-first-vocabulary-filter',
        'VocabularyFilterMethod': 'mask'
    }
)
```

```
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)
```

## 在串流轉錄中使用自訂詞彙篩選器

若要將自訂字彙篩選器與串流轉錄搭配使用，請參閱下列範例：

### AWS Management Console

1. 登入 [AWS Management Console](#)。
2. 在瀏覽窗格中，選擇 Real-time transcription (即時轉錄)。向下捲動至「內容移除設定」，如果此欄位已最小化，請展開此欄位。

## Real-time transcription [Info](#)

See how Amazon Transcribe creates a text copy of speech in real time. Choose **Start streaming** and talk.

**Transcription**

[Download full transcript](#)
[Start streaming](#)

Transcription output Current language: English, US

Choose **Start streaming** to begin a real-time transcription of what you speak into your microphone

00:00 of 15:00 min audio stream

- ▶ **Language settings**
- ▶ **Audio settings**
- ▶ **Content removal settings**
- ▶ **Customizations**

- 開啟 [字彙篩選]。從下拉式功能表中選擇自訂詞彙篩選條件，並指定過濾方法。

▼ **Content removal settings**

**Vocabulary filtering** [Info](#)  
 Vocabulary filtering removes, masks, or tags words that you specify in your vocabulary filter. Choose a vocabulary filter to see an example.

**Filter selection**  
 The vocabulary filters shown here are based on your language settings. You can choose up to one vocabulary filter per language. You can also [create a new vocabulary filter](#).

**Vocabulary filtering method** [Info](#)  
 Use a vocabulary filter to filter vocabularies from your transcript. For example, in the sentence "The quick brown fox jumps over the lazy dog," you remove the word "lazy" using the following options.

**Mask vocabulary**  
 Example: The quick brown fox jumps over the \*\*\* dog.

**Remove vocabulary**  
 Example: The quick brown fox jumps over the dog.

**Tag vocabulary**  
 Example: The quick brown fox jumps over the *lazy* dog.

新增您要套用至串流的設定。

4. 您現在已準備好轉錄串流。選取 [開始串流] 並開始說話。若要結束聽寫，請選取 [停止串流]。

## HTTP/2 串流

此範例會建立 HTTP/2 請求，其中包含您的自訂詞彙篩選器和篩選方法。如需使用 HTTP/2 串流搭配使用的詳細資訊 Amazon Transcribe，請參閱[設定 HTTP/2 串流](#)。如需有關特定參數和標題的更多詳細資訊 Amazon Transcribe，請參閱[StartStreamTranscription](#)。

```
POST /stream-transcription HTTP/2
host: transcribestreaming.us-west-2.amazonaws.com
X-Amz-Target: com.amazonaws.transcribe.Transcribe.StartStreamTranscription
Content-Type: application/vnd.amazon.eventstream
X-Amz-Content-Sha256: string
X-Amz-Date: 20220208T235959Z
Authorization: AWS4-HMAC-SHA256 Credential=access-key/20220208/us-west-2/transcribe/
aws4_request, SignedHeaders=content-type;host;x-amz-content-sha256;x-amz-date;x-amz-
target;x-amz-security-token, Signature=string
x-amzn-transcribe-language-code: en-US
x-amzn-transcribe-media-encoding: flac
x-amzn-transcribe-sample-rate: 16000
x-amzn-transcribe-vocabulary-filter-name: my-first-vocabulary-filter
x-amzn-transcribe-vocabulary-filter-method: mask
transfer-encoding: chunked
```

您可以在 API [參考中找到參數定義](#)；所有 AWS API 作業的共用參數會列在「[一般參數](#)」區段中。

## WebSocket 流

此範例會建立預先簽署的 URL，將您的自訂字彙篩選器套用至 WebSocket 串流。已加入分行符號以提高可讀性。如需搭配使用 WebSocket 串流的詳細資訊 Amazon Transcribe，請參閱[設定串 WebSocket 流](#)。如需參數的詳細資訊，請參閱[StartStreamTranscription](#)。

```
GET wss://transcribestreaming.us-west-2.amazonaws.com:8443/stream-transcription-
websocket?
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIAIOSFODNN7EXAMPLE%2F20220208%2Fus-
west-2%2Ftranscribe%2Faws4_request
&X-Amz-Date=20220208T235959Z
&X-Amz-Expires=300
&X-Amz-Security-Token=security-token
&X-Amz-Signature=string
&X-Amz-SignedHeaders=content-type%3Bhost%3Bx-amz-date
```

```
&language-code=en-US  
&media-encoding=flac  
&sample-rate=16000  
&vocabulary-filter-name=my-first-vocabulary-filter  
&vocabulary-filter-method=mask
```

您可以在 [API 參考](#) 中找到參數定義；所有 AWS API 作業的共用參數會列在「[一般參數](#)」區段中。



## 偵測有毒語音

有毒語音檢測旨在幫助中度涉及的社交媒體平台peer-to-peer對話，例如在線遊戲和社交聊天平台。使用有毒的言語可能會嚴重不利於個人，同伴團體和社區。標記有害語言可以幫助組織保持對話內幕，並維護一個安全和包容的在線環境，以使用戶自由創建，共享和參與。

Amazon Transcribe毒性偵測功能利用音訊和文字提示，在七個類別中識別和分類基於語音的有毒內容，包括性騷擾、仇恨言論、威脅、虐待、褻瀆、侮辱和圖形。除了文字之外,Amazon Transcribe毒性檢測使用語音提示，例如音調和音高來磨練語音中的有毒意圖。這是標準內容審核系統的改進，該系統旨在僅關注特定術語，而無需考慮意圖。

Amazon Transcribe標記和分類有毒的語音，這將必須手動處理的數據量降到最低。這使內容管理員能夠快速有效地管理其平台上的話語。

有毒的演講類別包括：

- 褻瀆: 包含不禮貌、粗俗或令人反感的單詞、短語或首字母縮略詞的語音。
- 仇恨言論：基於身份（例如種族，種族，性別，宗教，性取向，能力和國籍）批評，侮辱，譴責或使某個人或團體不人道化的演講。
- 性：使用直接或間接引用身體部位，身體特徵或性別來表示性興趣，活動或喚醒的語音。
- 侮辱: 包括貶低、羞辱、嘲笑、侮辱或貶低語言的言語。這種類型的語言也被標記為欺凌。
- 暴力或威脅: 語音，其中包括試圖對個人或團體造成疼痛，傷害或敵意的威脅。
- 图形: 使用視覺上描述性和令人不愉快的生動圖像的語音。這種類型的語言通常是故意冗長的，以擴大收件人的不適感。
- 騷擾或辱罵: 旨在影響收件人心理健康的言論，包括貶低和客觀性的術語。這種語言也被標記為騷擾。

毒性偵測會分析語音區段（自然暫停之間的語音），並將信賴度分數指派給這些區段。置信度分數是介於 0 和 1 之間的值。較高的可信度分數表示內容在相關類別中是有毒語音的可能性越大。您可以使用這些可信度分數，針對您的使用案例設定適當的毒性偵測閾值。

### Note

毒性檢測僅適用於美國英語的批量轉錄(en-US)。

檢視[範例輸出](#)以 JSON 格式顯示。

## 使用有毒語音檢測

### 在批量轉錄中使用有毒語音檢測

若要將有毒語音偵測與批次轉錄搭配使用，請參閱下列範例：

#### AWS Management Console

1. 登入 [AWS Management Console](#)。
2. 在導覽窗格中，選擇「轉錄工作」，然後選取建立工作（右上）。這將打開指定工作詳情頁面。

The screenshot shows the 'Specify job details' page in the AWS Management Console. The page is titled 'Specify job details' with an 'Info' link. It is divided into several sections:

- Job settings**
  - Name**: A text input field containing 'MyTranscriptionJob'. Below it, a note states: 'The name can be up to 200 characters long. Valid characters are a-z, A-Z, 0-9, . (period), \_ (underscore), and - (hyphen).'
  - Model type** (with an 'Info' link): A section with the instruction 'Choose the type of model to use for the transcription job.' It contains two radio button options:
    - General model** (selected): 'To use a model that is not specialized for a particular use case, choose this option. Configuration options vary between languages.'
    - Custom language model**: 'To use a model that you trained for your specific use case, choose this option. This model has fewer configuration options than the general model.'
  - Language settings**: A section with the instruction 'You can transcribe your audio file in a language that you specify or have Amazon Transcribe identify and transcribe it in the predominant language.' It contains two radio button options:
    - Specific language** (selected, with an 'Info' link): 'If you know the language spoken in your source audio, choose this option to get the most accurate results. The options available for additional processing vary between languages.'
    - Automatic language identification** (with an 'Info' link): 'If you don't know the language spoken in your audio files, choose this option. You have access to fewer options for additional processing than if you choose **Specific language**.'
  - Language**: A dropdown menu with 'English, US (en-US)' selected and a downward arrow.
- Additional settings**: A section with a right-pointing arrow icon.

- 在「」指定工作詳情頁面中，您也可以根據需要啟用 PII 密文。請注意，毒性偵測不支援其他列出的選項。選取下一步。這將帶您到設定工作-選擇性頁面。在音訊設定」面板中，選取毒性檢測。

### Audio settings

**Audio identification** [Info](#)  
Choose to split multi-channel audio into separate channels for transcription, or partition speakers in the input audio.

---

**Alternative results** [Info](#)  
Enable to view more transcription results

---

**Toxicity detection** [Info](#)  
Flag toxic speech in your transcription output

### Content removal

Content removal conceals information in the resulting transcript from your source audio file. Amazon Transcribe changes items in the transcript and does not modify the source audio.

**PII redaction** [Info](#)  
Label the type of PII and also mask the content with the PII entity type in the transcription output. For example, (123) 456-7890 will be masked as [PHONE].

---

**Vocabulary filtering** [Info](#)  
Vocabulary filtering can remove, mask or tag specified words in the final transcript.

### Customization

**Custom vocabulary** [Info](#)  
A custom vocabulary improves the accuracy of recognizing words and phrases specific to your use case.

Cancel

- 选择建立工作執行您的轉錄工作。
- 轉錄工作完成後，您可以從下載轉錄工作詳細資訊頁面中的下拉式選單。

## AWS CLI

此範例使用[start-transcription-job](#)命令和ToxicityDetection參數。如需詳細資訊，請參閱[StartTranscriptionJob](#)及[ToxicityDetection](#)。

```
aws transcribe start-transcription-job \  
--region us-west-2 \  
--transcription-job-name my-first-transcription-job \  
--media MediaFileUri=s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac \  
--output-bucket-name DOC-EXAMPLE-BUCKET \  
--output-key my-output-files/ \  
--language-code en-US \  
--toxicity-detection ToxicityCategories=ALL
```

這是另一個使用[start-transcription-job](#)命令，以及包括毒性檢測的請求主體。

```
aws transcribe start-transcription-job \  
--region us-west-2 \  
--cli-input-json file://filepath/my-first-toxicity-job.json
```

該文件my-first-toxicity-job.json包含下列要求主體。

```
{  
  "TranscriptionJobName": "my-first-transcription-job",  
  "Media": {  
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"  
  },  
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",  
  "OutputKey": "my-output-files/",  
  "LanguageCode": "en-US",  
  "ToxicityDetection": [  
    {  
      "ToxicityCategories": [ "ALL" ]  
    }  
  ]  
}
```

## AWS SDK for Python (Boto3)

此範例使用AWS SDK for Python (Boto3)以啟用ToxicityDetection為[開始轉錄工作](#)方法。如需詳細資訊，請參閱 [StartTranscriptionJob](#) 及 [ToxicityDetection](#)。

如需使用AWSSDK (包括特定功能、案例和跨服務範例) 請參閱[使用 SDK 進行 Amazon Transcribe 的代碼示例 AWS](#)章節。

```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
job_name = "my-first-transcription-job"
job_uri = "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
transcribe.start_transcription_job(
    TranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'DOC-EXAMPLE-BUCKET',
    OutputKey = 'my-output-files/',
    LanguageCode = 'en-US',
    ToxicityDetection = [
        {
            'ToxicityCategories': ['ALL']
        }
    ]
)

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)
```

## 範例輸出

有毒語音在您的轉錄輸出中被標記並分類。有毒語音的每個執行個體都會分類並指派一個置信度分數(介於 0 和 1 之間的值)。較大的可信度值表示內容在指定類別中是有毒語音的可能性就越大。

## 範例輸出

以下是 JSON 格式的範例輸出，顯示分類的有毒語音與相關聯的可信度分數。

```
{
  "jobName": "my-toxicity-job",
  "accountId": "111122223333",
  "results": {
    "transcripts": [...],
    "items": [...],
    "toxicity_detection": [
      {
        "text": "What the * are you doing man? That's why I didn't want to play
with your * . man it was a no, no I'm not calming down * man. I well I spent I spent
too much * money on this game.",
        "toxicity": 0.7638,
        "categories": {
          "profanity": 0.9913,
          "hate_speech": 0.0382,
          "sexual": 0.0016,
          "insult": 0.6572,
          "violence_or_threat": 0.0024,
          "graphic": 0.0013,
          "harassment_or_abuse": 0.0249
        },
        "start_time": 8.92,
        "end_time": 21.45
      },
      Items removed for brevity
      {
        "text": "What? Who? What the * did you just say to me? What's your
address? What is your * address? I will pull up right now on your * * man. Take your *
back to , tired of this **.",
        "toxicity": 0.9816,
        "categories": {
          "profanity": 0.9865,
          "hate_speech": 0.9123,
          "sexual": 0.0037,
          "insult": 0.5447,
          "violence_or_threat": 0.5078,
          "graphic": 0.0037,
          "harassment_or_abuse": 0.0613
        },
      },
    ]
  }
}
```

```
        "start_time": 43.459,  
        "end_time": 54.639  
      },  
    ],  
  },  
  ...  
  "status": "COMPLETED"  
}
```

# 編輯或識別資訊訊訊訊訊訊息碼

密文可用來遮罩或移除敏感內容，以個人識別資訊 (PII) 的形式，從您的成績單中。Amazon Transcribe 可編輯的 PII 類型會因批次和串流轉錄而有所不同。若要檢視每種轉錄方法的 PII 清單，請參閱[在批次工作中修訂 PII](#)和[在即時串流中修訂或識別 PII](#)。使用串流轉錄時，您也可以選擇標記 PII 而不加以編輯；如需輸出範例，請參閱[PII 識別輸出範例](#)

啟用密文後，您可以選擇只產生密文的成績單，或同時產生經過編輯的成績單和未編輯的成績單。如果您選擇只產生密文字記錄，請注意，您的媒體是唯一儲存完整交談的地方。如果您刪除原始媒體，則沒有未編輯 PII 的記錄。因此，除了編輯過的成績單之外，還要生成未編輯的成績單可能是謹慎的。

若要深入瞭解使用批次轉譯的 PII 編修，請參閱：[在批次工作中修訂 PII](#)。

若要深入瞭解 PII 編輯或使用串流轉譯進行識別，請參閱：[在即時串流中修訂或識別 PII](#)。

## Important

密文功能旨在識別和移除敏感資料。但是，由於機器學習的預測性質，Amazon Transcribe 可能無法識別並移除記錄中所有敏感資料的執行個體。我們強烈建議您檢閱任何已編輯的輸出，以確保其符合您的需求。

密文功能不符合美國 Health 保險流通與責任法案 (例如 1996 美國健康保險流通與責任法案 (HIPAA) 與美國健康保險流通與責任法案 (HIPAA) 與醫療隱私法案 (HIPAA)

如需密文功能 Amazon Transcribe 的影片逐步解說，請參閱[使用內容密文來識別及編輯 PII](#)。

## 在批次工作中修訂 PII

[PII] 在批次轉錄工作期間，從成績單中編輯個人識別資訊 (PII) 時，Amazon Transcribe 會在成績單的主要文字內文中取代每個識別的 PII 執行個體。您也可以檢視在轉錄輸出 word-for-word 部分中編輯的 PII 類型。如需了解輸出範例，請參閱[編輯輸出的示例 \(批處理\)](#)。

美國英文 (en) 和美國西班牙文 (en-US) 可使用批次轉錄進行編輯。es-US 修訂與[語言識別](#)不相容。

已編輯和未編輯的成績單都儲存在相同的輸出值區中。Amazon S3 Amazon Transcribe 將它們存儲在您指定的存儲桶中，或者存儲在服務管理的默認 Amazon S3 值區中。



## Amazon Transcribe 可識別批次轉錄的 PII 類型

PII 類型	描述
ADDRESS	實體地址，例如 100 Main Street, Anytown, USA 或 Suite #12, Building 123。地址可以包括街道、建築物、位置、城市、州、國家、縣、郵遞區號、分區、鄰里等。
ALL	修訂或識別此表格中列出的所有 PII 類型。
BANK_ACCOUNT_NUMBER	美國銀行帳戶號碼 這些通常介於 10-12 位數間，但是僅存在最後 4 位數字時，Amazon Transcribe 也可以識別銀行帳戶號碼。
BANK_ROUTING	美國銀行帳戶的分行代碼。這些通常長 9 位數字，但是只有最後 4 位數字存在時，Amazon Transcribe 也可以識別路由號碼。
CREDIT_DEBIT_CVV	VISA 上存在的 3 位數信用卡驗證碼 ( CVV ) MasterCard，以及發現信用卡和借記卡。在美國運通信用卡或簽帳卡，是一個 4 位數的數字代碼。
CREDIT_DEBIT_EXPIRY	信用卡或簽帳卡到期日 該數字通常為 4 位數，格式為月/年或 MM/YY。例如，Amazon Transcribe 可以識別到期日期，例如 2021 年 1 月 21 日、2021 年 1 月和 1 月。
CREDIT_DEBIT_NUMBER	信用卡或簽帳卡號碼。這些數字的長度可以從 13 到 16 位數不等，但是當僅存在最後 4 位數字時，Amazon Transcribe 也可以識別信用卡或借記卡號碼。
EMAIL	電子郵件位址，例如 efua.owusu@email.com。
NAME	個人的姓名。此實體類型不包含標題，例如先生、太太、小姐或博士 Amazon Transcribe 不會將此實體類型套用至屬於組織或地址一部分的

PII 類型	描述
	名稱。例如，將多約翰組織 Amazon Transcribe 識別為一個組織，而李四街作為一個地址。
PHONE	電話號碼。此實體類型還包括傳真和呼叫器號碼。
PIN	4 位數的個人識別號碼 (PIN)，允許某人存取其銀行帳戶資訊。
SSN	社會安全號碼 (SSN) 是一個 9 位數的號碼，發給美國公民，永久居民和臨時工作居民。Amazon Transcribe 當只有最後 4 位數字時，還可以識別社會安全號碼。

您可以使用 AWS Management Console、AWS CLI 或 AWS SDK 開始批次轉錄工作。

## AWS Management Console

1. 登入 [AWS Management Console](#)。
2. 在導覽窗格中，選擇轉錄作業，然後選擇建立作業(右上角)。這將開啟指定作業詳細資訊頁面。
3. 在指定作業詳細資訊頁面中填入想要的欄位後，選擇下一步以前往設定工作 - 選用頁面。您可以在此處找到具 PII 修訂切換的內容移除面板。

## Configure job - *optional* [Info](#)

### Audio settings

**Audio identification** [Info](#)  
Choose to split multi-channel audio into separate channels for transcription, or identify speakers in the input audio.

**Alternative results** [Info](#)  
Enable to view more transcription results

### Content removal

Content removal conceals information in the resulting transcript from your source audio file. Amazon Transcribe changes items in the transcript and does not modify the source audio.

**PII redaction** [Info](#)  
Label the type of PII and also mask the content with the PII entity type in the transcription output. For example, (123) 456-7890 will be masked as [PHONE].

4. 選 PII 修訂後，您可以選擇要修訂的所有 PII 類型。如果您選擇在工作輸出中包含未修訂的文字記錄方塊，您也可以選擇有未修訂的文字記錄。

### Content removal

Content removal conceals information in the resulting transcript from your source audio file. Amazon Transcribe changes items in the transcript and does not modify the source audio.

**PII redaction** [Info](#)  
 Label the type of PII and also mask the content with the PII entity type in the transcription output. For example, (123) 456-7890 will be masked as [PHONE].

**Include unredacted transcript in job output**  
 Returns unredacted version of the transcript in addition to the redacted version.

Select PII entity types (11 of 11 selected)

Select All

**Financial (6 of 6 selected)**

<input checked="" type="checkbox"/> BANK_ACCOUNT_NUMBER	<input checked="" type="checkbox"/> BANK_ROUTING	<input checked="" type="checkbox"/> CREDIT_DEBIT_NUMBER
<input checked="" type="checkbox"/> CREDIT_DEBIT_CVV	<input checked="" type="checkbox"/> CREDIT_DEBIT_EXPIRY	<input checked="" type="checkbox"/> PIN

**Personal (5 of 5 selected)**

<input checked="" type="checkbox"/> NAME	<input checked="" type="checkbox"/> ADDRESS	<input checked="" type="checkbox"/> PHONE
<input checked="" type="checkbox"/> EMAIL	<input checked="" type="checkbox"/> SSN	

---

**Vocabulary filtering** [Info](#)  
 Vocabulary filtering can remove, mask or tag specified words in the final transcript.

5. 選擇建立作業以執行轉錄作業。

## AWS CLI

此範例使用指 [start-transcription-job](#) 令和 `content-redaction` 參數。如需詳細資訊，請參閱 [StartTranscriptionJob](#) 及 [ContentRedaction](#)。

```
aws transcribe start-transcription-job \
--region us-west-2 \
--transcription-job-name my-first-transcription-job \
--media MediaFileUri=s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac \
--output-bucket-name DOC-EXAMPLE-BUCKET \
--output-key my-output-files/ \
--language-code en-US \
```

```
--content-redaction
RedactionType=PII,RedactionOutput=redacted,PiiEntityTypes=NAME,ADDRESS,BANK_ACCOUNT_NUMBER
```

以下是使用該[start-transcription-job](#)方法的另一個範例，而要求主體會針對該工作編輯 PII。

```
aws transcribe start-transcription-job \
--region us-west-2 \
--cli-input-json file://filepath/my-first-redaction-job.json
```

該文件 `my-first-redaction-job.json` 包含以下請求主體。

```
{
  "TranscriptionJobName": "my-first-transcription-job",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
  },
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",
  "OutputKey": "my-output-files/",
  "LanguageCode": "en-US",
  "ContentRedaction": {
    "RedactionOutput": "redacted",
    "RedactionType": "PII",
    "PiiEntityTypes": [
      "NAME",
      "ADDRESS",
      "BANK_ACCOUNT_NUMBER"
    ]
  }
}
```

## AWS SDK for Python (Boto3)

此範例會使用使用[開始](#)工作方法的 `ContentRedaction` 引數 AWS SDK for Python (Boto3) 來編輯內容。如需詳細資訊，請參閱 [StartTranscriptionJob](#) 及 [ContentRedaction](#)。

如需使用 AWS SDK 的其他範例，包括特定功能、案例和跨服務範例，請參閱本章。[使用 SDK 進行 Amazon Transcribe 的代碼示例 AWS](#)

```
from __future__ import print_function
import time
import boto3
```

```
transcribe = boto3.client('transcribe', 'us-west-2')
job_name = "my-first-transcription-job"
job_uri = "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
transcribe.start_transcription_job(
    TranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'DOC-EXAMPLE-BUCKET',
    OutputKey = 'my-output-files/',
    LanguageCode = 'en-US',
    ContentRedaction = {
        'RedactionOutput': 'redacted',
        'RedactionType': 'PII',
        'PiiEntityTypes': [
            'NAME', 'ADDRESS', 'BANK_ACCOUNT_NUMBER'
        ]
    }
)

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)
```

### Note

批次工作的 PII 編修僅支援以下範圍 AWS 區域：亞太區域 (香港)、亞太區域 (孟買)、亞太區域 (首爾)、亞太區域 (新加坡)、亞太區域 (雪梨)、亞太區域 (東京)、 GovCloud (美國西部)、加拿大 (中東)、歐洲 (法蘭克福)、歐洲 (愛爾蘭)、歐洲 (聖倫敦)、歐洲 (巴黎)、中東 (巴林)、美國南東 (保羅) (維吉尼亞北部)、美國東部 (俄亥俄)、美國西部 (奧勒岡) 和美國西部 (加利佛尼亞北部)。

## 在即時串流中修訂或識別 PII

從串流轉錄中修訂個人身分識別資訊 (PII) 時，Amazon Transcribe 會使用 [PII] 取代文字記錄中每個已識別的 PII 執行個體。

可用於串流轉錄的其他選項是 PII 識別。當您啟用 PII 識別時，會將轉錄結果中的 PII Amazon Transcribe 標示在物件下方。Entities 如需輸出範例，請參閱 [編輯流輸出示例](#) 和 [PII 識別輸出範例](#)。

使用以下英語方言可以使用以下英語方言對 PII 進行編輯和識別：澳大利亞語 ( en-AU )，英國 ( )，美國 ( en-GB ) 和西班牙美國方言 ( en-US )。es-US

串流工作的 PII 識別和修訂只有在完成音訊區段的轉錄時才會執行。

Amazon Transcribe 可辨識串流轉錄的 PII 類型

PII 類型	描述
ADDRESS	實體地址，例如 100 Main Street, Anytown, USA 或 Suite #12, Building 123。地址可以包括街道、建築物、位置、城市、州、國家、縣、郵遞區號、分區、鄰里等。
ALL	修訂或識別此表格中列出的所有 PII 類型。
BANK_ACCOUNT_NUMBER	美國銀行帳戶號碼 這些通常介於 10-12 位數間，但是僅存在最後 4 位數字時，Amazon Transcribe 也可以識別銀行帳戶號碼。
BANK_ROUTING	美國銀行帳戶的分行代碼。這些通常長 9 位數字，但是只有最後 4 位數字存在時，Amazon Transcribe 也可以識別路由號碼。
CREDIT_DEBIT_CVV	VISA 上存在的 3 位數信用卡驗證碼 ( CVV ) MasterCard，以及發現信用卡和借記卡。在美國運通信用卡或簽帳卡，是一個 4 位數的數字代碼。
CREDIT_DEBIT_EXPIRY	信用卡或簽帳卡到期日 該數字通常為 4 位數，格式為月/年或 MM/YY。例如，Amazon Transcribe 可以識別到期日期，例如 2021 年 1 月 21 日、2021 年 1 月和 1 月。
CREDIT_DEBIT_NUMBER	信用卡或簽帳卡號碼。這些數字的長度可以從 13 到 16 位數不等，但是當僅存在最後 4 位數字

PII 類型	描述
	時，Amazon Transcribe 也可以識別信用卡或借記卡號碼。
EMAIL	電子郵件位址，例如 efua.owusu@email.com。
NAME	個人的姓名。此實體類型不包含標題，例如先生、太太、小姐或博士 Amazon Transcribe 不會將此實體類型套用至屬於組織或地址一部分的名稱。例如，將多約翰組織 Amazon Transcribe 識別為一個組織，而李四街作為一個地址。
PHONE	電話號碼。此實體類型還包括傳真和呼叫器號碼。
PIN	4 位數的個人識別號碼 (PIN)，允許某人存取其銀行帳戶資訊。
SSN	社會安全號碼 (SSN) 是一個 9 位數的號碼，發給美國公民，永久居民和臨時工作居民。Amazon Transcribe 當只有最後 4 位數字時，還可以識別社會安全號碼。

您可以使用 AWS Management Console、WebSocket 或 HTTP/2 開始串流轉錄。

## AWS Management Console

1. 登入 [AWS Management Console](#)。
2. 在導覽窗格中，選擇即時轉錄。向下捲動至內容移除設定，如果此欄位已最小化，請展開此欄位。



**Transcription** Download full transcript Start streaming

Transcription output Current language: English, US

Choose **Start streaming** to begin a real-time transcription of what you speak into your microphone

00:00 of 15:00 min audio stream

- ▶ Language settings
- ▶ Audio settings
- ▶ Content removal settings
- ▶ Customizations

### 3. 開啟 PII 識別與編輯。

▼ Content removal settings

Vocabulary filtering [Info](#)  
Vocabulary filtering removes, masks, or tags words that you specify in your vocabulary filter. Choose a vocabulary filter to see an example.

PII Identification & redaction [Info](#)  
Identify or redact one or more types of personally identifiable information (PII) in your transcript

### 4. 選擇 僅限識別 或 識別與編輯，然後選擇您要在文字記錄中識別或編輯的 PII 實體類型。

▼ Content removal settings

Vocabulary filtering [Info](#)  
Vocabulary filtering removes, masks, or tags words that you specify in your vocabulary filter. Choose a vocabulary filter to see an example.

PII Identification & redaction [Info](#)  
Identify or redact one or more types of personally identifiable information (PII) in your transcript

Select PII detection type

Identification only  
Label the type of PII identified but not redact it in the transcription output

Identification & redaction  
Label the type of PII and also mask the content with the PII entity type in the transcription output. For example, (123)456-7890 will be masked as [PHONE]

Select PII entity types (22 of 22 selected)

Select All

Financial (6 of 6 selected)

<input checked="" type="checkbox"/> BANK_ACCOUNT_NUMBER	<input checked="" type="checkbox"/> BANK_ROUTING	<input checked="" type="checkbox"/> CREDIT_DEBIT_NUMBER
<input checked="" type="checkbox"/> CREDIT_DEBIT_CVV	<input checked="" type="checkbox"/> CREDIT_DEBIT_EXPIRY	<input checked="" type="checkbox"/> PIN

Personal (8 of 8 selected)

<input checked="" type="checkbox"/> NAME	<input checked="" type="checkbox"/> ADDRESS	<input checked="" type="checkbox"/> PHONE
<input checked="" type="checkbox"/> EMAIL	<input checked="" type="checkbox"/> SSN	<input checked="" type="checkbox"/> PASSPORT_NUMBER
<input checked="" type="checkbox"/> DRIVER_ID	<input checked="" type="checkbox"/> AGE	

Digital footprint (7 of 7 selected)

<input checked="" type="checkbox"/> URL	<input checked="" type="checkbox"/> USERNAME	<input checked="" type="checkbox"/> PASSWORD
<input checked="" type="checkbox"/> AWS_ACCESS_KEY	<input checked="" type="checkbox"/> AWS_SECRET_KEY	<input checked="" type="checkbox"/> IP_ADDRESS
<input checked="" type="checkbox"/> MAC_ADDRESS		

Other (1 of 1 selected)

DATE\_TIME

► Customizations

5. 您現在已準備好轉錄串流。選擇開始串流並開始說話。若要結束聽寫，選擇停止串流。

## WebSocket 流

此範例會建立在串流中使用 PII 密文 (或 PII 識別) 的預先簽署 URL。WebSocket 已加入分行符號以提高可讀性。如需搭配使用 WebSocket 串流的詳細資訊 Amazon Transcribe，請參閱[設定串 WebSocket 流](#)。如需參數詳細資訊，請參閱[StartStreamTranscription](#)。

```
GET wss://transcribestreaming.us-west-2.amazonaws.com:8443/stream-transcription-
websocket?
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIAIOSFODNN7EXAMPLE%2F20220208%2Fus-
west-2%2Ftranscribe%2Faws4_request
&X-Amz-Date=20220208T235959Z
&X-Amz-Expires=300
&X-Amz-Security-Token=security-token
```

```

&X-Amz-Signature=string
&X-Amz-SignedHeaders=content-type%3Bhost%3Bx-amz-date
&language-code=en-US
&media-encoding=flac
&sample-rate=16000
&pii-entity-types=NAME,ADDRESS
&content-redaction-type=PII (or &content-identification-type=PII)

```

您不能在相同的請求中同時使用 `content-identification-type` 和 `content-redaction-type`。

您可以在 [API 參考中找到參數定義](#)；所有 AWS API 作業的共用參數會列在「[一般參數](#)」區段中。

## HTTP/2 串流

此範例會建立啟用 PII 識別或 PII 修訂的 HTTP/2 請求。如需使用 HTTP/2 串流搭配使用的詳細資訊 Amazon Transcribe，請參閱。[設定 HTTP/2 串流](#) 如需有關特定參數和標題的更多詳細資訊 Amazon Transcribe，請參閱[StartStreamTranscription](#)。

```

POST /stream-transcription HTTP/2
host: transcribestreaming.us-west-2.amazonaws.com
X-Amz-Target: com.amazonaws.transcribe.Transcribe.StartStreamTranscription
Content-Type: application/vnd.amazon.eventstream
X-Amz-Content-Sha256: string
X-Amz-Date: 20220208T235959Z
Authorization: AWS4-HMAC-SHA256 Credential=access-key/20220208/us-west-2/transcribe/
aws4_request, SignedHeaders=content-type;host;x-amz-content-sha256;x-amz-date;x-amz-
target;x-amz-security-token, Signature=string
x-amzn-transcribe-language-code: en-US
x-amzn-transcribe-media-encoding: flac
x-amzn-transcribe-sample-rate: 16000
x-amzn-transcribe-content-identification-type: PII (or x-amzn-transcribe-content-
redaction-type: PII)
x-amzn-transcribe-pii-entity-types: NAME,ADDRESS
transfer-encoding: chunked

```

您不能在相同的請求中同時使用 `content-identification-type` 和 `content-redaction-type`。

您可以在 [API 參考中找到參數定義](#)；所有 AWS API 作業的共用參數會列在「[一般參數](#)」區段中。

**Note**

串流 PII 編修僅支援下列項目 AWS 區域：亞太區域 (首爾)、亞太區域 (雪梨)、亞太區域 (東京)、加拿大 (中部)、歐洲 (法蘭克福)、歐洲 (愛爾蘭)、歐洲 (倫敦)、美國東部 (維吉尼亞北部)、美國東部 (俄亥俄) 和美國西部 (奧勒岡)。

## PII 編修和識別輸出範例

下列範例顯示批次和串流工作的編輯輸出，以及串流工作的 PII 識別。

使用內容密文的轉錄工作會產生兩種類型的 confidence 值。自動語音辨識 (ASR) 可信度表示具有 type 類型的 pronunciation 或 punctuation 為特定用語。在下面的成績單輸出中，單詞 Good 具 confidence 有 1.0。這個可信度值表示 Amazon Transcribe 百分之百確信這份成績單中所說的單字是「好」。`[PII]` 標籤的 confidence 值是它標記為修訂的語音確實是 PII 的可信度。在下列成績單輸出中，confidence 的 0.9999 Amazon Transcribe 表示 99.99% 確定其在成績單中編輯的實體為 PII。

### 編輯輸出的示例 ( 批處理 )

```
{
  "jobName": "my-first-transcription-job",
  "accountId": "111122223333",
  "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": "NAME",
                        "category": "PII"
                    }
                ]
            }
        ],
        "type": "pronunciation"
    },
    Items removed for brevity
],
},
"status": "COMPLETED"
}

```

以下是未編輯的成績單以供比較：

```

{
  "jobName": "job id",
  "accountId": "111122223333",
  "isRedacted": false,
  "results": {
    "transcripts": [
      {

```

```

    "transcript": "Good morning, everybody. My name is Mike, and today I
feel like
my Social
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": "0.9999",
        "content": "Mike",
      }
    ],
    "type": "pronunciation"
  },
  Items removed for brevity
],
},
"status": "COMPLETED"
}

```

## 編輯流輸出示例

```
{
```

```
"TranscriptResultStream": {
  "TranscriptEvent": {
    "Transcript": {
      "Results": [
        {
          "Alternatives": [
            {
              "Transcript": "my name is [NAME]",
              "Items": [
                {
                  "Content": "my",
                  "EndTime": 0.3799375,
                  "StartTime": 0.0299375,
                  "Type": "pronunciation"
                },
                {
                  "Content": "name",
                  "EndTime": 0.5899375,
                  "StartTime": 0.3899375,
                  "Type": "pronunciation"
                },
                {
                  "Content": "is",
                  "EndTime": 0.7899375,
                  "StartTime": 0.5999375,
                  "Type": "pronunciation"
                },
                {
                  "Content": "[NAME]",
                  "EndTime": 1.0199375,
                  "StartTime": 0.7999375,
                  "Type": "pronunciation"
                }
              ]
            },
            {
              "Entities": [
                {
                  "Content": "[NAME]",
                  "Category": "PII",
                  "Type": "NAME",
                  "StartTime": 0.7999375,
                  "EndTime": 1.0199375,
                  "Confidence": 0.9989
                }
              ]
            }
          ]
        }
      ]
    }
  }
}
```





```
        "StartTime": 0.5999375,
        "Type": "pronunciation"
    },
    {
        "Content": "mike",
        "EndTime": 0.9199375,
        "StartTime": 0.7999375,
        "Type": "pronunciation"
    }
],
"Entities": [
    {
        "Content": "mike",
        "Category": "PII",
        "Type": "NAME",
        "StartTime" : 0.7999375,
        "EndTime" : 1.0199375,
        "Confidence": 0.9989
    }
]
},
"EndTime": 1.02,
"IsPartial": false,
"ResultId": "12345a67-8bc9-0de1-2f34-a5b678c90d12",
"StartTime": 0.0199375
}
}
}
}
```

## 組建影片字幕

Amazon Transcribe 支援用作視訊字幕的 WebVTT (\*.vtt) 和 SubRip (\*.srt) 輸出。您可以在設定批次視訊轉錄工作時，選取一種或兩種檔案類型。使用字幕功能時，系統會產生您選擇的字幕檔案和一般的成績單檔案 (包含其他資訊)。字幕和轉錄文件輸出到相同的目的地。

字幕會在朗讀文字的同時顯示，並保持可見，直到自然暫停或揚聲器完成通話為止。請注意，如果您在轉錄要求中啟用字幕，而您的音訊不包含語音，則不會建立字幕檔案。

### Important

Amazon Transcribe 字幕輸出使用的 0 預設起始索引，這與使用較廣泛的值不同 1。如果您需要的開始索引 1，您可以使用 [OutputStartIndex](#) 參數在 AWS Management Console 或 API 請求中指定此索引。

使用不正確的開始索引可能會導致與其他服務的相容性錯誤，因此在建立字幕之前，請務必確認您需要的起始索引。如果您不確定要使用哪個值，我們建議您選擇 1。如需詳細 [Subtitles](#) 資訊，請參閱。

支持字幕的功能：

- 內容編輯 — 任何經過編輯的內容在字幕和一般成績單輸出檔案中都會反映為 PII 「」。音頻不會改變。
- 詞彙過濾器 — 字幕文件是從轉錄文件生成的，因此您在標準轉錄輸出中過濾的任何單詞也會在字幕中過濾。篩選後的內容會顯示為空白字元，或 \*\*\* 在您的成績單和字幕檔案中顯示。音頻不會改變。
- 揚聲器日記 — 如果給定的字幕段中有多個揚聲器，則使用破折號來區分每個揚聲器。這同時適用於 WebVTT 和 SubRip 格式；例如：
  - --人 1 說的文字
  - --人 2 說的文字

字幕文件存儲在與轉錄輸出相同的 Amazon S3 位置。

請參閱 [Amazon 轉錄視頻小吃：在不編寫任何代碼的情況下創建視頻字幕](#) 以獲取創建字幕的視頻演練。

## 產生字幕檔

您可以使用AWS Management Console、AWS CLI或 AWSSDK 建立字幕檔案；請參閱下列範例：

### AWS Management Console

1. 登入 [AWS Management Console](#)。
2. 在功能窗格中，選擇「轉錄工作」，然後選取「建立工作」（右上角）。這會開啟 [指定工作詳細資訊] 頁面。字幕選項位於「輸出資料」面板中。
3. 選擇字幕檔案所需的格式，然後為開始索引選擇一個值。請注意，Amazon Transcribe預設值為0，但1使用更廣泛。如果您不確定要使用哪個值，我們建議您選擇1，因為這樣可能會改善與其他服務的相容性。

**Output data**

Output data location type info [Info](#)

Service-managed S3 bucket  
The output will be removed after 90 days when the job expires.

Customer specified S3 bucket  
The output will not be removed from bucket even after the job expires.

Subtitle file format [Info](#)

SRT (SubRip)

VTT (WebVTT)

Specify the start index

0 ▼

4. 填入您要包含在 [指定工作詳細資訊] 頁面上的任何其他欄位，然後選取 [下一步]。這會帶您前往 [設定工作-選擇性] 頁面。
5. 選取 [建立工作] 以執行轉錄工作。

### AWS CLI

此範例使用指 [start-transcription-job](#) 令和 [Subtitles](#) 參數。如需詳細資訊，請參閱 [StartTranscriptionJob](#) 及 [Subtitles](#)。

```
aws transcribe start-transcription-job \  
--region us-west-2 \  

```

```
--transcription-job-name my-first-transcription-job \  
--media MediaFileUri=s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac \  
--output-bucket-name DOC-EXAMPLE-BUCKET \  
--output-key my-output-files/ \  
--language-code en-US \  
--subtitles Formats=vtt,srt,OutputStartIndex=1
```

這是另一個使用 [start-transcription-job](#) 命令的示例，以及為該工作添加字幕的請求主體。

```
aws transcribe start-transcription-job \  
--region us-west-2 \  
--cli-input-json file://my-first-subtitle-job.json
```

該文件 *my-first-subtitle-job.json* 包含以下請求主體。

```
{  
  "TranscriptionJobName": "my-first-transcription-job",  
  "Media": {  
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"  
  },  
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",  
  "OutputKey": "my-output-files/",  
  "LanguageCode": "en-US",  
  "Subtitles": {  
    "Formats": [  
      "vtt", "srt"  
    ],  
    "OutputStartIndex": 1  
  }  
}
```

## AWS SDK for Python (Boto3)

[此範例會AWS SDK for Python \(Boto3\)使用使用Subtitles引數來新增字幕。](#) 如需詳細資訊，請參閱 [StartTranscriptionJob](#) 及 [Subtitles](#)。

如需使用 AWS SDK 的其他範例，包括特定功能、案例和跨服務範例，請參閱本章。 [使用 SDK 進行 Amazon Transcribe 的代碼示例 AWS](#)

```
from __future__ import print_function  
import time
```

```
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
job_name = "my-first-transcription-job"
job_uri = "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
transcribe.start_transcription_job(
    TranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'DOC-EXAMPLE-BUCKET',
    OutputKey = 'my-output-files/',
    LanguageCode = 'en-US',
    Subtitles = {
        'Formats': [
            'vtt', 'srt'
        ],
        'OutputStartIndex': 1
    }
)

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)
```

# 使用通話分析以分析客服中心音訊

使用 Amazon Transcribe 呼叫分析深入了解客戶代理互動。通話分析專為客服中心音訊而設計，可自動為您提供與每個通話和每個參與者相關的寶貴資料。您還可以在整個通話過程中精通特定時機的資料。例如，您可以將通話前幾秒鐘的客戶情緒與通話的後四分之一進行比較，以了解您的客服人員是否提供正面的體驗。[下一節](#)列出其他使用案例範例。

通話分析可用於通話後和即時轉錄。如果您要轉錄位於 Amazon S3 值區中的檔案，表示您正在執行通話後轉譯。如果您要轉錄音訊串流，您要執行的是即時轉錄。這兩種轉錄方法提供不同的通話分析見解和功能。如需每種方法的詳細資訊，請參閱 [通話後分析](#) 和 [即時通話分析](#)。

透過即時通話分析轉錄，您還可以在請求中包含[通話後分析](#)。您的通話後分析記錄會儲存在您在請求中指定的 Amazon S3 值區中。請參閱 [具有即時轉錄功能的通話後分析](#)，了解詳細資訊。

## 通話分析專用的 API 操作

通話

後：[CreateCallAnalyticsCategory](#)、[DeleteCallAnalyticsCategory](#)、[DeleteCallAnaly](#)

實時：[StartCallAnalyticsStreamTranscription](#)，

[StartCallAnalyticsStreamTranscriptionWebSocket](#)

## 常用案例

通話後轉錄：

- 監控一段時間內的問題頻率：使用[通話分類](#)以識別文字記錄中重複的關鍵字。
- 深入了解您的客戶服務體驗：利用[通話特徵](#) (非通話時間、通話時間、中斷、語音響度、通話速度) 和情緒分析，判斷通話期間是否適當解決客戶問題。
- 確保法規遵循或遵守公司政策：設定公司特定問候語或免責聲明的[關鍵字和片語](#)，以確認您的客服人員符合法規要求。
- 改善客戶個人資料的處理：在轉錄輸出或音訊檔案中使用 [PII 修訂](#) 功能，以協助保護客戶隱私。
- 改善員工培訓：使用標準 (情緒、非通話時間，中斷、通話速度) 以標記可用作正面或負面客戶互動範例的文字記錄。
- 衡量員工在提供正面客戶體驗的效率：使用[情緒分析](#)以衡量您的客服人員是否可以在進行通話時，將負面的客戶情緒轉化為正面情緒。

- 改善資料組織：根據[自訂類別](#) (包括關鍵字和片語、情緒、通話時間和中斷)，標記和排序通話。
- 使用生成式 AI 摘要整理通話的重要層面：使用[生成式通話摘要](#)取得簡潔的文字記錄摘要，當中包括一些重要元件，例如通話中討論的問題、行動項目及結果。

即時轉錄：

- 即時緩解升級：設定關鍵片語的[即時警示](#)，例如客戶說「與管理員談談」，以在通話開始升級時標記通話。您可以使用即時類別配對以建立即時警示。
- 改善客戶資料的處理：在轉錄輸出中使用 [PII 識別](#) 或 [PII 修訂](#)，以協助保護客戶隱私。
- 識別自訂關鍵字和片語：使用 [自訂類別](#) 以標記通話中的特定關鍵字。
- 自動識別問題：使用自動[問題偵測](#)以取得通話中識別的所有問題的簡短摘要。
- 衡量員工在提供正面客戶體驗的效率：使用[情緒分析](#)以衡量您的客服人員是否可以在進行通話時，將負面的客戶情緒轉化為正面情緒。
- 設定客服人員協助：使用您選擇的見解，為您的客服人員提供主動協助以解決客戶通話。請參閱[使用 Amazon 語言 AI 服務的聯絡中心的即時通話分析和客服人員協助](#)，了解詳細資訊。

若要將「呼叫分析」中可用的功能與「用於」Amazon Transcribe 和「Amazon Transcribe 醫療」的功能進行比較，請參閱[功能表格](#)。

若要開始使用，請參閱 [開始通話後分析轉錄](#) 和 [開始即時通話分析轉錄](#)。通話分析輸出與標準轉錄作業類似，但包含其他分析資料。若要檢視範例輸出，請參閱 [通話後分析輸出](#) 和 [即時通話分析輸出](#)。

## 考量事項和其他資訊

使用通話分析前，請注意：

- 通話分析僅支援雙聲道音訊，其中客服人員出現在一個聲道上，而客戶則出現在第二個聲道上。
- 通話後分析工作始終會啟用 [Job 佇列](#)，因此您只能同時進行 100 個通話分析工作。如果要請求增加配額，請使用 [AWS Service Quotas](#)。
- 通話後分析工作的輸入檔案不得大於 500 MB，且必須少於 4 小時。請注意，對於某些壓縮的非 WAV 音頻文件格式，文件大小限制可能會更小。
- 如果使用類別，您必須先建立所有想要的類別，然後才能開始進行通話分析轉錄。任何新類別都無法套用至現有的轉錄。若要了解如何建立新類別，請參閱 [建立通話後轉錄的類別](#) 和 [建立即時轉錄的類別](#)。

- 某些通話分析配額 Amazon Transcribe 與 Amazon Transcribe 醫療配額不同；如需詳細資訊，請參閱 [AWS 一般參考資料](#)。

**i** 透過 Machine Learning AWS earning 部落格深入瞭解

若要進一步了解通話分析選項，請參閱：

- [使用 Amazon 語言 AI 服務，發布您的聯絡中心通話分析](#)
- [使用 Amazon 語言 AI 服務，提供您的聯絡中心即時通話分析和客服人員協助](#)

要查看示例呼叫分析輸出和功能，請參閱我們的 [GitHub 演示](#)。我們還提供 [JSON 到 Word 文檔](#) 應用程序，以將您的成績單轉換為 easy-to-read 格式。

## 區域可用性和配額

以下支援呼叫分析 AWS 區域：

區域	轉錄類型
ap-northeast-1(東京)	post-call, real-time
ap-northeast-2(首爾)	post-call, real-time
ap-south-1(孟買)	post-call
ap-southeast-1(新加坡)	post-call
ap-southeast-2(雪梨)	post-call, real-time
ca-central-1(加拿大中部)	post-call, real-time
eu-central-1(法蘭克福)	post-call, real-time
eu-west-2(倫敦)	post-call, real-time
us-east-1(維吉尼亞北部)	post-call, real-time
us-west-2(奧勒岡)	post-call, real-time



請注意，[Amazon Transcribe](#)、[Amazon Transcribe Medical](#) 和通話分析的區域支援不同。

如需每個服務區域的端點，請參閱《AWS 一般參考》中的[服務端點](#)。

如需與轉錄相關的配額清單，請參閱《AWS 一般參考》中的 [Service Quotas](#)。有些配額可以按請求變更。如果可調整欄包含「是」，您可以要求增加。若要執行此作業，請選擇提供的連結。

## 通話後分析

通話分析提供通話後分析，這對於監視客戶服務趨勢非常有用。

通話後轉錄提供以下見解：

- [通話特徵](#)，包括通話時間、非通話時間、發言者響度、中斷、通話速度、問題、結果和行動項目
- [生成式通話摘要](#)，可建立完整通話的簡潔摘要
- 使用規則搭配[自訂分類](#)，您可以使用這些規則以精通特定關鍵字和條件
- 文字記錄和音訊檔案的 [PII 修訂](#)
- 每個來電者在通話中不同點的發言者[情緒](#)

## 通話後見解

本節詳細介紹可用於通話後分析轉錄的見解。

### 通話特徵

通話特徵功能使用以下標準以衡量客服人員與客戶間的互動品質：

- **中斷**：衡量是否以及何時一個參與者打斷另一個參與者的中間句。頻繁的干擾可能與粗魯或憤怒有關，並且可能與一或兩個參與者的負面情緒相關。
- **響度**：衡量每位參與者的發言音量。使用此指標以查看通話者或客服人員是否大聲說話或大叫，這通常表示感到生氣。此指標以 0 到 100 的範圍表示為標準化值 (指定區段中每秒語音的語音層級)，其中較高的值表示聲音越大。
- **非通話時間**：衡量沒有語音的時間段。使用此指標以查看是否長時間靜音，例如客服人員保留客戶通話過長時間。
- **通話速度**：衡量兩位參與者的說話速度。如果一個參與者說得太快，理解力可能會受到影響。此指標單位為每分鐘字數。
- **通話時間**：衡量每位參與者通話期間發言的時間量 (以毫秒為單位)。使用此指標以協助識別一位參與者是否主導通話，或對話是否達到平衡。

- 問題、結果和行動項目：從通話文字記錄中識別問題、結果和行動項目。

這是[輸出範例](#)。

## 生成式通話摘要

生成呼叫摘要會建立整個呼叫的簡明摘要，擷取關鍵元件，例如呼叫原因、解決問題所採取的步驟，以及後續步驟。

使用生成式通話摘要，您可以：

- 減少通話期間和通話後手動記錄的需求。
- 提高客服人員的效率，因為人員可花更多時間與佇列等待的來電方交談，而非進行通話後續工作。
- 加速主管檢閱，因為通話摘要檢閱的速度比整個文字記錄更快。

若要使用生成式通話摘要搭配通話後分析作業，請參閱[啟用生成式通話摘要](#)。如需輸出範例，請參閱[生成式通話摘要輸出範例](#)。生成式通話摘要會另行定價 (請參閱[定價頁面](#))。

### Note

生成式通話摘要目前已在 us-east-1 和 us-west-2 中推出。以下英語方言支援此功能：澳洲英文 (en-AU)、英國英文 (en-GB)、印度英文 (en-IN)、愛爾蘭英文 (en-IE)、蘇格蘭英文 (en-AB)、美國英文 (en-US) 和威爾斯英文 (en-WL)。

## 自訂分類

使用通話分類以標記通話中的關鍵字、片語、情緒或動作。我們的分類選項可以幫助您分類升級，例如發生許多中斷的負面情緒通話，或將通話安排到特定類別，例如公司部門。

您可以新增至類別的條件包括：

- 非通話時間：客戶或客服人員商都不說話的時間段。
- 中斷：客戶或客服人員中斷其他人發言時。
- 客戶或客服人員情緒：客戶或客服人員在指定時段內的感受。如果指定時段內至少有 50% 的對話轉動 (兩位發言者 back-and-forth 之間) 符合指定的情緒，則 Amazon Transcribe 會將情緒視為相符項目。

- **關鍵字或片語**：根據精確片語，符合部分轉錄。例如，如果您為「我要與經理談談」片語設定篩選，Amazon Transcribe 會篩選該精確片語。

您也可以標記先前條件的反向標記 (通話時間、缺少中斷、不存在情緒，以及缺少特定片語)。

這是[輸出範例](#)。

如需有關類別的詳細資訊，或了解如何建立新類別，請參閱 [建立通話後轉錄的類別](#)。

## 敏感資料修訂

敏感資料修訂會取代文字記錄和音訊檔案中的個人身分識別資訊 (PII)。修訂的文字記錄會使用 [PII] 取代原始文字；修訂後的音訊檔案會以靜音取代口頭的個人資訊。此參數有助於保護客戶資訊。

### Note

美國英文 ( ) 和美國西班牙文 (en-US) 支援通話後 PII 編修。es-US

若要檢視使用此功能修訂的 PII 清單，或了解有關使用 Amazon Transcribe 修訂的更多資訊，請參閱 [編輯或識別資訊訊訊訊訊訊訊息碼](#)。

這是[輸出範例](#)。

## 情緒分析

情緒分析預估客戶和客服人員整個通話中的感受。此指標同時表示為定量值 (範圍從 5 到 -5) 和定性值 (positive、neutral、mixed 或 negative)。每季和每次通話提供定量值；每回合提供定性值。

此指標有助於識別您的客服人員是否可以在通話結束時讓不安的客戶感到滿意。

情緒分析有效 out-of-the-box，因此不支援自訂，例如模型訓練或自訂類別。

這是[輸出範例](#)。

## 建立通話後轉錄的類別

通話後分析支援建立自訂類別，讓您量身打造文字記錄分析，以適當滿足您的特定業務需求。

您可以根據需要，建立任意數量的類別，以涵蓋一系列不同的情況。您必須為您建立的每個類別建立 1 到 20 個規則。每個規則都依據四個標準之一：中斷、關鍵字、非通話時間或情緒。如需將這些條件

與 [CreateCallAnalyticsCategory](#) 操作搭配使用的詳細資訊，請參閱 [通話後分析類別的規則條件](#) 一節。

如果媒體中的內容符合您在指定類別中指定的所有規則，Amazon Transcribe 將標示輸出為該類別。如需 JSON 輸出中類別相符項目的範例，請參閱 [通話分類輸出](#)。

以下是您可以使用自訂類別執行的幾個範例：

- 隔離具有特定特徵的通話，例如以負面客戶情緒結束的通話
- 標記和追蹤特定的關鍵字集以識別客戶問題的趨勢
- 監控合規性，例如通話的前幾秒鐘內，客服人員說出 (或省略) 特定片語
- 標記許多客服人員中斷和負面客戶情緒的通話，以取得對客戶體驗的見解
- 比較多個類別以衡量相關性，例如分析使用歡迎片語的客服人員是否與正面的客戶情緒相關

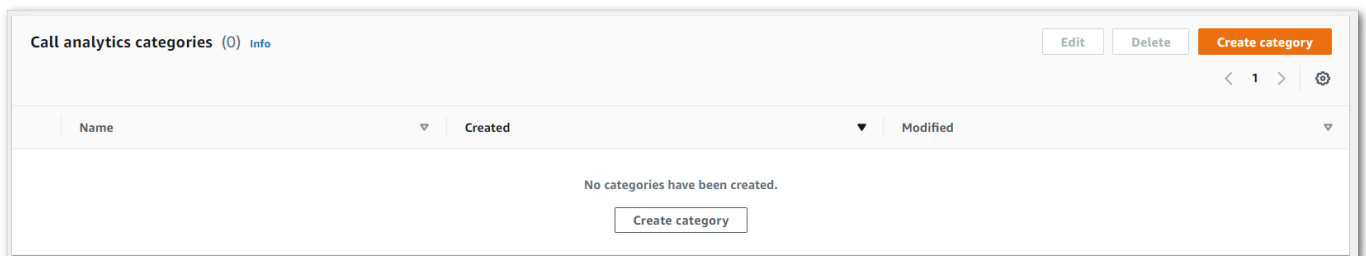
## 通話後與即時類別

建立新類別時，您可以指定建立為通話後分析類別 (POST\_CALL) 或建立為即時通話分析類別 (REAL\_TIME)。如果未指定選項，則預設會將類別建立為通話後類別。完成通話後分析轉錄後，可以在您的輸出中使用通話後分析類別相符的項目。

若要建立通話後分析的新類別，您可以使用 AWS Management Console、AWS CLI 或 AWS SDK；請參閱下列範例：

### AWS Management Console

1. 在功能窗格的下 Amazon Transcribe，選擇 [Amazon Transcribe 通話分析]。
2. 選擇通話分析類別，將引導您進入通話分析類別頁面。選擇建立類別。



3. 您現在已進入建立類別頁面。輸入類別的名稱，然後在類別類型下拉式功能表中，選擇「批次通話分析」。

**Category settings**

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

**Category type [Info](#)**

**Batch call analytics**

**Real time call analytics**

Use a template (recommended)  
 Use a template to edit predefined rules.

Create from scratch  
 If you know the rules that you want to define, choose this option.

**Template type [Info](#)**  
 Choose the template for the category that most closely matches the one you want to create.

4. 您可以選擇範本，以建立您的類別，也可以從頭開始建立範本。

如果使用範本：選擇使用範本 (建議選項)，選擇您要的範本，然後選擇建立類別。

**Category settings**

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

**Category type [Info](#)**

**Category creation method [Info](#)**

Use a template (recommended)  
 Use a template to edit predefined rules.

Create from scratch  
 If you know the rules that you want to define, choose this option.

**Template type [Info](#)**  
 Choose the template for the category that most closely matches the one you want to create.

Non-talk time exceeds 5 minutes for the whole call

Customer sentiment is negative for the last 5 minutes of the call

Agent spoke over the customer more than 15 seconds for the entire call

5. 如果建立自訂類別：選擇從頭開始建立。

## Create category [Info](#)

### Category settings

**Category name**

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

**Category creation method [Info](#)**

Use a template (recommended)  
Use a template to edit predefined rules.

Create from scratch  
If you know the rules that you want to define, choose this option.

### Rules

All the rule conditions must be met for a transcription job to be classified in this category.

▼ Rule 1 Delete rule

**Rule type [Info](#)**  
Choose the rule that you want to define.

**Add rule**

You can add up to 19 more rules.

6. 使用下拉式功能表將規則新增至您的類別。每個類別最多可新增 20 個規則。

**Rules**  
All the rule conditions must be met for a transcription job to be classified in this category.

▼ **Rule 1** Delete rule

When no word has been spoken for more than **5 minute(s)** during the **entire call**.

**Rule type** [Info](#)  
Choose the rule that you want to define.

- Non-talk time** (selected)  
Trigger the rule when nothing has been said for more than the time that you specify.
- Interruption time**  
Trigger the rule when the speaker has been interrupted for more than the time that you specify.
- Transcript content match**  
Trigger the rule when the speaker says the words or phrases that you specify.
- Transcript sentiment match**  
Trigger the rule when the sentiment of the speaker matches the sentiment that you specify.

Add rule

You can add up to 19 more rules.

7. 以下是帶兩個規則的類別範例：通話期間中斷客戶超過 15 秒的客服人員，以及在通話的最後兩分鐘內，客戶或客服人員感覺到負面情緒。

### Rules

All the rule conditions must be met for a transcription job to be classified in this category.

**▼ Rule 1** Delete rule

When the duration of the interruption was more than 15 second(s) during the entire call when the speaker was agent.

**Rule type** [Info](#)  
Choose the rule that you want to define.

Interruption time ▼

**Logic** [Info](#)  
Define the conditions that must be met.

When the duration of the interruption was more than   ▼

during the  ▼

when the speaker was  ▼

AND

**▼ Rule 2** Delete rule

When the sentiment is negative during the last 2 minute(s) when the speaker was either.

**Rule type** [Info](#)  
Choose the rule that you want to define.

Transcript sentiment match ▼

**Logic** [Info](#)  
Define the conditions that must be met.

When the sentiment is  ▼

during the  ▼   ▼

when the speaker was  ▼

AND

Add rule

You can add up to 18 more rules.

8. 將規則新增至類別時，選擇建立類別。

## AWS CLI

此範例使用 [create-call-analytics-category](#) 指令。如需詳細資訊，請參閱 [CreateCallAnalyticsCategory](#)、[CategoryProperties](#) 及 [Rule](#)。

下列範例會建立帶規則的類別：

- 客戶在前 60,000 毫秒內被中斷。這些中斷的持續時間至少持續 10,000 毫秒。



- 進入通話的 10% 到 80% 間，靜音持續至少 20,000 毫秒。
- 客服人員在通話中的某個時段有負面情緒。
- 在通話的前 10,000 毫秒中，未使用「歡迎」或「您好」單詞。

此範例使用 [create-call-analytics-category](#) 指令，以及新增多個規則到類別的請求主文。

```
aws transcribe create-call-analytics-category \  
--cli-input-json file://filepath/my-first-analytics-category.json
```

檔案 my-first-analytics-category.json 包含以下請求主文。

```
{  
  "CategoryName": "my-new-category",  
  "InputType": "POST_CALL",  
  "Rules": [  
    {  
      "InterruptionFilter": {  
        "AbsoluteTimeRange": {  
          "First": 60000  
        },  
        "Negate": false,  
        "ParticipantRole": "CUSTOMER",  
        "Threshold": 10000  
      }  
    },  
    {  
      "NonTalkTimeFilter": {  
        "Negate": false,  
        "RelativeTimeRange": {  
          "EndPercentage": 80,  
          "StartPercentage": 10  
        },  
        "Threshold": 20000  
      }  
    },  
    {  
      "SentimentFilter": {  
        "ParticipantRole": "AGENT",  
        "Sentiments": [  
          "NEGATIVE"  
        ]  
      }  
    }  
  ]  
}
```

```

    }
  },
  {
    "TranscriptFilter": {
      "Negate": true,
      "AbsoluteTimeRange": {
        "First": 10000
      },
      "Targets": [
        "welcome",
        "hello"
      ],
      "TranscriptFilterType": "EXACT"
    }
  }
]
}

```

### AWS SDK for Python (Boto3)

此範例會使 AWS SDK for Python (Boto3) 用建立類別，使用 `CategoryName` 和 `Rules` 引數來建立類別。如需詳細資訊，請參閱 [CreateCallAnalyticsCategory](#)、[CategoryProperties](#) 及 [Rule](#)。

如需使用 AWS SDK 的其他範例，包括特定功能、案例和跨服務範例，請參閱本章。 [使用 SDK 進行 Amazon Transcribe 的代碼示例 AWS](#)

下列範例會建立帶規則的類別：

- 客戶在前 60,000 毫秒內被中斷。這些中斷的持續時間至少持續 10,000 毫秒。
- 進入通話的 10% 到 80% 間，靜音持續至少 20,000 毫秒。
- 客服人員在通話中的某個時段有負面情緒。
- 在通話的前 10,000 毫秒中，未使用「歡迎」或「您好」單詞。

```

from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
category_name = "my-new-category"
transcribe.create_call_analytics_category(
    CategoryName = category_name,
    InputType = POST_CALL,

```

```
Rules = [  
  {  
    'InterruptionFilter': {  
      'AbsoluteTimeRange': {  
        'First': 60000  
      },  
      'Negate': False,  
      'ParticipantRole': 'CUSTOMER',  
      'Threshold': 10000  
    }  
  },  
  {  
    'NonTalkTimeFilter': {  
      'Negate': False,  
      'RelativeTimeRange': {  
        'EndPercentage': 80,  
        'StartPercentage': 10  
      },  
      'Threshold': 20000  
    }  
  },  
  {  
    'SentimentFilter': {  
      'ParticipantRole': 'AGENT',  
      'Sentiments': [  
        'NEGATIVE'  
      ]  
    }  
  },  
  {  
    'TranscriptFilter': {  
      'Negate': True,  
      'AbsoluteTimeRange': {  
        'First': 10000  
      },  
      'Targets': [  
        'welcome',  
        'hello'  
      ],  
      'TranscriptFilterType': 'EXACT'  
    }  
  }  
]
```

```
)  
  
result = transcribe.get_call_analytics_category(CategoryName = category_name)  
print(result)
```

## 通話後分析類別的規則條件

本節概述您可以使用 [CreateCallAnalyticsCategory](#) API 操作建立的自訂 POST\_CALL 規則類型。

### 中斷配對

使用中斷 ([InterruptionFilter](#) 資料類型) 的規則設計為符合下列項目：

- 客服人員中斷客戶的範例
- 客戶中斷客服人員的範例
- 中斷其他參與者的參與者
- 缺乏中斷

以下是可搭配 [InterruptionFilter](#) 使用的參數範例：

```
"InterruptionFilter": {  
  "AbsoluteTimeRange": {  
    Specify the time frame, in milliseconds, when the match should occur  
  },  
  "RelativeTimeRange": {  
    Specify the time frame, in percentage, when the match should occur  
  },  
  "Negate": Specify if you want to match the presence or absence of interruptions,  
  "ParticipantRole": Specify if you want to match speech from the agent, the customer, or both,  
  "Threshold": Specify a threshold for the amount of time, in seconds, interruptions occurred during the call  
},
```

請參閱 [CreateCallAnalyticsCategory](#) 及 [InterruptionFilter](#)，了解這些參數與這些參數相關的有效值的詳細資訊。

### 關鍵字配對

使用關鍵字 ([TranscriptFilter](#) 資料類型) 的規則設計為符合下列項目：

- 客服人員、客戶或兩者說出的自訂單字或片語
- 客服人員、客戶或兩者都未說出的自訂單字或片語
- 在特定時間範圍內發生的自訂單字或片語

以下是可搭配 [TranscriptFilter](#) 使用的參數範例：

```
"TranscriptFilter": {
  "AbsoluteTimeRange": {
    Specify the time frame, in milliseconds, when the match should occur
  },
  "RelativeTimeRange": {
    Specify the time frame, in percentage, when the match should occur
  },
  "Negate": Specify if you want to match the presence or absence of your custom keywords,
  "ParticipantRole": Specify if you want to match speech from the agent, the customer, or both,
  "Targets": [ The custom words and phrases you want to match ],
  "TranscriptFilterType": Use this parameter to specify an exact match for the specified targets
}
```

請參閱 [CreateCallAnalyticsCategory](#) 及 [TranscriptFilter](#)，了解這些參數與這些參數相關的有效值的詳細資訊。

### 非通話時間配對

使用非通話時間 ([NonTalkTimeFilter](#) 資料類型) 的規則設計為符合下列項目：

- 在整個通話的指定時間內出現靜音
- 在整個通話的指定時間內說話

以下是可搭配 [NonTalkTimeFilter](#) 使用的參數範例：

```
"NonTalkTimeFilter": {
  "AbsoluteTimeRange": {
    Specify the time frame, in milliseconds, when the match should occur
  },
  "RelativeTimeRange": {
    Specify the time frame, in percentage, when the match should occur
  }
}
```

```

},
  "Negate": Specify if you want to match the presence or absence of speech,
  "Threshold": Specify a threshold for the amount of time, in seconds, silence (or speech) occurred during the call
},

```

請參閱 [CreateCallAnalyticsCategory](#) 及 [NonTalkTimeFilter](#)，了解這些參數與這些參數相關的有效值的詳細資訊。

## 情緒配對

使用情緒 ([SentimentFilter](#) 資料類型) 的規則設計為符合下列項目：

- 客戶、客服人員或兩者是否在通話的指定點表達出正面情緒
- 客戶、客服人員或兩者是否在通話的指定點表達出負面情緒
- 客戶、客服人員或兩者是否在通話的指定點表達出中性情緒
- 客戶、客服人員或兩者是否在通話的指定點表達出混合情緒

以下是可搭配 [SentimentFilter](#) 使用的參數範例：

```

"SentimentFilter": {
  "AbsoluteTimeRange": {
    Specify the time frame, in milliseconds, when the match should occur
  },
  "RelativeTimeRange": {
    Specify the time frame, in percentage, when the match should occur
  },
  "Negate": Specify if you want to match the presence or absence of your chosen sentiment,
  "ParticipantRole": Specify if you want to match speech from the agent, the customer, or both,
  "Sentiments": [ The sentiments you want to match ]
},

```

請參閱 [CreateCallAnalyticsCategory](#) 及 [SentimentFilter](#)，了解這些參數與這些參數相關的有效值的詳細資訊。

## 開始通話後分析轉錄

在開始通話後分析轉錄之前，您必須在音頻中創建 Amazon Transcribe 要匹配的所有 [類別](#)。

**Note**

通話分析轉錄無法追溯配對至新類別。只有您開始通話分析轉錄前建立的類別，才能套用至該轉錄輸出。

如果您已建立一或多個類別，且音訊符合至少一個類別中的所有規則，Amazon Transcribe 將標記輸出為符合的類別。如果您選擇不使用類別，或您的音訊與類別中指定的規則不符，系統就不會標示您的文字記錄。

若要開始通話後分析轉錄，您可以使用 AWS Management Console、AWS CLI 或 AWS SDK；請參閱下列範例：

### AWS Management Console

使用以下程序以開始啟動通話後分析工作。會以該類別標示符合某個類別定義之所有特徵的通話。

1. 在功能窗格的 [Amazon Transcribe 通話分析] 下方，選擇 [通話分析工作]。
2. 選擇建立作業。

## Configure job - *optional* [Info](#)

### Content removal

Content removal conceals information in the resulting transcript from your source audio file. Amazon Transcribe changes items in the transcript and does not modify the source audio.

**PII redaction** [Info](#)

Label the type of PII and also mask the content with the PII entity type in the transcription output. For example, (123) 456-7890 will be masked as [PHONE].

**Vocabulary filtering** [Info](#)

Vocabulary filtering can remove, mask or tag specified words in the final transcript.

### Customization

**Custom vocabulary** [Info](#)

A custom vocabulary improves the accuracy of recognizing words and phrases specific to your use case.

### Summarization

**Generative call summarization** [Info](#)

Generative call summarization provides a summary of the transcript, including important components of the conversation.


### Categories

Create categories to classify calls. For example, you can create a category for all cancellation requests. When you run an analytics job, Amazon Transcribe applies that category to all calls that request cancellation.

#### Call analytics categories (1) [Info](#)

< 1 > 

	Name	Type	Created	Modified
<input type="radio"/>	CatchNegativeSentiment	POST_CALL	February 17 2023, 10:43 (UTC-08:00)	February 17 2023, 10:43 (UTC-08:00)

If the above categories aren't relevant to your use case, you can create a new category. [Create a new category.](#) 



3. 在指定作業詳細資訊頁面上，提供有關通話分析工作的資訊，包括輸入資料的位置。

## Specify job details [Info](#)

### Job settings

**Name**

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

**Model type** [Info](#)

Choose the type of model to use for the transcription job.

**General model**

To use a model that is not specialized for a particular use case, choose this option. Configuration options vary between languages.

**Custom language model**

To use a model that you trained for your specific use case, choose this option. This model has fewer configuration options than the general model.

**Language settings**

You can transcribe your audio file in a language that you specify or have Amazon Transcribe identify and transcribe it in the predominant language.

**Specific language** [Info](#)

If you know the language spoken in your source audio, choose this option to get the most accurate results. The options available for additional processing vary between languages.

**Automatic language identification** [Info](#)

If you don't know the language spoken in your audio files, choose this option. You have access to fewer options for additional processing than if you choose **Specific language**.

**Language**

Choose the language of the input audio.

指定輸出資料的所需 Amazon S3 位置以及要使用的 IAM 角色。

### Output data

Output data location type info [Info](#)

Service-managed S3 bucket  
The output will be removed after 90 days when the job expires.

Customer specified S3 bucket  
The output will not be removed from bucket even after the job expires.

### Access permissions

IAM role [Info](#)

Use an existing IAM role

Create an IAM role  
By choosing **Create job** you are authorizing creation of this role.

Permissions to access  
Your role has access to these resources. The KMS key permission is used only if your input bucket is encrypted

Input S3 bucket and KMS decrypt permission to input bucket

Any S3 bucket and any KMS keys

Role name  
Roles are prefixed with "AmazonTranscribeServiceRoleFullAccess-". Your newly created role has full access to the S3 bucket and KMS key for your account.

The name can be up to 64 characters long

▼ **Role permissions details**

Your new role has these permissions to give Amazon Transcribe access to the resources that you've specified.

Service	Access level	Resource
S3	List, Read, Write	All resources
Key Management Service	GenerateDataKey, Decrypt	All resources

4. 選擇下一步。
5. 位於設定工作，開啟您要包含在通話分析工作中的任何選用功能。如果您先前已建立類別，這些類別會在類別面板顯示，並會自動套用至您的通話分析工作。

## Configure job - *optional* [Info](#)

### Content removal

Content removal conceals information in the resulting transcript from your source audio file. Amazon Transcribe changes items in the transcript and does not modify the source audio.

**PII redaction** [Info](#)

Label the type of PII and also mask the content with the PII entity type in the transcription output. For example, (123) 456-7890 will be masked as [PHONE].

**Vocabulary filtering** [Info](#)

Vocabulary filtering can remove, mask or tag specified words in the final transcript.

### Customization

**Custom vocabulary** [Info](#)

A custom vocabulary improves the accuracy of recognizing words and phrases specific to your use case.

### Summarization

**Generative call summarization** [Info](#)

Generative call summarization provides a summary of the transcript, including important components of the conversation.


### Categories

Create categories to classify calls. For example, you can create a category for all cancellation requests. When you run an analytics job, Amazon Transcribe applies that category to all calls that request cancellation.

#### Call analytics categories (1) [Info](#)

< 1 > 

	Name	Type	Created	Modified
<input type="radio"/>	CatchNegativeSentiment	POST_CALL	February 17 2023, 10:43 (UTC-08:00)	February 17 2023, 10:43 (UTC-08:00)

If the above categories aren't relevant to your use case, you can create a new category. [Create a new category.](#) 

## 6. 選擇建立作業。

### AWS CLI

此範例使用 [start-call-analytics-job](#) 指令和 `channel-definitions` 參數。如需詳細資訊，請參閱 [StartCallAnalyticsJob](#) 及 [ChannelDefinition](#)。

```
aws transcribe start-call-analytics-job \  
--region us-west-2 \  
--call-analytics-job-name my-first-call-analytics-job \  
--media MediaFileUri=s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac \  
--output-location s3://DOC-EXAMPLE-BUCKET/my-output-files/ \  
--data-access-role-arn arn:aws:iam::111122223333:role/ExampleRole \  
--channel-definitions ChannelId=0,ParticipantRole=AGENT \  
ChannelId=1,ParticipantRole=CUSTOMER
```

以下是使用 [start-call-analytics-job](#) 的另一個範例，以及啟用該工作的通話分析的請求主文。

```
aws transcribe start-call-analytics-job \  
--region us-west-2 \  
--cli-input-json file://filepath/my-call-analytics-job.json
```

檔案 `my-call-analytics-job.json` 包含以下請求主文。

```
{  
  "CallAnalyticsJobName": "my-first-call-analytics-job",  
  "DataAccessRoleArn": "arn:aws:iam::111122223333:role/ExampleRole",  
  "Media": {  
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"  
  },  
  "OutputLocation": "s3://DOC-EXAMPLE-BUCKET/my-output-files/",  
  "ChannelDefinitions": [  
    {  
      "ChannelId": 0,  
      "ParticipantRole": "AGENT"  
    },  
    {  
      "ChannelId": 1,  
      "ParticipantRole": "CUSTOMER"  
    }  
  ]  
}
```

```
}
```

## AWS SDK for Python (Boto3)

這個範例會使用啟 AWS SDK for Python (Boto3) 動呼叫分析工作，使用開[始呼叫分析工作](#)。如需詳細資訊，請參閱 [StartCallAnalyticsJob](#) 及 [ChannelDefinition](#)。

如需使用 AWS SDK 的其他範例，包括特定功能、案例和跨服務範例，請參閱本章。[使用 SDK 進行 Amazon Transcribe 的代碼示例 AWS](#)

```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
job_name = "my-first-call-analytics-job"
job_uri = "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
output_location = "s3://DOC-EXAMPLE-BUCKET/my-output-files/"
data_access_role = "arn:aws:iam::111122223333:role/ExampleRole"
transcribe.start_call_analytics_job(
    CallAnalyticsJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    DataAccessRoleArn = data_access_role,
    OutputLocation = output_location,
    ChannelDefinitions = [
        {
            'ChannelId': 0,
            'ParticipantRole': 'AGENT'
        },
        {
            'ChannelId': 1,
            'ParticipantRole': 'CUSTOMER'
        }
    ]
)

while True:
    status = transcribe.get_call_analytics_job(CallAnalyticsJobName = job_name)
    if status['CallAnalyticsJob']['CallAnalyticsJobStatus'] in ['COMPLETED', 'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
```

```
print(status)
```

## 通話後分析輸出

通話後分析記錄會以區段的 turn-by-turn 格式顯示。它們包括通話分類、通話特徵 (響度分數、中斷、未通話時間、通話速度)、通話摘要 (問題、結果和行動項目)、修訂和情緒。此外，在文字記錄末尾處會提供對話特性的摘要。

若要提高準確性並進一步根據您的使用案例自訂文字記錄，例如包括產業專用術語，請在您的通話分析要求中新增[自訂詞彙](#)或[自訂語言模型](#)。若要遮蔽、移除或標記轉錄結果中不想要的文字 (例如褻瀆性文字)，請新增[字彙篩選](#)功能。如果您不確定要傳遞至媒體檔案的語言代碼為何，可以啟用[批次語言識別](#)，以自動識別媒體檔案中的語言。

以下章節示範見解層級上的 JSON 輸出範例。如需編譯的輸出，請參閱[編譯的通話後分析輸出](#)。

### 通話分類

以下是轉錄輸出中類別配對的外觀。此範例顯示 40040 毫秒時間戳記到 42460 毫秒時間戳記的音訊與「positive-resolution」類別相符。在這種情況下，自訂「positive-resolution」類別要求在對話的最後幾秒內的正面情緒。

```
"Categories": {
  "MatchedDetails": {
    "positive-resolution": {
      "PointsOfInterest": [
        {
          "BeginOffsetMillis": 40040,
          "EndOffsetMillis": 42460
        }
      ]
    }
  },
  "MatchedCategories": [
    "positive-resolution"
  ]
},
```

### 通話特徵

以下是轉錄輸出中通話特徵的外觀。請注意，每個對話回合都會提供響度分數，而所有其他特性則會在文字記錄的末尾處提供。

```
"LoudnessScores": [  
  87.54,  
  88.74,  
  90.16,  
  86.36,  
  85.56,  
  85.52,  
  81.79,  
  87.74,  
  89.82  
],  
  
...  
  
"ConversationCharacteristics": {  
  "NonTalkTime": {  
    "Instances": [],  
    "TotalTimeMillis": 0  
  },  
  "Interruptions": {  
    "TotalCount": 2,  
    "TotalTimeMillis": 10700,  
    "InterruptionsByInterrupter": {  
      "AGENT": [  
        {  
          "BeginOffsetMillis": 26040,  
          "DurationMillis": 5510,  
          "EndOffsetMillis": 31550  
        }  
      ],  
      "CUSTOMER": [  
        {  
          "BeginOffsetMillis": 770,  
          "DurationMillis": 5190,  
          "EndOffsetMillis": 5960  
        }  
      ]  
    }  
  },  
  "TotalConversationDurationMillis": 42460,  
  
...  
}
```



```

    "TalkSpeed": {
      "DetailsByParticipant": {
        "AGENT": {
          "AverageWordsPerMinute": 150
        },
        "CUSTOMER": {
          "AverageWordsPerMinute": 167
        }
      }
    },
    "TalkTime": {
      "DetailsByParticipant": {
        "AGENT": {
          "TotalTimeMillis": 32750
        },
        "CUSTOMER": {
          "TotalTimeMillis": 18010
        }
      },
      "TotalTimeMillis": 50760
    }
  },
},

```

### 問題、行動項目和後續步驟

- 在以下範例中，問題被識別為從字元 7 開始，並於字元 51 結束，這是指文字的以下部分：「我想取消我的食譜訂閱」。

```

"Content": "Well, I would like to cancel my recipe subscription.",

"IssuesDetected": [
  {
    "CharacterOffsets": {
      "Begin": 7,
      "End": 51
    }
  }
],

```

- 在以下範例中，結果被識別為從字元 12 開始，並於字元 78 結束，這是指文字的以下部分：「我對您的帳戶進行所有變更，現在已套用此折扣」。

```

"Content": "Wonderful. I made all changes to your account and now this discount is applied, please check.",
"OutcomesDetected": [
  {
    "CharacterOffsets": {
      "Begin": 12,
      "End": 78
    }
  }
],

```

- 在以下範例中，行動項目被標識為從字元 0 開始，並於字元 103 結束，這是指文字的以下部分：  
「我今天將傳送一封包含所有詳細資訊的電子郵件給您，然後下週我會回電以繼續追蹤」。

```

"Content": "I will send an email with all the details to you today, and I will call you back next week to follow up. Have a wonderful evening.",
"ActionItemsDetected": [
  {
    "CharacterOffsets": {
      "Begin": 0,
      "End": 103
    }
  }
],

```

## 生成式通話摘要

以下是轉錄輸出中生成式通話摘要的外觀：

```

"ContactSummary": {
  "AutoGenerated": {
    "OverallSummary": {
      "Content": "A customer wanted to check to see if we had a bag allowance. We told them that we didn't have it, but we could add the bag from Canada to Calgary and then do the one coming back as well."
    }
  }
}

```

在下列情況下，分析工作將在不產生摘要的情況下完成：

- 對話內容不足：對話必須至少包含來自代理和客戶的一個回合。當交談內容不足時，服務會傳回錯誤代碼不足。
- 安全護欄：對話必須滿足到位的安全護欄，以確保生成適當的摘要。如果不符合這些護欄，服務會傳回錯誤代碼 FAILED\_SAFETY\_ 指導方針。

錯誤代碼可以在輸出的Skipped部分AnalyticsJobDetails中找到。您也可以在此[GetCallAnalyticsJob](#) API 回應CallAnalyticsJobDetails中找到錯誤原因。

### 錯誤輸出範例

```
{
  "JobStatus": "COMPLETED",
  "AnalyticsJobDetails": {
    "Skipped": [
      {
        "Feature": "GENERATIVE_SUMMARIZATION",
        "ReasonCode": "INSUFFICIENT_CONVERSATION_CONTENT",
        "Message": "The conversation needs to have at least one turn from both
the participants to generate summary"
      }
    ]
  },
  "LanguageCode": "en-US",
  "AccountId": "*****",
  "JobName": "Test2-copy",
  ...
}
```

### 情緒分析

以下是轉錄輸出中情緒分析的外觀。

- 定性 turn-by-turn 情緒值：

```
"Content": "That's very sad to hear. Can I offer you a 50% discount to have you stay
with us?",
...
```

```

"BeginOffsetMillis": 12180,
"EndOffsetMillis": 16960,
"Sentiment": "NEGATIVE",
"ParticipantRole": "AGENT"

...

"Content": "That is a very generous offer. And I accept.",

...

"BeginOffsetMillis": 17140,
"EndOffsetMillis": 19860,
"Sentiment": "POSITIVE",
"ParticipantRole": "CUSTOMER"

```

- 整個通話的量化情緒值：

```

"Sentiment": {
  "OverallSentiment": {
    "AGENT": 2.5,
    "CUSTOMER": 2.1
  },

```

- 每個參與者和每四分之一的通話的量化情緒值：

```

"SentimentByPeriod": {
  "QUARTER": {
    "AGENT": [
      {
        "Score": 0.0,
        "BeginOffsetMillis": 0,
        "EndOffsetMillis": 9862
      },
      {
        "Score": -5.0,
        "BeginOffsetMillis": 9862,
        "EndOffsetMillis": 19725
      },
      {
        "Score": 5.0,
        "BeginOffsetMillis": 19725,
        "EndOffsetMillis": 29587
      }
    ]
  }
}

```

```

    },
    {
      "Score": 5.0,
      "BeginOffsetMillis": 29587,
      "EndOffsetMillis": 39450
    }
  ],
  "CUSTOMER": [
    {
      "Score": -2.5,
      "BeginOffsetMillis": 0,
      "EndOffsetMillis": 10615
    },
    {
      "Score": 5.0,
      "BeginOffsetMillis": 10615,
      "EndOffsetMillis": 21230
    },
    {
      "Score": 2.5,
      "BeginOffsetMillis": 21230,
      "EndOffsetMillis": 31845
    },
    {
      "Score": 5.0,
      "BeginOffsetMillis": 31845,
      "EndOffsetMillis": 42460
    }
  ]
}

```

## PII 修訂

以下是轉錄輸出中 PII 修訂的外觀。

```

"Content": "[PII], my name is [PII], how can I help?",
"Redaction": [{
  "Confidence": "0.9998",
  "Type": "NAME",
  "Category": "PII"
}]

```

如需詳細資訊，請參閱[修訂批次作業中的 PII](#)。

## 語言識別

若啟用此功能，則轉錄輸出中語言識別的外觀如下。

```
"LanguageIdentification": [{
  "Code": "en-US",
  "Score": "0.8299"
}, {
  "Code": "en-NZ",
  "Score": "0.0728"
}, {
  "Code": "zh-TW",
  "Score": "0.0695"
}, {
  "Code": "th-TH",
  "Score": "0.0156"
}, {
  "Code": "en-ZA",
  "Score": "0.0121"
}]
```

在上面的輸出範例中，語言識別將填入語言代碼和可信度分數。分數最高的結果將選為轉錄的語言代碼。如需模式詳細資訊，請參閱[識別媒體中的優勢語言](#)。

## 編譯的通話後分析輸出

為了簡潔起見，以下轉錄輸出的某些內容將以省略符號取代。

此範例包含選用功能-生成呼叫摘要。

```
{
  "JobStatus": "COMPLETED",
  "LanguageCode": "en-US",
  "Transcript": [
    {
      "LoudnessScores": [
        78.63,
        78.37,
        77.98,
        74.18
      ],

```

```
"Content": "[PII], my name is [PII], how can I help?",
...

"Content": "Well, I would like to cancel my recipe subscription.",
"IssuesDetected": [
  {
    "CharacterOffsets": {
      "Begin": 7,
      "End": 51
    }
  }
],
...

"Content": "That's very sad to hear. Can I offer you a 50% discount to have
you stay with us?",
"Items": [
  ...
],
"Id": "649afe93-1e59-4ae9-a3ba-a0a613868f5d",
"BeginOffsetMillis": 12180,
"EndOffsetMillis": 16960,
"Sentiment": "NEGATIVE",
"ParticipantRole": "AGENT"
},
{
  "LoudnessScores": [
    80.22,
    79.48,
    82.81
  ],
  "Content": "That is a very generous offer. And I accept.",
  "Items": [
    ...
  ],
  "Id": "f9266cba-34df-4ca8-9cea-4f62a52a7981",
  "BeginOffsetMillis": 17140,
  "EndOffsetMillis": 19860,
  "Sentiment": "POSITIVE",
  "ParticipantRole": "CUSTOMER"
},
{
```

```
...

    "Content": "Wonderful. I made all changes to your account and now this
discount is applied, please check.",
    "OutcomesDetected": [
        {
            "CharacterOffsets": {
                "Begin": 12,
                "End": 78
            }
        }
    ],
    ...

    "Content": "I will send an email with all the details to you today, and I
will call you back next week to follow up. Have a wonderful evening.",
    "Items": [
        ...
    ],
    "Id": "78cd0923-cafd-44a5-a66e-09515796572f",
    "BeginOffsetMillis": 31800,
    "EndOffsetMillis": 39450,
    "Sentiment": "POSITIVE",
    "ParticipantRole": "AGENT"
},
{
    "LoudnessScores": [
        78.54,
        68.76,
        67.76
    ],
    "Content": "Thank you very much, sir. Goodbye.",
    "Items": [
        ...
    ],
    "Id": "5c5e6be0-8349-4767-8447-986f995af7c3",
    "BeginOffsetMillis": 40040,
    "EndOffsetMillis": 42460,
    "Sentiment": "POSITIVE",
    "ParticipantRole": "CUSTOMER"
}
],
```



```
...

"Categories": {
  "MatchedDetails": {
    "positive-resolution": {
      "PointsOfInterest": [
        {
          "BeginOffsetMillis": 40040,
          "EndOffsetMillis": 42460
        }
      ]
    }
  },
  "MatchedCategories": [
    "positive-resolution"
  ]
},

...

"ConversationCharacteristics": {
  "NonTalkTime": {
    "Instances": [],
    "TotalTimeMillis": 0
  },
  "Interruptions": {
    "TotalCount": 2,
    "TotalTimeMillis": 10700,
    "InterruptionsByInterrupter": {
      "AGENT": [
        {
          "BeginOffsetMillis": 26040,
          "DurationMillis": 5510,
          "EndOffsetMillis": 31550
        }
      ],
      "CUSTOMER": [
        {
          "BeginOffsetMillis": 770,
          "DurationMillis": 5190,
          "EndOffsetMillis": 5960
        }
      ]
    }
  }
}
```

```
    }
  },
  "TotalConversationDurationMillis": 42460,
  "Sentiment": {
    "OverallSentiment": {
      "AGENT": 2.5,
      "CUSTOMER": 2.1
    },
    "SentimentByPeriod": {
      "QUARTER": [
        "AGENT": [
          {
            "Score": 0.0,
            "BeginOffsetMillis": 0,
            "EndOffsetMillis": 9862
          },
          {
            "Score": -5.0,
            "BeginOffsetMillis": 9862,
            "EndOffsetMillis": 19725
          },
          {
            "Score": 5.0,
            "BeginOffsetMillis": 19725,
            "EndOffsetMillis": 29587
          },
          {
            "Score": 5.0,
            "BeginOffsetMillis": 29587,
            "EndOffsetMillis": 39450
          }
        ],
        "CUSTOMER": [
          {
            "Score": -2.5,
            "BeginOffsetMillis": 0,
            "EndOffsetMillis": 10615
          },
          {
            "Score": 5.0,
            "BeginOffsetMillis": 10615,
            "EndOffsetMillis": 21230
          }
        ]
      ]
    }
  }
}
```

```

        "Score": 2.5,
        "BeginOffsetMillis": 21230,
        "EndOffsetMillis": 31845
    },
    {
        "Score": 5.0,
        "BeginOffsetMillis": 31845,
        "EndOffsetMillis": 42460
    }
]
}
},
"TalkSpeed": {
    "DetailsByParticipant": {
        "AGENT": {
            "AverageWordsPerMinute": 150
        },
        "CUSTOMER": {
            "AverageWordsPerMinute": 167
        }
    }
},
"TalkTime": {
    "DetailsByParticipant": {
        "AGENT": {
            "TotalTimeMillis": 32750
        },
        "CUSTOMER": {
            "TotalTimeMillis": 18010
        }
    },
    "TotalTimeMillis": 50760
},
"ContactSummary": { // Optional feature - Generative call summarization
    "AutoGenerated": {
        "OverallSummary": {
            "Content": "The customer initially wanted to cancel but the agent
convinced them to stay by offering a 50% discount, which the customer accepted after
reconsidering cancelling given the significant savings. The agent ensured the discount
was applied and said they would follow up to ensure the customer remained happy with
the revised subscription."
        }
    }
}
}

```

```
    }  
  },  
  "AnalyticsJobDetails": {  
    "Skipped": []  
  },  
  ...  
}
```

## 啟用生成式通話摘要

### Note

由 Amazon 基岩提供支援：AWS 實作[自動濫用偵測](#)。因為採用生成式 AI 技術的聯絡後摘要是在 Amazon Bedrock 的基礎上，所以使用者可以充分利用 Amazon Bedrock 中實作的控制措施，強制執行人工智慧 (AI) 的安全、保障和負責任使用目標。

若要使用生成式通話摘要搭配通話後分析作業，請參閱下列範例：

### AWS Management Console

在「摘要」面板中啟用「生成式通話摘要」，以在輸出中接收摘要。

## Configure job - *optional* [Info](#)

### Content removal

Content removal conceals information in the resulting transcript from your source audio file. Amazon Transcribe changes items in the transcript and does not modify the source audio.

- PII redaction** [Info](#)  
Label the type of PII and also mask the content with the PII entity type in the transcription output. For example, (123) 456-7890 will be masked as [PHONE].
- Vocabulary filtering** [Info](#)  
Vocabulary filtering can remove, mask or tag specified words in the final transcript.

### Customization

- Custom vocabulary** [Info](#)  
A custom vocabulary improves the accuracy of recognizing words and phrases specific to your use case.

### Summarization

- Generative call summarization** [Info](#)  
Generative call summarization provides a summary of the transcript, including important components of the conversation.


### Categories

Create categories to classify calls. For example, you can create a category for all cancellation requests. When you run an analytics job, Amazon Transcribe applies that category to all calls that request cancellation.

#### Call analytics categories (1) [Info](#)

< 1 > 

	Name	Type	Created	Modified
<input type="radio"/>	CatchNegativeSentiment	POST_CALL	February 17 2023, 10:43 (UTC-08:00)	February 17 2023, 10:43 (UTC-08:00)

If the above categories aren't relevant to your use case, you can create a new category. [Create a new category.](#) 

## AWS CLI

此範例使用 [start-call-analytics-job](#) 命令和 Settings 參數搭配 Summarization 子參數。如需詳細資訊，請參閱 [StartCallAnalyticsJob](#)。

```
aws transcribe start-call-analytics-job \  
--region us-west-2 \  
--call-analytics-job-name my-first-call-analytics-job \  
--media MediaFileUri=s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac \  
--output-location s3://DOC-EXAMPLE-BUCKET/my-output-files/ \  
--data-access-role-arn arn:aws:iam::111122223333:role/ExampleRole \  
--channel-definitions ChannelId=0,ParticipantRole=AGENT  
ChannelId=1,ParticipantRole=CUSTOMER  
--settings '{"Summarization":{"GenerateAbstractiveSummary":true}}'
```

以下是使用 [start-call-analytics-job](#) 命令的另一個範例，以及啟用該作業摘要的請求本文。

```
aws transcribe start-call-analytics-job \  
--region us-west-2 \  
--cli-input-json file://filepath/my-call-analytics-job.json
```

檔案 `my-call-analytics-job.json` 包含以下請求主文。

```
{  
  "CallAnalyticsJobName": "my-first-call-analytics-job",  
  "DataAccessRoleArn": "arn:aws:iam::111122223333:role/ExampleRole",  
  "Media": {  
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"  
  },  
  "OutputLocation": "s3://DOC-EXAMPLE-BUCKET/my-output-files/",  
  "ChannelDefinitions": [  
    {  
      "ChannelId": 0,  
      "ParticipantRole": "AGENT"  
    },  
    {  
      "ChannelId": 1,  
      "ParticipantRole": "CUSTOMER"  
    }  
  ]  
}
```

```
    }
  ],
  "Settings": {
    "Summarization":{
      "GenerateAbstractiveSummary": true
    }
  }
}
```

## AWS SDK for Python (Boto3)

此範例會使用啟動呼叫分析，並 AWS SDK for Python (Boto3) 使用 [start\\_call\\_Analytics\\_job](#) 方法啟動摘要功能。如需詳細資訊，請參閱 [StartCallAnalyticsJob](#)。

如需使用 AWS SDK 的其他範例，包括特定功能、案例和跨服務範例，請參閱本章。[使用 SDK 進行 Amazon Transcribe 的代碼示例 AWS](#)

```
from __future__ import print_function
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
job_name = "my-first-call-analytics-job"
job_uri = "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
output_location = "s3://DOC-EXAMPLE-BUCKET/my-output-files/"
data_access_role = "arn:aws:iam::111122223333:role/ExampleRole"
transcribe.start_call_analytics_job(
    CallAnalyticsJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    DataAccessRoleArn = data_access_role,
    OutputLocation = output_location,
    ChannelDefinitions = [
        {
            'ChannelId': 0,
            'ParticipantRole': 'AGENT'
        },
        {
            'ChannelId': 1,
            'ParticipantRole': 'CUSTOMER'
        }
    ]
}
```

```
],
Settings = {
  "Summarization":
    {
      "GenerateAbstractiveSummary": true
    }
}
)

while True:
  status = transcribe.get_call_analytics_job(CallAnalyticsJobName = job_name)
  if status['CallAnalyticsJob']['CallAnalyticsJobStatus'] in ['COMPLETED', 'FAILED']:
    break
  print("Not ready yet...")
  time.sleep(5)
print(status)
```

## 即時通話分析

即時通話分析提供即時見解，可用於解決問題並緩解發生的升級情況。

即時通話分析提供以下見解：

- 使用規則標記特定關鍵字和片語的[類別事件](#)；類別事件可用於建立[即時警示](#)
- [問題偵測](#)識別每個音訊區段中所說的問題
- 文字記錄中的 [PII \(敏感資料\) 識別](#)
- 文字記錄中的 [PII \(敏感資料\) 修訂](#)
- 每個語音區段的[情緒分析](#)

除了即時通話分析外，還 Amazon Transcribe 可以在媒體串流上執行[通話後分析](#)。您可以使用 [PostCallAnalyticsSettings](#) 參數，納入通話後分析到即時通話分析請求。

## 即時見解

本節詳細介紹即時通話分析轉錄的可用見解。

### 類別事件

使用類別事件，您可以根據精確的關鍵字或片語，配對轉錄。例如，如果您為「我要與管理員說話」短語設定篩選 Amazon Transcribe 器，則會篩選該確切片語。



這是[輸出範例](#)。

如需建立即時通話分析類別的詳細資訊，請參閱 [建立即時轉錄的類別](#)。

**i** Tip

類別事件可讓您設定即時警示；請參閱 [建立類別配對的即時警示](#)，了解詳細資訊。

## 問題偵測

問題偵測提供每個音訊區段中偵測到問題的簡短摘要。使用問題偵測功能，您可以：

- 減少通話期間和通話後手動記錄的需求
- 提高客服人員效率，讓他們可以更快回應客戶

**i** Note

以下英文方言支援問題偵測：澳洲英文 (en-AU)、英國英文 (en-GB) 和美國英文 (en-US)。

問題偵測適用於所有產業和商業，且須視情境而定。它的工作原理 out-of-the-box，因此不支持自定義，例如模型培訓或自定義類別。

每個完整的音訊區段會使用即時通話分析執行問題偵測。

這是[輸出範例](#)。

## PII(敏感資料) 識別

敏感資料識別會在文字記錄中標記個人身分識別資訊 (PII)。此參數有助於保護客戶資訊。

**i** Note

以下英語方言支援即時 PII 識別：澳洲 (en-AU)、英國 (en-GB)、美國 (en-US) 和西班牙語方言 (es-US)。

每個完整的音訊區段會使用即時通話分析執行 PII 識別。



您可以根據需要，建立任意數量的類別，以涵蓋一系列不同的情況。您必須為您建立的每個類別建立 1 到 20 個規則。即時通話分析轉錄僅支援使用 [TranscriptFilter](#) (關鍵字配對) 的規則。如需規則與 [CreateCallAnalyticsCategory](#) 操作搭配使用的詳細資訊，請參閱 [即時通話分析類別的規則條件](#) 一節。

如果媒體中的內容符合您在指定類別中指定的所有規則，Amazon Transcribe 將標示輸出為該類別。請參閱 [類別事件輸出](#)，了解 JSON 輸出格式中相符類別的範例。

以下是您可以使用自訂類別執行的幾個範例：

- 標記和追蹤特定的關鍵字集，識別需要立即關注的問題
- 監控合規性，例如客服人員說出 (或省略) 特定片語
- 即時標記特定單字和片語；然後您可以設定類別相符項目以設定立即警示。例如，如果您為客戶建立「與經理對話」的即時通話分析類別，您可以為此即時類別配對，設定 [事件警示](#)，以通知值班經理。

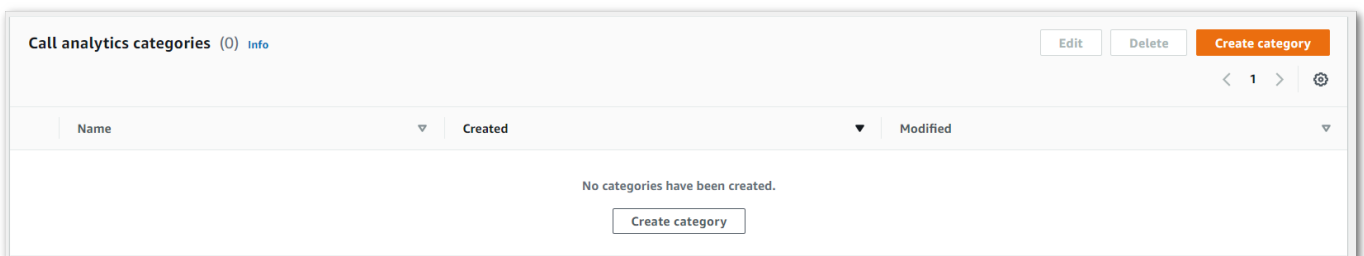
## 通話後與即時類別

建立新類別時，您可以指定建立為通話後類別 (POST\_CALL) 或建立為即時類別 (REAL\_TIME)。如果未指定選項，則預設會將類別建立為通話後類別。即時類別相符項目可用於建立即時警示。如需詳細資訊，請參閱 [建立類別配對的即時警示](#)。

若要建立即時通話分析的新類別，您可以使用 AWS Management Console、AWS CLI 或 AWS SDK；請參閱下列範例：

### AWS Management Console

1. 在功能窗格的下 Amazon Transcribe，選擇 [Amazon Transcribe 通話分析]。
2. 選擇通話分析類別，將引導您進入通話分析類別頁面。選擇建立類別按鈕。



3. 您現在已進入建立類別頁面。輸入類別的名稱，然後在類別類型下拉式功能表中，選擇「即時通話分析」。

### Category settings

**Category name**

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

**Category type [Info](#)**

**Batch call analytics**

**Real time call analytics**

Use a template (recommended)  
 Use a template to edit predefined rules.

Create from scratch  
 If you know the rules that you want to define, choose this option.

**Template type [Info](#)**

Choose the template for the category that most closely matches the one you want to create.

4. 您可以選擇範本，以建立您的類別，也可以從頭開始建立範本。

如果使用範本：選擇使用範本 (建議選項)，選擇您要的範本，然後選擇建立類別。

### Category settings

**Category name**

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

**Category type [Info](#)**

**Category creation method [Info](#)**

Use a template (recommended)  
 Use a template to edit predefined rules.

Create from scratch  
 If you know the rules that you want to define, choose this option.

**Template type [Info](#)**

Choose the template for the category that most closely matches the one you want to create.

Customer content is negative and mentioned manager

5. 如果建立自訂類別：選擇從頭開始建立。

## Create category [Info](#)

### Category settings

**Category name**

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

**Category creation method [Info](#)**

Use a template (recommended)  
Use a template to edit predefined rules.

Create from scratch  
If you know the rules that you want to define, choose this option.

### Rules

All the rule conditions must be met for a transcription job to be classified in this category.

▼ Rule 1 Delete rule

**Rule type [Info](#)**  
Choose the rule that you want to define.

**Add rule**

You can add up to 19 more rules.

6. 使用下拉式功能表將規則新增至您的類別。每個類別最多可新增 20 個規則。使用即時通話分析轉錄時，您只能納入涉及文字記錄內容配對的規則。任何相符項目都會即時標記。

### Rules

All the rule conditions must be met for a transcription job to be classified in this category.

▼ Rule 1 Delete rule

**Rule type [Info](#)**  
Choose the rule that you want to define.

**Transcript content match**  
Trigger the rule when the speaker says the words or phrases that you specify.

**Add rule**

You can add up to 19 more rules.

7. 這是帶一個規則的類別範例：通話的任何時間點說出「與經理談談」的客戶。

### Rules

All the rule conditions must be met for a transcription job to be classified in this category.

▼ **Rule 1** Delete rule

When any of the words were mentioned during the entire call when the speaker was customer.

**Rule type** [Info](#)  
Choose the rule that you want to define.

Transcript content match ▼

**Logic** [Info](#)  
Define the conditions that must be met.

When any of the words were mentioned ▼

during the entire call ▼

when the speaker was customer ▼

**Words or phrases** [Info](#)  
Enter the words or phrases that you want to look for in the transcript. You can enter up to 100 words or phrases.

The word or phrase can be up to 2,000 characters.

You can add up to 19 more rules.

8. 將規則新增至類別時，選擇建立類別。

## AWS CLI

此範例使用 [create-call-analytics-category](#) 指令。如需詳細資訊，請參閱 [CreateCallAnalyticsCategory](#)、[CategoryProperties](#) 及 [Rule](#)。

下列範例會建立帶規則的類別：

- 客戶在通話中的任何時間點說出「與經理談談」這句。

此範例使用 [create-call-analytics-category](#) 指令，以及新增規則至類別的請求主文。

```
aws transcribe create-call-analytics-category \
--cli-input-json file://filepath/my-first-analytics-category.json
```

檔案 my-first-analytics-category.json 包含以下請求主文。

```
{
  "CategoryName": "my-new-real-time-category",
  "InputType": "REAL_TIME",
  "Rules": [
    {
      "TranscriptFilter": {
        "Negate": false,
        "Targets": [
          "speak to the manager"
        ],
        "TranscriptFilterType": "EXACT"
      }
    }
  ]
}
```

### AWS SDK for Python (Boto3)

此範例會使 AWS SDK for Python (Boto3) 用建立類別，使用 `CategoryName` 和 `Rules` 引數來建立類別。如需詳細資訊，請參閱 [CreateCallAnalyticsCategory](#)、[CategoryProperties](#) 及 [Rule](#)。

如需使用 AWS SDK 的其他範例，包括特定功能、案例和跨服務範例，請參閱本章。 [使用 SDK 進行 Amazon Transcribe 的代碼示例 AWS](#)

下列範例會建立帶規則的類別：

- 客戶在通話中的任何時間點說出「與經理談談」這句。

```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
category_name = "my-new-real-time-category"
transcribe.create_call_analytics_category(
    CategoryName = category_name,
    InputType = "REAL_TIME",
    Rules = [
        {
            'TranscriptFilter': {
                'Negate': False,
                'Targets': [
                    'speak to the manager'
                ]
            }
        }
    ]
)
```

```

        ],
        'TranscriptFilterType': 'EXACT'
    }
}
]
)

result = transcribe.get_call_analytics_category(CategoryName = category_name)
print(result)

```

## 即時通話分析類別的規則條件

本節概述您可以使用 [CreateCallAnalyticsCategory](#) API 操作建立的自訂 REAL\_TIME 規則類型。

問題偵測會自動執行，因此您不需要建立任何規則或類別以標記問題。

請注意，即時通話分析轉錄只支援關鍵字相符項目。如果您要建立包含中斷、靜音或情緒的類別，請參閱 [通話後分析類別的規則條件](#)。

### 關鍵字配對

使用關鍵字 ([TranscriptFilter](#) 資料類型) 的規則設計為符合下列項目：

- 客服人員、客戶或兩者說出的自訂單字或片語
- 客服人員、客戶或兩者都未說出的自訂單字或片語
- 在特定時間範圍內發生的自訂單字或片語

以下是可搭配 [TranscriptFilter](#) 使用的參數範例：

```

"TranscriptFilter": {
  "AbsoluteTimeRange": {
    Specify the time frame, in milliseconds, when the match should occur
  },
  "RelativeTimeRange": {
    Specify the time frame, in percentage, when the match should occur
  },
  "Negate": Specify if you want to match the presence or absence of your custom keywords,
  "ParticipantRole": Specify if you want to match speech from the agent, the customer, or both,
  "Targets": [ The custom words and phrases you want to match ],

```



```
"TranscriptFilterType": Use this parameter to specify an exact match for the specified targets
}
```

請參閱 [CreateCallAnalyticsCategory](#) 及 [TranscriptFilter](#)，了解這些參數與這些參數相關的有效值的詳細資訊。

## 具有即時轉錄功能的通話後分析

通話後分析是使用即時通話分析轉錄時的可選功能。除了標準的[即時分析見解](#)外，通話後分析還提供下列功能：

- 行動項目：列出通話中識別的任何行動項目
- 中斷：衡量是否以及何時一個參與者打斷另一個參與者的中間句。
- 問題：提供通話中識別的問題
- 響度：衡量每位參與者的發言音量
- 非通話時間：衡量沒有語音的時間段
- 結果：提供通話中識別的結果或解決方案
- 通話速度：衡量兩位參與者的說話速度
- 通話時間：衡量每位參與者通話期間發言的時間量 (以毫秒為單位)

啟用時，來自音訊串流的通話後分析會產生類似於音訊[檔案的通話後分析記錄](#)，並將其儲存在中指定的儲存 Amazon S3 貯體中。OutputLocation 此外，通話後分析會記錄您的音訊串流，並將其儲存為同一儲存 Amazon S3 貯體中的音訊檔案 (WAV 格式)。如果啟用密文，經過編輯的文字記錄和編輯過的音訊檔案也會儲存在指定的值區中。Amazon S3 使用音訊串流啟用通話後分析，會如下所述產生兩個至四個檔案：

- 如果未啟用修訂，您的輸出檔案為：
  1. 未修訂的文字記錄
  2. 未修訂的音訊檔案
- 如果在沒有未修訂選項 (redacted) 的情況下啟用修訂，您的輸出檔案為：
  1. 修訂後的文字記錄
  2. 修訂後的音訊檔案
- 如果使用未修訂選項 (redacted\_and\_unredacted) 啟用修訂，您的輸出檔案為：
  1. 修訂後的文字記錄

2. 修訂後的音訊檔案
3. 未修訂的文字記錄
4. 未修訂的音訊檔案

請注意，如果您的請求啟用通話後分析 ([PostCallAnalyticsSettings](#))，且正在使用 FLAC 或 OPUS-OGG 媒體，則不會顯示文字記錄的 loudnessScore，也不會建立串流的音訊錄音。轉錄可能也無法為持續超過 90 分鐘的長時間運行音頻流提供通話後分析。

如需使用音訊串流的通話後分析的見解的詳細資訊，請參閱[通話後分析見解](#)一節。

#### Tip

如果您的即時通話分析請求啟用通話後分析，您的所有 POST\_CALL 和 REAL-TIME 類別都會套用至您的通話後分析轉錄。

## 啟用通話後分析

若要啟用通話後分析，您必須在即時通話分析請求中包含 [PostCallAnalyticsSettings](#) 參數。啟用 PostCallAnalyticsSettings 時，必須包含下列參數：

- OutputLocation：您要 Amazon S3 儲存通話後成績單的值區。
- DataAccessRoleArn：具有指定 Amazon S3 儲存貯體存取權限之 Amazon S3 角色的 Amazon Resource Name (ARN)。請注意，您還必須使用[即時分析的信任政策](#)。

如果您想要修訂版的文字記錄，可以在請求中包含 ContentRedactionOutput 或 ContentRedactionType。如需這些參數的詳細資訊，請參閱《API 參考資料》中的 [StartCallAnalyticsStreamTranscription](#)。

若要在啟用通話後分析的情況下啟動即時通話分析轉錄，您可以使用 AWS Management Console(僅限示範)、HTTP/2 或 WebSockets 如需範例，請參閱 [開始即時通話分析轉錄](#)。

#### Important

目前，AWS Management Console 僅提供帶有預加載音頻示例的實時呼叫分析的演示。如果您想要使用自己的音訊，您必須使用 API (HTTP/2 或 SDK)。WebSockets

## 通話後分析輸出範例

通話後記錄會依區段顯示 turn-by-turn 格式。它們包括通話特徵、情緒、通話摘要、問題偵測和 (選擇性) PII 修訂。如果通話後類別與音訊內容相符，這些類別也會在您的輸出出現。

若要提高準確性並進一步根據您的使用案例自訂文字記錄，例如包括產業專用術語，請在您的通話分析要求中新增 [自訂詞彙](#) 或 [自訂語言模型](#)。若要遮蔽、移除或標記轉錄結果中不想要的字詞 (例如褻瀆性文字)，請新增 [字彙篩選](#) 功能。

以下是編譯通話後分析輸出範例：

```
{
  "JobStatus": "COMPLETED",
  "LanguageCode": "en-US",
  "AccountId": "1234567890",
  "Channel": "VOICE",
  "Participants": [{
    "ParticipantRole": "AGENT"
  }],
  {
    "ParticipantRole": "CUSTOMER"
  }],
  "SessionId": "12a3b45c-de6f-78g9-0123-45h6ab78c901",
  "ContentMetadata": {
    "Output": "Raw"
  }
  "Transcript": [{
    "LoudnessScores": [
      78.63,
      78.37,
      77.98,
      74.18
    ],
    "Content": "[PII], my name is [PII], how can I help?",
    ...
    "Content": "Well, I would like to cancel my recipe subscription.",
    "IssuesDetected": [{
      "CharacterOffsets": {
        "Begin": 7,
        "End": 51
      }
    }
  ]
}
```

```

    ]],
    ...

    "Content": "That's very sad to hear. Can I offer you a 50% discount to have you
stay with us?",
    "Id": "649afe93-1e59-4ae9-a3ba-a0a613868f5d",
    "BeginOffsetMillis": 12180,
    "EndOffsetMillis": 16960,
    "Sentiment": "NEGATIVE",
    "ParticipantRole": "AGENT"
  },
  {
    "LoudnessScores": [
      80.22,
      79.48,
      82.81
    ],
    "Content": "That is a very generous offer. And I accept.",
    "Id": "f9266cba-34df-4ca8-9cea-4f62a52a7981",
    "BeginOffsetMillis": 17140,
    "EndOffsetMillis": 19860,
    "Sentiment": "POSITIVE",
    "ParticipantRole": "CUSTOMER"
  },
  ...

  "Content": "Wonderful. I made all changes to your account and now this discount
is applied, please check.",
  "OutcomesDetected": [{
    "CharacterOffsets": {
      "Begin": 12,
      "End": 78
    }
  }
  ]],
  ...

  "Content": "I will send an email with all the details to you today, and I will
call you back next week to follow up. Have a wonderful evening.",
  "Id": "78cd0923-cafd-44a5-a66e-09515796572f",
  "BeginOffsetMillis": 31800,
  "EndOffsetMillis": 39450,
  "Sentiment": "POSITIVE",

```

```
    "ParticipantRole": "AGENT"
  },
  {
    "LoudnessScores": [
      78.54,
      68.76,
      67.76
    ],
    "Content": "Thank you very much, sir. Goodbye.",
    "Id": "5c5e6be0-8349-4767-8447-986f995af7c3",
    "BeginOffsetMillis": 40040,
    "EndOffsetMillis": 42460,
    "Sentiment": "POSITIVE",
    "ParticipantRole": "CUSTOMER"
  }
],
...

"Categories": {
  "MatchedDetails": {
    "positive-resolution": {
      "PointsOfInterest": [{
        "BeginOffsetMillis": 40040,
        "EndOffsetMillis": 42460
      }]
    }
  },
  "MatchedCategories": [
    "positive-resolution"
  ]
},
...

"ConversationCharacteristics": {
  "NonTalkTime": {
    "Instances": [],
    "TotalTimeMillis": 0
  },
  "Interruptions": {
    "TotalCount": 2,
    "TotalTimeMillis": 10700,
    "InterruptionsByInterrupter": {
```

```
    "AGENT": [{
      "BeginOffsetMillis": 26040,
      "DurationMillis": 5510,
      "EndOffsetMillis": 31550
    }],
    "CUSTOMER": [{
      "BeginOffsetMillis": 770,
      "DurationMillis": 5190,
      "EndOffsetMillis": 5960
    }]
  }
},
"TotalConversationDurationMillis": 42460,
"Sentiment": {
  "OverallSentiment": {
    "AGENT": 2.5,
    "CUSTOMER": 2.1
  },
  "SentimentByPeriod": {
    "QUARTER": {
      "AGENT": [{
        "Score": 0.0,
        "BeginOffsetMillis": 0,
        "EndOffsetMillis": 9862
      }],
      {
        "Score": -5.0,
        "BeginOffsetMillis": 9862,
        "EndOffsetMillis": 19725
      },
      {
        "Score": 5.0,
        "BeginOffsetMillis": 19725,
        "EndOffsetMillis": 29587
      },
      {
        "Score": 5.0,
        "BeginOffsetMillis": 29587,
        "EndOffsetMillis": 39450
      }
    ],
    "CUSTOMER": [{
      "Score": -2.5,
      "BeginOffsetMillis": 0,
```

```
        "EndOffsetMillis": 10615
      },
      {
        "Score": 5.0,
        "BeginOffsetMillis": 10615,
        "EndOffsetMillis": 21230
      },
      {
        "Score": 2.5,
        "BeginOffsetMillis": 21230,
        "EndOffsetMillis": 31845
      },
      {
        "Score": 5.0,
        "BeginOffsetMillis": 31845,
        "EndOffsetMillis": 42460
      }
    ]
  }
},
"TalkSpeed": {
  "DetailsByParticipant": {
    "AGENT": {
      "AverageWordsPerMinute": 150
    },
    "CUSTOMER": {
      "AverageWordsPerMinute": 167
    }
  }
},
"TalkTime": {
  "DetailsByParticipant": {
    "AGENT": {
      "TotalTimeMillis": 32750
    },
    "CUSTOMER": {
      "TotalTimeMillis": 18010
    }
  },
  "TotalTimeMillis": 50760
}
},
...
```

}

## 開始即時通話分析轉錄

在開始進行即時通話分析轉錄之前，您必須建立 Amazon Transcribe 要在通話中符合的所有[類別](#)。

### Note

通話分析轉錄無法追溯配對至新類別。只有您開始通話分析轉錄前建立的類別，才能套用至該轉錄輸出。

如果您已建立一或多個類別，且音訊符合至少一個類別中的所有規則，Amazon Transcribe 將標記輸出為符合的類別。如果您選擇不使用類別，或您的音訊與類別中指定的規則不符，系統就不會標示您的文字記錄。

若要將通話後分析納入即時通話分析轉錄中，您必須使用 `OutputLocation` 參數，在請求中提供 Amazon S3 儲存貯體。您還必須包含有指定儲存貯體寫入權限的 `DataAccessRoleArn`。您的即時通話分析串流工作階段完成後，系統會在指定的儲存貯體中產生並儲存獨立的文字記錄。

使用即時通話分析，您還可以選擇建立即時類別警示；請參閱 [建立類別配對的即時警示](#)，了解指示。

若要啟動即時通話分析轉錄，您可以使用 HTTP/2 或 WebSockets；如需範例，請參閱以下內容：AWS Management Console

### Important

目前，AWS Management Console 僅提供帶有預加載音頻示例的實時呼叫分析的演示。如果您想要使用自己的音訊，您必須使用 API (HTTP/2 或 SDK)。WebSockets

## AWS Management Console

使用以下程序以開始通話分析請求。會以該類別標示符合某個類別定義之所有特徵的通話。

### Note

僅在 AWS Management Console 可使用示範。若要啟動自訂即時分析轉錄，您必須使用 [API](#)。



1. 在功能窗格的 [Amazon Transcribe 通話分析] 下方，選擇 [分析即時通話]。

Amazon Transcribe > Real-time Analytics

## Real-time Analytics [info](#)

Transcribe Real-time Call Analytics combines powerful speech-to-text and natural language processing (NLP) models that are trained specifically to understand customer service and sales calls. With Transcribe Call Analytics, developers can get a redacted and unredacted transcript, and insights such as customer and agent sentiment, detected issues, and supervisor alerts during the live call.

### How it works

This demo experience has been configured to use preloaded audio examples of customer-agent interactions. Before starting the demo, you can optionally create categories in the Category Management page and update content redaction settings under the advance settings

**Step 1: Specify input audio**

Input audio file

Insurance complaints (en-US)

00:00/00:00

**Step 2: Review call categories - optional**

Categorize your calls based on custom keywords or phrases.

View categories

**Step 3: Configure output - optional**

Apply content redaction settings to your calls.

Configure advanced settings

### Post-call Analytics

Post-call analytics enabled with real-time analytics provides consolidated transcript and audio backup, with the associated analytics, along with further insights such as call summaries and conversation characteristics like non-talk time, interruptions, loudness, and talk speed, after the end of the call in the provided Amazon S3 bucket.

Post-call Analytics

Start streaming

2. 對於步驟 1：指定輸入音訊，從下拉式功能表選擇示範測試檔案。



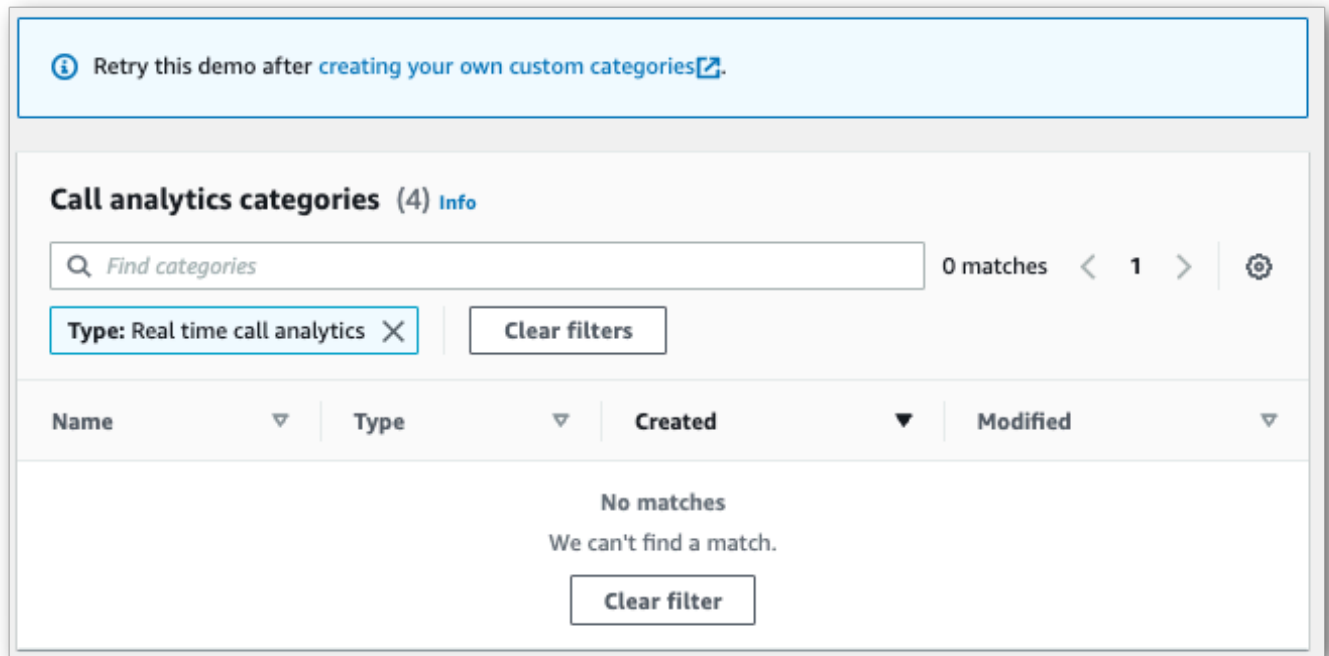
### Step 1: Specify input audio

Input audio file

Insurance complaints (en-US)	▲
Insurance complaints (en-US)	✓
Hospitality complaints (en-US)	

3. 對於步驟 2：檢視通話類別，您可以選擇檢視先前建立的即時通話分析類別。所有即時通話分析類別都會套用至您的轉錄。

選擇檢視類別會開啟新窗格，顯示您現有的即時通話分析類別，並提供建立新類別的連結。



4. 對於步驟 3：配置輸入和輸出，您可以選擇套用其他設定。

選擇配置進階設定會開啟新窗格，您可以在其中指定內容修訂設定。

Use the following options to identify or redact content from your transcript. Other settings such as Custom Vocabulary, Custom Language Models, Partial results stabilization, Vocabulary Filtering are available through the API, SDK, CLI

## ▼ Content removal

### PII Identification & redaction [Info](#)

Identify or redact one or more types of personally identifiable information (PII) in your transcript

#### Select PII detection type

##### Identification only

Label the type of PII identified but not redact it in the transcription output

##### Identification & redaction

Label the type of PII and also mask the content with the PII entity type in the transcription output.  
For example, (123)456-7890 will be masked as [PHONE]

#### Select PII entity types (11 of 11 selected)

##### Select All

##### Financial (6 of 6 selected)

BANK\_ACCOUNT\_NUMBER

BANK\_ROUTING

CREDIT\_DEBIT\_NUMBER

CREDIT\_DEBIT\_CVV

CREDIT\_DEBIT\_EXPIRY

PIN

##### Personal (5 of 5 selected)


NAME

ADDRESS

PHONE

EMAIL

SSN

 The updates that you make here will only be applied when you start stream again.

Cancel

Save

完成所有選擇後，選擇儲存，返回主頁面。

- 若要套用其他分析，您可以開啟通話後分析。這提供與通話後分析轉錄相同的分析，包括中斷、響度、非通話時間、通話速度、通話時間、問題、行動項目和結果。通話後分析輸出會在與即時通話分析文字記錄不同的檔案儲存。

#### Post-call Analytics

Post-call analytics enabled with real-time analytics provides consolidated transcript and audio backup, with the associated analytics, along with further insights such as call summaries and conversation characteristics like non-talk time, interruptions, loudness, and talk speed, after the end of the call in the provided Amazon S3 bucket.

Post-call Analytics

如果您套用通話後分析，則必須指定 Amazon S3 輸出檔案目的地和 IAM 角色。您可以選擇加密輸出。

**Post-call Analytics**  
Post-call analytics enabled with real-time analytics provides consolidated transcript and audio backup, with the associated analytics, along with further insights such as call summaries and conversation characteristics like non-talk time, interruptions, loudness, and talk speed, after the end of the call in the provided Amazon S3 bucket.

Post-call Analytics

**Output file destination on S3** [Info](#)  
Choose the location to store the output of the post-call analytics. If you input a location in an Amazon S3 bucket that doesn't yet exist, it will be created for you.

Resource URI

Format: s3://bucket, s3://bucket/prefix/, or s3://bucket/prefix/object.

Encryption [Info](#)

**IAM role** [Info](#)

[Create an IAM role](#) that grants access to the output bucket and KMS key (if specified) with the trust policy shown below

▶ Trust Policy

## 6. 選擇開始串流。

### HTTP/2 串流

此範例會建立啟用通話分析的 HTTP/2 請求。如需搭配使用 HTTP/2 串流的詳細資訊 Amazon Transcribe，請參閱。[設定 HTTP/2 串流](#)如需特定參數和標頭的詳細資訊 Amazon Transcribe，請參閱[StartCallAnalyticsStreamTranscription](#)。

此範例包括[通話後分析](#)。如果您不想進行通話後分析，請從請求中移除該 `PostCallAnalyticsSettings` 區段。

請注意，下列範例中顯示的組態事件必須作為串流中的第一個事件傳遞。

```
POST /stream-transcription HTTP/2
host: transcribestreaming.us-west-2.amazonaws.com
X-Amz-Target: com.amazonaws.transcribe.Transcribe.StartCallAnalyticsStreamTranscription
Content-Type: application/vnd.amazon.eventstream
X-Amz-Content-Sha256: string
X-Amz-Date: 20220208T235959Z
Authorization: AWS4-HMAC-SHA256 Credential=access-key/20220208/us-west-2/transcribe/
aws4_request, SignedHeaders=content-type;host;x-amz-content-sha256;x-amz-date;x-amz-
target;x-amz-security-token, Signature=string
x-amzn-transcribe-language-code: en-US
x-amzn-transcribe-media-encoding: flac
x-amzn-transcribe-sample-rate: 16000
transfer-encoding: chunked

{
```

```

"AudioStream": {
  "AudioEvent": {
    "AudioChunk": blob
  },
  "ConfigurationEvent": {
    "ChannelDefinitions": [
      {
        "ChannelId": 0,
        "ParticipantRole": "AGENT"
      },
      {
        "ChannelId": 1,
        "ParticipantRole": "CUSTOMER"
      }
    ],
    "PostCallAnalyticsSettings": {
      "OutputLocation": "s3://DOC-EXAMPLE-BUCKET/my-output-files/",
      "DataAccessRoleArn": "arn:aws:iam::111122223333:role/ExampleRole"
    }
  }
}
}
}

```

您可以在 [API 參考](#) 中找到參數定義；所有 [AWS API](#) 作業的共用參數會列在「[一般參數](#)」區段中。

## WebSocket 流

此範例會建立在 WebSocket 串流中使用呼叫分析的預先簽署 URL。已加入分行符號以提高可讀性。如需搭配使用 WebSocket 串流的詳細資訊 Amazon Transcribe，請參閱 [設定串 WebSocket 流](#)。如需參數詳細資訊，請參閱 [StartCallAnalyticsStreamTranscription](#)。

此範例包括 [通話後分析](#)。如果您不想進行通話後分析，請從請求中移除該 `PostCallAnalyticsSettings` 區段。

請注意，下列範例中顯示的組態事件必須作為串流中的第一個事件傳遞。

```

GET wss://transcribestreaming.us-west-2.amazonaws.com:8443/call-analytics-stream-
transcription-websocket?
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIAIOSFODNN7EXAMPLE%2F20220208%2Fus-
west-2%2Ftranscribe%2Faws4_request
&X-Amz-Date=20220208T235959Z
&X-Amz-Expires=300

```

```

&X-Amz-Security-Token=security-token
&X-Amz-Signature=string
&X-Amz-SignedHeaders=content-type%3Bhost%3Bx-amz-date
&language-code=en-US
&media-encoding=flac
&sample-rate=16000

{
  "AudioStream": {
    "AudioEvent": {
      "AudioChunk": blob
    },
    "ConfigurationEvent": {
      "ChannelDefinitions": [
        {
          "ChannelId": 0,
          "ParticipantRole": "AGENT"
        },
        {
          "ChannelId": 1,
          "ParticipantRole": "CUSTOMER"
        }
      ],
      "PostCallAnalyticsSettings": {
        "OutputLocation": "s3://DOC-EXAMPLE-BUCKET/my-output-files/",
        "DataAccessRoleArn": "arn:aws:iam::111122223333:role/ExampleRole"
      }
    }
  }
}

```

您可以在 [API 參考](#) 中找到參數定義；所有 [AWS API](#) 作業的共用參數會列在「[一般參數](#)」區段中。

### Tip

上述 HTTP/2 和 WebSocket 示例包括調用後分析。如果您不想進行通話後分析，請從請求中移除該 `PostCallAnalyticsSettings` 區段。

如果啟用 `PostCallAnalyticsSettings`，則必須傳送組態事件作為第一個事件。如前面的範例所示，您的組態事件包括 `ChannelDenifitions` 和 `PostStreamAnalyticsSettings` 的設定。

二進位資料會連同 `content-type application/octet-stream` 以二進位訊息的格式傳遞，且組態事件會連同 `content-type application/json` 以文字訊息的格式傳遞。

如需詳細資訊，請參閱 [設定串流轉錄](#)。

## 建立類別配對的即時警示

若要設定即時警示，您必須先建立帶 REAL\_TIME 標記的 [TranscriptFilterType](#) 類別。此標記可讓您的類別套用至即時通話分析轉錄。

如需建立新類別的說明，請參閱 [建立即時轉錄的類別](#)。

您開始進行即時通話分析轉錄時，所有有 REAL\_TIME 標識的類別都會自動套用至區段層級的轉錄輸出。如果有 TranscriptFilterType 相符項目，會在您的文字記錄 CategoryEvent 區段下方出現。然後，您可以使用此參數及其子參數，MatchedCategories 及 MatchedDetails 設定自訂即時警示。

以下是 CategoryEvent 相符項目的即時通話分析轉錄輸出的範例：

```
"CategoryEvent": {
  "MatchedCategories": [ "shipping-complaint" ],
  "MatchedDetails": {
    "my package never arrived" : {
      "TimestampRanges": [
        {
          "BeginOffsetMillis": 19010,
          "EndOffsetMillis": 22690
        }
      ]
    }
  }
},
```

前一個範例代表與「我的包裹從未送達」的語音完全相符的文字，代表屬於「shipping-complaint」類別中的規則。

您可以設定即時警示，以包含所列參數的任何組合。例如，您可以將警示設定為僅包含符合的片語 (MatchedDetails) 或僅包含類別名稱 (MatchedCategories)。或者，您可以將警示設定為包含所有參數。

設定即時警示的方式需視組織的介面和所需的警示類型而定。例如，您可以設定 CategoryEvent 配對以傳送快顯通知、電子郵件、文字或系統可接受的任何其他警示。

## 即時通話分析輸出

即時通話分析記錄會以區段的 turn-by-turn 格式顯示。其中包括類別事件、問題偵測、情緒，以及 PII 識別和修訂。類別事件可讓您設定即時警示；請參閱 [建立類別配對的即時警示](#)，了解詳細資訊。

若要提高準確性並進一步根據您的使用案例自訂文字記錄，例如包括產業專用術語，請在您的通話分析要求中新增 [自訂詞彙](#) 或 [自訂語言模型](#)。若要遮蔽、移除或標記轉錄結果中不想要的字詞 (例如褻瀆性文字)，請新增 [字彙篩選](#) 功能。

以下章節顯示即時通話分析轉錄的 JSON 輸出範例。

### 類別事件

以下是轉錄輸出中類別配對的外觀。此範例顯示 19010 毫秒時間戳記到 22690 毫秒時間戳記的音訊與「network-complaint」類別相符。在這種情況下，自訂「network-complaint」類別要求客戶說出「網路問題」(完全相符的詞)。

```
"CategoryEvent": {
  "MatchedCategories": [
    "network-complaint"
  ],
  "MatchedDetails": {
    "network issues" : {
      "TimestampRanges": [
        {
          "BeginOffsetMillis": 9299375,
          "EndOffsetMillis": 7899375
        }
      ]
    }
  }
},
```

### 問題偵測

以下是轉錄輸出中問題偵測配對的外觀。該範例顯示從字元 26 到字元 62 的文字已描述問題。

```
"UtteranceEvent": {
  ...
  "Transcript": "Wang Xiulan I'm tired of the network issues my phone is having.",
  ...
  "IssuesDetected": [
```



```

    {
      "CharacterOffsets": {
        "BeginOffsetChar": 26,
        "EndOffsetChar": 62
      }
    }
  ],
},

```

## 情緒

以下是轉錄輸出中情緒分析的外觀。

```

"UtteranceEvent": {
  ...
  "Sentiment": "NEGATIVE",
  "Items": [{
    ...
  ]
}

```

## PII 識別

以下是轉錄輸出中 PII 識別的外觀。

```

"Entities": [
  {
    "Content": "Wang Xiulan",
    "Category": "PII",
    "Type": "NAME",
    "BeginOffsetMillis": 7999375,
    "EndOffsetMillis": 199375,
    "Confidence": 0.9989
  }
],

```

## PII 修訂

以下是轉錄輸出中 PII 修訂的外觀。

```

"Content": "[NAME]. Hi, [NAME]. I'm [NAME] Happy to be helping you today.",
"Redaction": {
  "RedactedTimestamps": [
    {

```

```

        "BeginOffsetMillis": 32670,
        "EndOffsetMillis": 33343
    },
    {
        "BeginOffsetMillis": 33518,
        "EndOffsetMillis": 33858
    },
    {
        "BeginOffsetMillis": 34068,
        "EndOffsetMillis": 34488
    }
]
},

```

## 編譯的即時通話分析輸出

為了簡潔起見，以下轉錄輸出的某些內容將以省略符號取代。

```

{
  "CallAnalyticsTranscriptResultStream": {
    "BadRequestException": {},
    "ConflictException": {},
    "InternalFailureException": {},
    "LimitExceededException": {},
    "ServiceUnavailableException": {},
    "UtteranceEvent": {
      "UtteranceId": "58c27f92-7277-11ec-90d6-0242ac120003",
      "ParticipantRole": "CUSTOMER",
      "IsPartial": false,
      "Transcript": "Wang Xiulan I'm tired of the network issues my phone is
having.",
      "BeginOffsetMillis": 19010,
      "EndOffsetMillis": 22690,
      "Sentiment": "NEGATIVE",
      "Items": [{
        "Content": "Wang",
        "BeginOffsetMillis": 379937,
        "EndOffsetMillis": 299375,
        "Type": "pronunciation",
        "Confidence": 0.9961,
        "VocabularyFilterMatch": false
      }],
    }
  }
}

```

```
        "Content": "Xiulan",
        "EndOffsetMillis": 5899375,
        "BeginOffsetMillis": 3899375,
        "Type": "pronunciation",
        "Confidence": 0.9961,
        "VocabularyFilterMatch": false
    },
    ...
    {
        "Content": "network",
        "EndOffsetMillis": 199375,
        "BeginOffsetMillis": 9299375,
        "Type": "pronunciation",
        "Confidence": 0.9961,
        "VocabularyFilterMatch": false
    },
    {
        "Content": "issues",
        "EndOffsetMillis": 7899375,
        "BeginOffsetMillis": 5999375,
        "Type": "pronunciation",
        "Confidence": 0.9961,
        "VocabularyFilterMatch": false
    },
    {
        "Content": "my",
        "EndOffsetMillis": 9199375,
        "BeginOffsetMillis": 7999375,
        "Type": "pronunciation",
        "Confidence": 0.9961,
        "VocabularyFilterMatch": false
    },
    {
        "Content": "phone",
        "EndOffsetMillis": 199375,
        "BeginOffsetMillis": 9299375,
        "Type": "pronunciation",
        "Confidence": 0.9961,
        "VocabularyFilterMatch": false
    },
    ...
],
"Entities": [{
    "Content": "Wang Xiulan",
```

```
        "Category": "PII",
        "Type": "NAME",
        "BeginOffsetMillis": 7999375,
        "EndOffsetMillis": 199375,
        "Confidence": 0.9989
    }],
    "IssuesDetected": [{
        "CharacterOffsets": {
            "BeginOffsetChar": 26,
            "EndOffsetChar": 62
        }
    }
  ],
  "CategoryEvent": {
    "MatchedCategories": [
      "network-complaint"
    ],
    "MatchedDetails": {
      "network issues" : {
        "TimestampRanges": [
          {
            "BeginOffsetMillis": 9299375,
            "EndOffsetMillis": 7899375
          }
        ]
      }
    }
  }
}
```

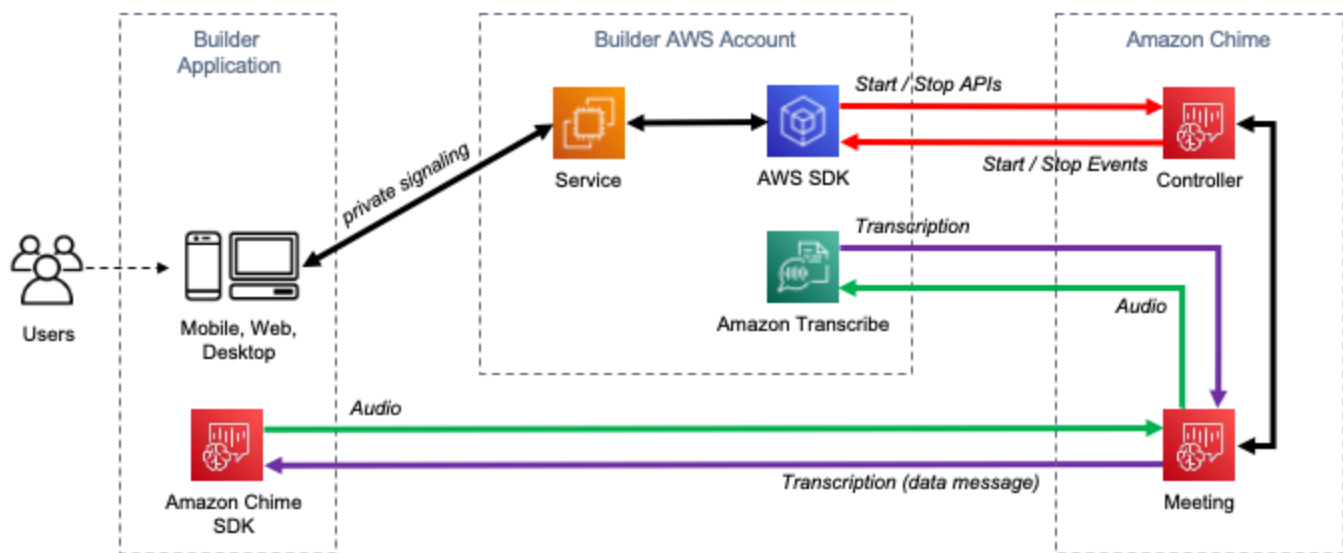
## 實時轉錄您的Amazon Chime電話

Amazon Transcribe與Amazon Chime SDK 集成，便於實時轉錄您的Amazon Chime呼叫。

當您使用Amazon Chime SDK API 要求轉錄時，會Amazon Chime開始串流音訊，Amazon Transcribe並在通話期間繼續進行。

Amazon ChimeSDK 使用其「活動講話器」算法來選擇前兩個活躍的發言者，然後通過單個流將其音頻Amazon Transcribe作為兩個單獨的通道發送。會議參與者會透過Amazon Chime SDK 資料訊息收到使用者歸屬的轉錄。您可以在 [Amazon ChimeSDK 開發人員指南](#) 中檢視傳遞範例。

Amazon Chime轉錄的數據流描述如下圖所示：



如需有關如何設定即時Amazon Chime轉錄的其他資訊和詳細指示，請參閱 [Amazon ChimeSDK 開發人員指南](#) 中的 [使用Amazon Chime SDK 即時轉譯](#)。如需 API 作業，請參閱 [Amazon ChimeSDK API 參考資料](#)。

### 📌 透過 Machine LAWS earning 部落格深入瞭解

若要深入瞭解如何透過即時轉譯提升準確性，請參閱：

- [Amazon ChimeSDK 會議現在支援使用Amazon Transcribe和即時轉錄Amazon Transcribe Medical](#)
- [Amazon Chime遠程醫療解決方案的 SDK](#)

# 使用 SDK 進行 Amazon Transcribe 的代碼示例 AWS

下列程式碼範例說明如何搭配 AWS 軟體開發套件 (SDK) 使用 Amazon Transcribe。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境和跨服務範例中查看內容中的動作。

Scenarios (案例) 是向您展示如何呼叫相同服務中的多個函數來完成特定任務的程式碼範例。

Cross-service examples (跨服務範例) 是跨多個 AWS 服務執行的應用程式範例。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱[搭配 AWS SDK 使用此服務](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

## 程式碼範例

- [使用 SDK 進行 Amazon Transcribe 的操作 AWS](#)
  - [搭CreateVocabulary配 AWS 開發套件或 CLI 使用](#)
  - [搭DeleteMedicalTranscriptionJob配 AWS 開發套件或 CLI 使用](#)
  - [搭DeleteTranscriptionJob配 AWS 開發套件或 CLI 使用](#)
  - [搭DeleteVocabulary配 AWS 開發套件或 CLI 使用](#)
  - [搭GetTranscriptionJob配 AWS 開發套件或 CLI 使用](#)
  - [搭GetVocabulary配 AWS 開發套件或 CLI 使用](#)
  - [搭ListMedicalTranscriptionJobs配 AWS 開發套件或 CLI 使用](#)
  - [搭ListTranscriptionJobs配 AWS 開發套件或 CLI 使用](#)
  - [搭ListVocabularies配 AWS 開發套件或 CLI 使用](#)
  - [搭StartMedicalTranscriptionJob配 AWS 開發套件或 CLI 使用](#)
  - [搭StartStreamTranscriptionAsync配 AWS 開發套件或 CLI 使用](#)
  - [搭StartTranscriptionJob配 AWS 開發套件或 CLI 使用](#)
  - [搭UpdateVocabulary配 AWS 開發套件或 CLI 使用](#)
- [使用軟體開發套件 Amazon Transcribe 的案例 AWS](#)
  - [使用開發套件建立和優化 Amazon Transcribe 自訂詞彙 AWS](#)
  - [使 Transcribe 開發套件使用 Amazon Transcribe 音訊和取得任務資料 AWS](#)
- [使用 SDK 進行 Amazon Transcribe 的跨服務示例 AWS](#)
  - [建置 Amazon Transcribe 應用程式](#)

- [建置 Amazon Transcribe 串流應用程式](#)
- [使用 AWS SDK 將文本轉換為語音並返回文本](#)

## 使用 SDK 進行 Amazon Transcribe 的操作 AWS

下列程式碼範例示範如何使 AWS 用 SDK 執行個別 Amazon Transcribe 動作。這些摘錄會呼叫 Amazon Transcribe API，是必須在內容中執行之大型程式的程式碼摘錄。每個範例都包含一個連結 GitHub，您可以在其中找到設定和執行程式碼的指示。

下列範例僅包含最常使用的動作。如需完整清單，請參閱 [《Amazon Transcribe API 參考資料》](#)。

### 範例

- [搭CreateVocabulary配 AWS 開發套件或 CLI 使用](#)
- [搭DeleteMedicalTranscriptionJob配 AWS 開發套件或 CLI 使用](#)
- [搭DeleteTranscriptionJob配 AWS 開發套件或 CLI 使用](#)
- [搭DeleteVocabulary配 AWS 開發套件或 CLI 使用](#)
- [搭GetTranscriptionJob配 AWS 開發套件或 CLI 使用](#)
- [搭GetVocabulary配 AWS 開發套件或 CLI 使用](#)
- [搭ListMedicalTranscriptionJobs配 AWS 開發套件或 CLI 使用](#)
- [搭ListTranscriptionJobs配 AWS 開發套件或 CLI 使用](#)
- [搭ListVocabularies配 AWS 開發套件或 CLI 使用](#)
- [搭StartMedicalTranscriptionJob配 AWS 開發套件或 CLI 使用](#)
- [搭StartStreamTranscriptionAsync配 AWS 開發套件或 CLI 使用](#)
- [搭StartTranscriptionJob配 AWS 開發套件或 CLI 使用](#)
- [搭UpdateVocabulary配 AWS 開發套件或 CLI 使用](#)

## 搭CreateVocabulary配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用CreateVocabulary。

動作範例是大型程式的程式碼摘錄，必須在內容中執行。您可以在下列程式碼範例的內容中看到此動作：

- [建立和優化自訂詞彙](#)

## .NET

### AWS SDK for .NET

#### Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
/// <summary>
/// Create a custom vocabulary using a list of phrases. Custom vocabularies
/// improve transcription accuracy for one or more specific words.
/// </summary>
/// <param name="languageCode">The language code of the vocabulary.</param>
/// <param name="phrases">Phrases to use in the vocabulary.</param>
/// <param name="vocabularyName">Name for the vocabulary.</param>
/// <returns>The state of the custom vocabulary.</returns>
public async Task<VocabularyState> CreateCustomVocabulary(LanguageCode
languageCode,
    List<string> phrases, string vocabularyName)
{
    var response = await _amazonTranscribeService.CreateVocabularyAsync(
        new CreateVocabularyRequest
        {
            LanguageCode = languageCode,
            Phrases = phrases,
            VocabularyName = vocabularyName
        });
    return response.VocabularyState;
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for .NET API 參考[CreateVocabulary](#)中的。

## CLI

### AWS CLI

#### 建立自訂詞彙



以下 `create-vocabulary` 範例會建立自訂詞彙。若要建立自訂詞彙，您必須先建立一個文字檔案，其中包含要更準確轉錄的所有字詞。對於 `vocabulary-file-uri`，指定該文字檔案的亞馬遜簡單儲存服務 (Amazon S3) URI。針對 `language-code`，指定對應您的自訂詞彙語言的語言代碼。針對 `vocabulary-name`，指定您自訂詞彙的稱呼。

```
aws transcribe create-vocabulary \  
  --language-code language-code \  
  --vocabulary-name cli-vocab-example \  
  --vocabulary-file-uri s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/the-text-file-  
for-the-custom-vocabulary.txt
```

輸出：

```
{  
  "VocabularyName": "cli-vocab-example",  
  "LanguageCode": "language-code",  
  "VocabularyState": "PENDING"  
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的[自訂詞彙](#)。

- 如需 API 詳細資訊，請參閱 AWS CLI 命令參考 [CreateVocabulary](#) 中的。

## Python

適用於 Python (Boto3) 的 SDK

### Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
def create_vocabulary(  
    vocabulary_name, language_code, transcribe_client, phrases=None,  
    table_uri=None  
):  
    """  
    Creates a custom vocabulary that can be used to improve the accuracy of  
    transcription jobs. This function returns as soon as the vocabulary  
    processing
```

```

is started. Call get_vocabulary to get the current status of the vocabulary.
The vocabulary is ready to use when its status is 'READY'.

:param vocabulary_name: The name of the custom vocabulary.
:param language_code: The language code of the vocabulary.
                       For example, en-US or nl-NL.
:param transcribe_client: The Boto3 Transcribe client.
:param phrases: A list of comma-separated phrases to include in the
vocabulary.
:param table_uri: A table of phrases and pronunciation hints to include in
the
                   vocabulary.
:return: Information about the newly created vocabulary.
"""
try:
    vocab_args = {"VocabularyName": vocabulary_name, "LanguageCode":
language_code}
    if phrases is not None:
        vocab_args["Phrases"] = phrases
    elif table_uri is not None:
        vocab_args["VocabularyFileUri"] = table_uri
    response = transcribe_client.create_vocabulary(**vocab_args)
    logger.info("Created custom vocabulary %s.", response["VocabularyName"])
except ClientError:
    logger.exception("Couldn't create custom vocabulary %s.",
vocabulary_name)
    raise
else:
    return response

```

- 如需 API 的詳細資訊，請參閱AWS 開發套件[CreateVocabulary](#)中的 Python (博托 3) API 參考。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱[搭配 AWS SDK 使用此服務](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

## 搭配DeleteMedicalTranscriptionJob配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用DeleteMedicalTranscriptionJob。

## .NET

### AWS SDK for .NET

#### Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
/// <summary>
/// Delete a medical transcription job. Also deletes the transcript
associated with the job.
/// </summary>
/// <param name="jobName">Name of the medical transcription job to delete.</
param>
/// <returns>True if successful.</returns>
public async Task<bool> DeleteMedicalTranscriptionJob(string jobName)
{
    var response = await
        _amazonTranscribeService.DeleteMedicalTranscriptionJobAsync(
            new DeleteMedicalTranscriptionJobRequest()
            {
                MedicalTranscriptionJobName = jobName
            });
    return response.HttpStatusCode == HttpStatusCode.OK;
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for .NET API 參考[DeleteMedicalTranscriptionJob](#)中的。

## CLI

### AWS CLI

#### 刪除醫學轉錄作業

以下 delete-medical-transcription-job 範例會刪除醫學轉錄作業。

```
aws transcribe delete-medical-transcription-job \  
  --medical-transcription-job-name medical-transcription-job-name
```

此命令不會產生輸出。

如需詳細資訊，請參閱 Amazon Transcribe 開發人員指南 [DeleteMedicalTranscriptionJob](#) 中的。

- 如需 API 詳細資訊，請參閱 AWS CLI 命令參考 [DeleteMedicalTranscriptionJob](#) 中的。

## JavaScript

適用於 JavaScript (v3) 的開發套件

### Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

建立用戶端。

```
import { TranscribeClient } from "@aws-sdk/client-transcribe";  
// Set the AWS Region.  
const REGION = "REGION"; //e.g. "us-east-1"  
// Create an Amazon Transcribe service client object.  
const transcribeClient = new TranscribeClient({ region: REGION });  
export { transcribeClient };
```

刪除醫學轉錄作業。

```
// Import the required AWS SDK clients and commands for Node.js  
import { DeleteMedicalTranscriptionJobCommand } from "@aws-sdk/client-transcribe";  
import { transcribeClient } from "../libs/transcribeClient.js";  
  
// Set the parameters  
export const params = {  
  MedicalTranscriptionJobName: "MEDICAL_JOB_NAME", // For example,  
  'medical_transcription_demo'  
};
```

```
export const run = async () => {
  try {
    const data = await transcribeClient.send(
      new DeleteMedicalTranscriptionJobCommand(params)
    );
    console.log("Success - deleted");
    return data; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

- 如需詳細資訊，請參閱 [《AWS SDK for JavaScript 開發人員指南》](#)。
- 如需 API 詳細資訊，請參閱 AWS SDK for JavaScript API 參考 [DeleteMedicalTranscriptionJob](#) 中的。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 [搭配 AWS SDK 使用此服務](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

## 搭 `DeleteTranscriptionJob` 配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用 `DeleteTranscriptionJob`。

動作範例是大型程式的程式碼摘錄，必須在內容中執行。您可以在下列程式碼範例的內容中看到此動作：

- [建立和優化自訂詞彙](#)

.NET

AWS SDK for .NET

### Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
/// <summary>
/// Delete a transcription job. Also deletes the transcript associated with
the job.
/// </summary>
/// <param name="jobName">Name of the transcription job to delete.</param>
/// <returns>True if successful.</returns>
public async Task<bool> DeleteTranscriptionJob(string jobName)
{
    var response = await
_amazonTranscribeService.DeleteTranscriptionJobAsync(
    new DeleteTranscriptionJobRequest()
    {
        TranscriptionJobName = jobName
    });
    return response.HttpStatusCode == HttpStatusCode.OK;
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for .NET API 參考[DeleteTranscriptionJob](#)中的。

## CLI

### AWS CLI

#### 刪除其中一項轉錄作業

以下 delete-transcription-job 範例會刪除其中一項轉錄作業。

```
aws transcribe delete-transcription-job \
    --transcription-job-name your-transcription-job
```

此命令不會產生輸出。

如需詳細資訊，請參閱 Amazon Transcribe 開發人員指南[DeleteTranscriptionJob](#)中的。

- 如需 API 詳細資訊，請參閱 AWS CLI 命令參考[DeleteTranscriptionJob](#)中的。

## JavaScript

### 適用於 JavaScript (v3) 的開發套件

#### Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

#### 刪除轉錄作業。

```
// Import the required AWS SDK clients and commands for Node.js
import { DeleteTranscriptionJobCommand } from "@aws-sdk/client-transcribe";
import { transcribeClient } from "../libs/transcribeClient.js";

// Set the parameters
export const params = {
  TranscriptionJobName: "JOB_NAME", // Required. For example, 'transcription_demo'
};

export const run = async () => {
  try {
    const data = await transcribeClient.send(
      new DeleteTranscriptionJobCommand(params)
    );
    console.log("Success - deleted");
    return data; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

#### 建立用戶端。

```
import { TranscribeClient } from "@aws-sdk/client-transcribe";
// Set the AWS Region.
const REGION = "REGION"; //e.g. "us-east-1"
// Create an Amazon Transcribe service client object.
```

```
const transcribeClient = new TranscribeClient({ region: REGION });
export { transcribeClient };
```

- 如需詳細資訊，請參閱 [《AWS SDK for JavaScript 開發人員指南》](#)。
- 如需 API 詳細資訊，請參閱 AWS SDK for JavaScript API 參考 [DeleteTranscriptionJob](#) 中的。

## Python

### 適用於 Python (Boto3) 的 SDK

#### Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
def delete_job(job_name, transcribe_client):
    """
    Deletes a transcription job. This also deletes the transcript associated with
    the job.

    :param job_name: The name of the job to delete.
    :param transcribe_client: The Boto3 Transcribe client.
    """
    try:
        transcribe_client.delete_transcription_job(TranscriptionJobName=job_name)
        logger.info("Deleted job %s.", job_name)
    except ClientError:
        logger.exception("Couldn't delete job %s.", job_name)
        raise
```

- 如需 API 的詳細資訊，請參閱 AWS 開發套件 [DeleteTranscriptionJob](#) 中的 Python (博托 3) API 參考。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 [搭配 AWS SDK 使用此服務](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。



## 搭DeleteVocabulary配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用DeleteVocabulary。

動作範例是大型程式的程式碼摘錄，必須在內容中執行。您可以在下列程式碼範例的內容中看到此動作：

- [建立和優化自訂詞彙](#)

.NET

AWS SDK for .NET

### Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
/// <summary>
/// Delete an existing custom vocabulary.
/// </summary>
/// <param name="vocabularyName">Name of the vocabulary to delete.</param>
/// <returns>True if successful.</returns>
public async Task<bool> DeleteCustomVocabulary(string vocabularyName)
{
    var response = await _amazonTranscribeService.DeleteVocabularyAsync(
        new DeleteVocabularyRequest
        {
            VocabularyName = vocabularyName
        });
    return response.HttpStatusCode == HttpStatusCode.OK;
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for .NET API 參考[DeleteVocabulary](#)中的。

## CLI

### AWS CLI

#### 刪除自訂詞彙

以下 `delete-vocabulary` 範例會刪除自訂詞彙。

```
aws transcribe delete-vocabulary \  
  --vocabulary-name vocabulary-name
```

此命令不會產生輸出。

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的 [自訂詞彙](#)。

- 如需 API 詳細資訊，請參閱AWS CLI 命令參考 [DeleteVocabulary](#) 中的。

## Python

### 適用於 Python (Boto3) 的 SDK

#### Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
def delete_vocabulary(vocabulary_name, transcribe_client):  
    """  
    Deletes a custom vocabulary.  
  
    :param vocabulary_name: The name of the vocabulary to delete.  
    :param transcribe_client: The Boto3 Transcribe client.  
    """  
    try:  
        transcribe_client.delete_vocabulary(VocabularyName=vocabulary_name)  
        logger.info("Deleted vocabulary %s.", vocabulary_name)  
    except ClientError:  
        logger.exception("Couldn't delete vocabulary %s.", vocabulary_name)  
        raise
```

- 如需 API 的詳細資訊，請參閱AWS 開發套件[DeleteVocabulary](#)中的 Python (博托 3) API 參考。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱[搭配 AWS SDK 使用此服務](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

## 搭配 GetTranscriptionJob 配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用 GetTranscriptionJob。

動作範例是大型程式的程式碼摘錄，必須在內容中執行。您可以在下列程式碼範例的內容中看到此動作：

- [建立和優化自訂詞彙](#)
- [轉錄音訊並取得工作資料](#)

.NET

AWS SDK for .NET

### Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
/// <summary>
/// Get details about a transcription job.
/// </summary>
/// <param name="jobName">A unique name for the transcription job.</param>
/// <returns>A TranscriptionJob instance with information on the requested
job.</returns>
public async Task<TranscriptionJob> GetTranscriptionJob(string jobName)
{
    var response = await _amazonTranscribeService.GetTranscriptionJobAsync(
```

```
        new GetTranscriptionJobRequest()
        {
            TranscriptionJobName = jobName
        });
    return response.TranscriptionJob;
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for .NET API 參考[GetTranscriptionJob](#)中的。

## CLI

### AWS CLI

#### 取得特定轉錄作業的相關資訊

以下 `get-transcription-job` 範例會取得特定轉錄作業的相關資訊。若要存取轉錄結果，請使用 `TranscriptFileUri` 參數。使用 `MediaFileUri` 參數可查看您在此工作中轉錄的音訊檔案。您可以使用 `Settings` 物件來查看您在轉錄作業中已啟用的選用功能。

```
aws transcribe get-transcription-job \
    --transcription-job-name your-transcription-job
```

輸出：

```
{
  "TranscriptionJob": {
    "TranscriptionJobName": "your-transcription-job",
    "TranscriptionJobStatus": "COMPLETED",
    "LanguageCode": "language-code",
    "MediaSampleRateHertz": 48000,
    "MediaFormat": "mp4",
    "Media": {
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.file-
extension"
    },
    "Transcript": {
      "TranscriptFileUri": "https://Amazon-S3-file-location-of-
transcription-output"
    },
  },
}
```

```
"StartTime": "2020-09-18T22:27:23.970000+00:00",
"CreationTime": "2020-09-18T22:27:23.948000+00:00",
"CompletionTime": "2020-09-18T22:28:21.197000+00:00",
"Settings": {
    "ChannelIdentification": false,
    "ShowAlternatives": false
},
"IdentifyLanguage": true,
"IdentifiedLanguageScore": 0.8672199249267578
}
}
```

如需詳細資訊，請參閱 [Amazon Transcribe 開發人員指南中的入門 \(AWS 命令列界面\)](#)。

- 如需 API 詳細資訊，請參閱 AWS CLI 命令參考 [GetTranscriptionJob](#) 中的。

## Python

適用於 Python (Boto3) 的 SDK

### Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
def get_job(job_name, transcribe_client):
    """
    Gets details about a transcription job.

    :param job_name: The name of the job to retrieve.
    :param transcribe_client: The Boto3 Transcribe client.
    :return: The retrieved transcription job.
    """
    try:
        response = transcribe_client.get_transcription_job(
            TranscriptionJobName=job_name
        )
        job = response["TranscriptionJob"]
        logger.info("Got job %s.", job["TranscriptionJobName"])
    except ClientError:
        logger.exception("Couldn't get job %s.", job_name)
```

```
        raise
    else:
        return job
```

- 如需 API 的詳細資訊，請參閱AWS 開發套件[GetTranscriptionJob](#)中的 Python (博托 3) API 參考。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱[搭配 AWS SDK 使用此服務](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

## 搭GetVocabulary配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用GetVocabulary。

動作範例是大型程式的程式碼摘錄，必須在內容中執行。您可以在下列程式碼範例的內容中看到此動作：

- [建立和優化自訂詞彙](#)

.NET

AWS SDK for .NET

### Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
/// <summary>
/// Get information about a custom vocabulary.
/// </summary>
/// <param name="vocabularyName">Name of the vocabulary.</param>
/// <returns>The state of the custom vocabulary.</returns>
public async Task<VocabularyState> GetCustomVocabulary(string vocabularyName)
```

```
{
    var response = await _amazonTranscribeService.GetVocabularyAsync(
        new GetVocabularyRequest()
        {
            VocabularyName = vocabularyName
        });
    return response.VocabularyState;
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for .NET API 參考[GetVocabulary](#)中的。

## CLI

### AWS CLI

#### 取得自訂詞彙的相關資訊

以下 `get-vocabulary` 範例會取得先前建立的自訂詞彙的相關資訊。

```
aws transcribe get-vocabulary \
    --vocabulary-name cli-vocab-1
```

#### 輸出：

```
{
  "VocabularyName": "cli-vocab-1",
  "LanguageCode": "language-code",
  "VocabularyState": "READY",
  "LastModifiedTime": "2020-09-19T23:22:32.836000+00:00",
  "DownloadUri": "https://link-to-download-the-text-file-used-to-create-your-custom-vocabulary"
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的[自訂詞彙](#)。

- 如需 API 詳細資訊，請參閱 AWS CLI 命令參考[GetVocabulary](#)中的。

## Python

適用於 Python (Boto3) 的 SDK

### Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
def get_vocabulary(vocabulary_name, transcribe_client):
    """
    Gets information about a custom vocabulary.

    :param vocabulary_name: The name of the vocabulary to retrieve.
    :param transcribe_client: The Boto3 Transcribe client.
    :return: Information about the vocabulary.
    """
    try:
        response =
transcribe_client.get_vocabulary(VocabularyName=vocabulary_name)
        logger.info("Got vocabulary %s.", response["VocabularyName"])
    except ClientError:
        logger.exception("Couldn't get vocabulary %s.", vocabulary_name)
        raise
    else:
        return response
```

- 如需 API 的詳細資訊，請參閱AWS 開發套件[GetVocabulary](#)中的 Python (博托 3) API 參考。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱[搭配 AWS SDK 使用此服務](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

## 搭ListMedicalTranscriptionJobs配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用ListMedicalTranscriptionJobs。



## .NET

### AWS SDK for .NET

#### Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
/// <summary>
/// List medical transcription jobs, optionally with a name filter.
/// </summary>
/// <param name="jobNameContains">Optional name filter for the medical
transcription jobs.</param>
/// <returns>A list of summaries about medical transcription jobs.</returns>
public async Task<List<MedicalTranscriptionJobSummary>>
ListMedicalTranscriptionJobs(
    string? jobNameContains = null)
{
    var response = await
_amazonTranscribeService.ListMedicalTranscriptionJobsAsync(
    new ListMedicalTranscriptionJobsRequest()
    {
        JobNameContains = jobNameContains
    });
    return response.MedicalTranscriptionJobSummaries;
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for .NET API 參考[ListMedicalTranscriptionJobs](#)中的。

## CLI

### AWS CLI

列出醫學轉錄作業

下列 `list-medical-transcription-jobs` 範例會列出與您的 AWS 帳戶和地區相關聯的醫療轉錄工作。若要取得有關特定轉錄工作的詳細資訊，請複製轉錄輸出中的 `MedicalTranscriptionJobName` 參數值，然後為命令的 `MedicalTranscriptionJobName` 選項指定該值。`get-medical-transcription-job` 若要查看更多轉錄工作，請複製 `NextToken` 參數的值，再次執行 `list-medical-transcription-jobs` 命令，然後在 `--next-token` 選項中指定該值。

```
aws transcribe list-medical-transcription-jobs
```

輸出：

```
{
  "NextToken": "3/PblzkiGhzjER3KHuQt2fmbPLF7cDYafjFMEoGn440N/
gsuUSTIkGyanvRE6WMXfd/ZTEc2EZj+P9eii/
z102FDYli6RLI0WoRX4RwMisVrh9G0Kie0Y8ikBCdtqLZB10Wa9McC+eb0l
+LaDtZPC4u6ttoHLRL1EfzqstHXSgapXg3tEBtm9piIaPB6M0M5BB6t86+qtmocTR/
qrteHZBBudhTfbcWhsxaqujHiiUvFdm3BQbKKWIW06yV9b+4f38oD2lVIan
+vfUs3gBYA15VTDmXXzQPBOHPjtwmFI+IWX15nSUjWuN3TUylHgPWzDaYT8qBtu0Z+3UG4V6b
+K2CC0XszXg5rBq9hYgNzy4XoFh/6s5DoSnzq49Q9xHgHdT2yBADFmvFK7myZBsJ75+2vQZ0SvpWUPy3WT/32zFAc
+mFYfUjtTZ8n/jq7aQEjQ42A
+X/7K6Jg0cdVPtEg8P1Dr5kgYYG3q30mYXX37U3FZuJmnTI63VtIXsNn0U5eGoY0btpk00Nq9UkzgSJxqj84ZD5n
+S0EGy9ZUYBJRRcGeYUM3Q4DbSJfUwSAqcFdLIWZdp8qIREMQIBWy7BLwSdyqsQo2vRrd53hm5aWM7SVf6pPq6X/
IXR5+1eU00D8/coaTT4ES2DerbV6RkV4o0VT1d0SdVX/
MmtkNG8nYj8PqU07w7988quh1ZP6D80veJS1q73tUUR9MjnGernW2tAnvnLNhdefBcD
+sZVfYq3iBMFY7wTy1P1G6NqW9GrYDYox3tTPW1D7phpbVSyKrh/
PdYrps5UxnsGoA1b7L/FfAXDfUoGrGUB4N3JsPYXX9D++g+6gV1qBBs/
Wff934aKqfD6UTggm/zV3GA0WiBpfvAZRvEb924i6yGHYMC7y5401ZAwSBupmI
+FFd13CaP04kN1vJlth6aM5vUPXg4BpyUhtbRhwD/KxCvf9K0tLJGyL1A==",
  "MedicalTranscriptionJobSummaries": [
    {
      "MedicalTranscriptionJobName": "vocabulary-dictation-medical-
transcription-job",
      "CreationTime": "2020-09-21T21:17:27.016000+00:00",
      "StartTime": "2020-09-21T21:17:27.045000+00:00",
      "CompletionTime": "2020-09-21T21:17:59.561000+00:00",
      "LanguageCode": "en-US",
      "TranscriptionJobStatus": "COMPLETED",
      "OutputLocationType": "CUSTOMER_BUCKET",
      "Specialty": "PRIMARYCARE",
      "Type": "DICTATION"
    },
    {
```

```
    "MedicalTranscriptionJobName": "alternatives-dictation-medical-
transcription-job",
    "CreationTime": "2020-09-21T21:01:14.569000+00:00",
    "StartTime": "2020-09-21T21:01:14.592000+00:00",
    "CompletionTime": "2020-09-21T21:01:43.606000+00:00",
    "LanguageCode": "en-US",
    "TranscriptionJobStatus": "COMPLETED",
    "OutputLocationType": "CUSTOMER_BUCKET",
    "Specialty": "PRIMARYCARE",
    "Type": "DICTATION"
  },
  {
    "MedicalTranscriptionJobName": "alternatives-conversation-medical-
transcription-job",
    "CreationTime": "2020-09-21T19:09:18.171000+00:00",
    "StartTime": "2020-09-21T19:09:18.199000+00:00",
    "CompletionTime": "2020-09-21T19:10:22.516000+00:00",
    "LanguageCode": "en-US",
    "TranscriptionJobStatus": "COMPLETED",
    "OutputLocationType": "CUSTOMER_BUCKET",
    "Specialty": "PRIMARYCARE",
    "Type": "CONVERSATION"
  },
  {
    "MedicalTranscriptionJobName": "speaker-id-conversation-medical-
transcription-job",
    "CreationTime": "2020-09-21T18:43:37.157000+00:00",
    "StartTime": "2020-09-21T18:43:37.265000+00:00",
    "CompletionTime": "2020-09-21T18:44:21.192000+00:00",
    "LanguageCode": "en-US",
    "TranscriptionJobStatus": "COMPLETED",
    "OutputLocationType": "CUSTOMER_BUCKET",
    "Specialty": "PRIMARYCARE",
    "Type": "CONVERSATION"
  },
  {
    "MedicalTranscriptionJobName": "multichannel-conversation-medical-
transcription-job",
    "CreationTime": "2020-09-20T23:46:44.053000+00:00",
    "StartTime": "2020-09-20T23:46:44.081000+00:00",
    "CompletionTime": "2020-09-20T23:47:35.851000+00:00",
    "LanguageCode": "en-US",
    "TranscriptionJobStatus": "COMPLETED",
    "OutputLocationType": "CUSTOMER_BUCKET",
```

```
        "Specialty": "PRIMARYCARE",
        "Type": "CONVERSATION"
    }
]
}
```

如需詳細資訊，請參閱 Amazon Transcribe 開發人員指南中的 <https://docs.aws.amazon.com/transcribe/latest/dg/batch-med-transcription.html>。

- 如需 API 詳細資訊，請參閱 AWS CLI 命令參考 [ListMedicalTranscriptionJobs](#) 中的。

## JavaScript

適用於 JavaScript (v3) 的開發套件

### Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

建立用戶端。

```
import { TranscribeClient } from "@aws-sdk/client-transcribe";
// Set the AWS Region.
const REGION = "REGION"; //e.g. "us-east-1"
// Create an Amazon Transcribe service client object.
const transcribeClient = new TranscribeClient({ region: REGION });
export { transcribeClient };
```

列出醫學轉錄作業。

```
// Import the required AWS SDK clients and commands for Node.js
import { StartMedicalTranscriptionJobCommand } from "@aws-sdk/client-transcribe";
import { transcribeClient } from "../libs/transcribeClient.js";

// Set the parameters
export const params = {
  MedicalTranscriptionJobName: "MEDICAL_JOB_NAME", // Required
  OutputBucketName: "OUTPUT_BUCKET_NAME", // Required
```

```
Specialty: "PRIMARYCARE", // Required. Possible values are 'PRIMARYCARE'
Type: "JOB_TYPE", // Required. Possible values are 'CONVERSATION' and
'DICTATION'
LanguageCode: "LANGUAGE_CODE", // For example, 'en-US'
MediaFormat: "SOURCE_FILE_FORMAT", // For example, 'wav'
Media: {
  MediaFileUri: "SOURCE_FILE_LOCATION",
  // 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. For example,
  // "https://transcribe-demo.s3-REGION.amazonaws.com/hello_world.wav"
},
};

export const run = async () => {
  try {
    const data = await transcribeClient.send(
      new StartMedicalTranscriptionJobCommand(params)
    );
    console.log("Success - put", data);
    return data; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

- 如需詳細資訊，請參閱 [《AWS SDK for JavaScript 開發人員指南》](#)。
- 如需 API 詳細資訊，請參閱 AWS SDK for JavaScript API 參考 [ListMedicalTranscriptionJobs](#) 中的。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 [搭配 AWS SDK 使用此服務](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

## 搭 ListTranscriptionJobs 配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用 ListTranscriptionJobs。

## .NET

### AWS SDK for .NET

#### Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
/// <summary>
/// List transcription jobs, optionally with a name filter.
/// </summary>
/// <param name="jobNameContains">Optional name filter for the transcription
jobs.</param>
/// <returns>A list of transcription job summaries.</returns>
public async Task<List<TranscriptionJobSummary>>
ListTranscriptionJobs(string? jobNameContains = null)
{
    var response = await _amazonTranscribeService.ListTranscriptionJobsAsync(
        new ListTranscriptionJobsRequest()
        {
            JobNameContains = jobNameContains
        });
    return response.TranscriptionJobSummaries;
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for .NET API 參考[ListTranscriptionJobs](#)中的。

## CLI

### AWS CLI

列出您的轉錄作業

下列list-transcription-jobs範例會列出與您的 AWS 帳戶和區域相關聯的轉錄工作。

```
aws transcribe list-transcription-jobs
```

輸出：

```
{
  "NextToken": "NextToken",
  "TranscriptionJobSummaries": [
    {
      "TranscriptionJobName": "speak-id-job-1",
      "CreationTime": "2020-08-17T21:06:15.391000+00:00",
      "StartTime": "2020-08-17T21:06:15.416000+00:00",
      "CompletionTime": "2020-08-17T21:07:05.098000+00:00",
      "LanguageCode": "language-code",
      "TranscriptionJobStatus": "COMPLETED",
      "OutputLocationType": "SERVICE_BUCKET"
    },
    {
      "TranscriptionJobName": "job-1",
      "CreationTime": "2020-08-17T20:50:24.207000+00:00",
      "StartTime": "2020-08-17T20:50:24.230000+00:00",
      "CompletionTime": "2020-08-17T20:52:18.737000+00:00",
      "LanguageCode": "language-code",
      "TranscriptionJobStatus": "COMPLETED",
      "OutputLocationType": "SERVICE_BUCKET"
    },
    {
      "TranscriptionJobName": "sdk-test-job-4",
      "CreationTime": "2020-08-17T20:32:27.917000+00:00",
      "StartTime": "2020-08-17T20:32:27.956000+00:00",
      "CompletionTime": "2020-08-17T20:33:15.126000+00:00",
      "LanguageCode": "language-code",
      "TranscriptionJobStatus": "COMPLETED",
      "OutputLocationType": "SERVICE_BUCKET"
    },
    {
      "TranscriptionJobName": "Diarization-speak-id",
      "CreationTime": "2020-08-10T22:10:09.066000+00:00",
      "StartTime": "2020-08-10T22:10:09.116000+00:00",
      "CompletionTime": "2020-08-10T22:26:48.172000+00:00",
      "LanguageCode": "language-code",
      "TranscriptionJobStatus": "COMPLETED",
      "OutputLocationType": "SERVICE_BUCKET"
    },
    {
      "TranscriptionJobName": "your-transcription-job-name",
      "CreationTime": "2020-07-29T17:45:09.791000+00:00",
```

```
        "StartTime": "2020-07-29T17:45:09.826000+00:00",
        "CompletionTime": "2020-07-29T17:46:20.831000+00:00",
        "LanguageCode": "language-code",
        "TranscriptionJobStatus": "COMPLETED",
        "OutputLocationType": "SERVICE_BUCKET"
    }
]
}
```

如需詳細資訊，請參閱 [Amazon Transcribe 開發人員指南中的入門 \(AWS 命令列界面\)](#)。

- 如需 API 詳細資訊，請參閱 AWS CLI 命令參考 [ListTranscriptionJobs](#) 中的。

## Java

適用於 Java 2.x 的 SDK

### Note

還有更多關於 [GitHub](#)。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

```
public class ListTranscriptionJobs {
    public static void main(String[] args) {
        TranscribeClient transcribeClient = TranscribeClient.builder()
            .region(Region.US_EAST_1)
            .build();

        listTranscriptionJobs(transcribeClient);
    }

    public static void listTranscriptionJobs(TranscribeClient
transcribeClient) {
        ListTranscriptionJobsRequest listJobsRequest =
ListTranscriptionJobsRequest.builder()
            .build();

        transcribeClient.listTranscriptionJobsPaginator(listJobsRequest).stream()
            .flatMap(response ->
response.transcriptionJobSummaries().stream())
```



```
        .forEach(jobSummary -> {
            System.out.println("Job Name: " +
                jobSummary.transcriptionJobName());
            System.out.println("Job Status: " +
                jobSummary.transcriptionJobStatus());
            System.out.println("Output Location: " +
                jobSummary.outputLocationType());
            // Add more information as needed

            // Retrieve additional details for the job if necessary
            GetTranscriptionJobResponse jobDetails =
                transcribeClient.getTranscriptionJob(
                    GetTranscriptionJobRequest.builder()

                        .transcriptionJobName(jobSummary.transcriptionJobName())
                        .build());

            // Display additional details
            System.out.println("Language Code: " +
                jobDetails.transcriptionJob().languageCode());
            System.out.println("Media Format: " +
                jobDetails.transcriptionJob().mediaFormat());
            // Add more details as needed

            System.out.println("-----");
        });
    }
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Java 2.x API 參考 [ListTranscriptionJobs](#) 中的。

## JavaScript

適用於 JavaScript (v3) 的開發套件

### Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

列出轉錄作業。

```
// Import the required AWS SDK clients and commands for Node.js

import { ListTranscriptionJobsCommand } from "@aws-sdk/client-transcribe";
import { transcribeClient } from "../libs/transcribeClient.js";

// Set the parameters
export const params = {
  JobNameContains: "KEYWORD", // Not required. Returns only transcription
  // job names containing this string
};

export const run = async () => {
  try {
    const data = await transcribeClient.send(
      new ListTranscriptionJobsCommand(params)
    );
    console.log("Success", data.TranscriptionJobSummaries);
    return data; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

建立用戶端。

```
import { TranscribeClient } from "@aws-sdk/client-transcribe";
// Set the AWS Region.
const REGION = "REGION"; //e.g. "us-east-1"
// Create an Amazon Transcribe service client object.
const transcribeClient = new TranscribeClient({ region: REGION });
export { transcribeClient };
```

- 如需詳細資訊，請參閱 [《AWS SDK for JavaScript 開發人員指南》](#)。
- 如需 API 詳細資訊，請參閱 AWS SDK for JavaScript API 參考 [ListTranscriptionJobs](#) 中的。

## Python

### 適用於 Python (Boto3) 的 SDK

#### Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
def list_jobs(job_filter, transcribe_client):
    """
    Lists summaries of the transcription jobs for the current AWS account.

    :param job_filter: The list of returned jobs must contain this string in
    their
                       names.
    :param transcribe_client: The Boto3 Transcribe client.
    :return: The list of retrieved transcription job summaries.
    """
    try:
        response =
transcribe_client.list_transcription_jobs(JobNameContains=job_filter)
        jobs = response["TranscriptionJobSummaries"]
        next_token = response.get("NextToken")
        while next_token is not None:
            response = transcribe_client.list_transcription_jobs(
                JobNameContains=job_filter, NextToken=next_token
            )
            jobs += response["TranscriptionJobSummaries"]
            next_token = response.get("NextToken")
        logger.info("Got %s jobs with filter %s.", len(jobs), job_filter)
    except ClientError:
        logger.exception("Couldn't get jobs with filter %s.", job_filter)
        raise
    else:
        return jobs
```

- 如需 API 的詳細資訊，請參閱AWS 開發套件[ListTranscriptionJobs](#)中的 Python (博托 3) API 參考。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱[搭配 AWS SDK 使用此服務](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

## 搭ListVocabularies配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用ListVocabularies。

動作範例是大型程式的程式碼摘錄，必須在內容中執行。您可以在下列程式碼範例的內容中看到此動作：

- [建立和優化自訂詞彙](#)

.NET

AWS SDK for .NET

### Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
/// <summary>
/// List custom vocabularies for the current account. Optionally specify a
name
/// filter and a specific state to filter the vocabularies list.
/// </summary>
/// <param name="nameContains">Optional string the vocabulary name must
contain.</param>
/// <param name="stateEquals">Optional state of the vocabulary.</param>
/// <returns>List of information about the vocabularies.</returns>
public async Task<List<VocabularyInfo>> ListCustomVocabularies(string?
nameContains = null,
    VocabularyState? stateEquals = null)
{
    var response = await _amazonTranscribeService.ListVocabulariesAsync(
```

```
        new ListVocabulariesRequest()
        {
            NameContains = nameContains,
            StateEquals = stateEquals
        });
    return response.Vocabularies;
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for .NET API 參考 [ListVocabularies](#) 中的。

## CLI

### AWS CLI

#### 列出您的自訂詞彙

下列 `list-vocabularies` 範例會列出與您的 AWS 帳戶和地區相關聯的自訂字彙。

```
aws transcribe list-vocabularies
```

輸出：

```
{
  "NextToken": "NextToken",
  "Vocabularies": [
    {
      "VocabularyName": "ards-test-1",
      "LanguageCode": "language-code",
      "LastModifiedTime": "2020-04-27T22:00:27.330000+00:00",
      "VocabularyState": "READY"
    },
    {
      "VocabularyName": "sample-test",
      "LanguageCode": "language-code",
      "LastModifiedTime": "2020-04-24T23:04:11.044000+00:00",
      "VocabularyState": "READY"
    },
    {
      "VocabularyName": "CRLF-to-LF-test-3-1",
      "LanguageCode": "language-code",
      "LastModifiedTime": "2020-04-24T22:12:22.277000+00:00",
```

```

        "VocabularyState": "READY"
    },
    {
        "VocabularyName": "CRLF-to-LF-test-2",
        "LanguageCode": "language-code",
        "LastModifiedTime": "2020-04-24T21:53:50.455000+00:00",
        "VocabularyState": "READY"
    },
    {
        "VocabularyName": "CRLF-to-LF-1-1",
        "LanguageCode": "language-code",
        "LastModifiedTime": "2020-04-24T21:39:33.356000+00:00",
        "VocabularyState": "READY"
    }
]
}

```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的[自訂詞彙](#)。

- 如需 API 詳細資訊，請參閱AWS CLI 命令參考[ListVocabularies](#)中的。

## Python

適用於 Python (Boto3) 的 SDK

### Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```

def list_vocabularies(vocabulary_filter, transcribe_client):
    """
    Lists the custom vocabularies created for this AWS account.

    :param vocabulary_filter: The returned vocabularies must contain this string
    in
                               their names.
    :param transcribe_client: The Boto3 Transcribe client.
    :return: The list of retrieved vocabularies.
    """
    try:

```

```
response =
transcribe_client.list_vocabularies(NameContains=vocabulary_filter)
vocabs = response["Vocabularies"]
next_token = response.get("NextToken")
while next_token is not None:
    response = transcribe_client.list_vocabularies(
        NameContains=vocabulary_filter, NextToken=next_token
    )
    vocabs += response["Vocabularies"]
    next_token = response.get("NextToken")
logger.info(
    "Got %s vocabularies with filter %s.", len(vocabs), vocabulary_filter
)
except ClientError:
    logger.exception(
        "Couldn't list vocabularies with filter %s.", vocabulary_filter
    )
    raise
else:
    return vocabs
```

- 如需 API 的詳細資訊，請參閱AWS 開發套件[ListVocabularies](#)中的 Python (博托 3) API 參考。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱[搭配 AWS SDK 使用此服務](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

## 搭StartMedicalTranscriptionJob配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用StartMedicalTranscriptionJob。

## .NET

### AWS SDK for .NET

#### Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
/// <summary>
/// Start a medical transcription job for a media file. This method returns
/// as soon as the job is started.
/// </summary>
/// <param name="jobName">A unique name for the medical transcription job.</
param>
/// <param name="mediaFileUri">The URI of the media file, typically an Amazon
S3 location.</param>
/// <param name="mediaFormat">The format of the media file.</param>
/// <param name="outputBucketName">Location for the output, typically an
Amazon S3 location.</param>
/// <param name="transcriptionType">Conversation or dictation transcription
type.</param>
/// <returns>A MedicalTransactionJob instance with information on the new
job.</returns>
public async Task<MedicalTranscriptionJob> StartMedicalTranscriptionJob(
    string jobName, string mediaFileUri,
    MediaFormat mediaFormat, string outputBucketName,
    Amazon.TranscribeService.Type transcriptionType)
{
    var response = await
    _amazonTranscribeService.StartMedicalTranscriptionJobAsync(
        new StartMedicalTranscriptionJobRequest()
        {
            MedicalTranscriptionJobName = jobName,
            Media = new Media()
            {
                MediaFileUri = mediaFileUri
            },
            MediaFormat = mediaFormat,
            LanguageCode =
```



```

        LanguageCode
            .EnUS, // The value must be en-US for medical
transcriptions.
        OutputBucketName = outputBucketName,
        OutputKey =
            jobName, // The value is a key used to fetch the output of
the transcription.
        Specialty = Specialty.PRIMARYCARE, // The value PRIMARYCARE must
be set.
        Type = transcriptionType
    });
    return response.MedicalTranscriptionJob;
}

```

- 如需 API 詳細資訊，請參閱 AWS SDK for .NET API 參考 [StartMedicalTranscriptionJob](#) 中的。

## CLI

### AWS CLI

#### 範例 1：轉錄儲存為音訊檔案的醫學聽寫

以下 `start-medical-transcription-job` 範例會轉錄音訊檔案。您可在 `OutputBucketName` 參數中指定轉錄輸出的位置。

```
aws transcribe start-medical-transcription-job \
  --cli-input-json file://myfile.json
```

`myfile.json` 的內容：

```
{
  "MedicalTranscriptionJobName": "simple-dictation-medical-transcription-job",
  "LanguageCode": "language-code",
  "Specialty": "PRIMARYCARE",
  "Type": "DICTATION",
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
  }
}
```

```
}
```

輸出：

```
{
  "MedicalTranscriptionJob": {
    "MedicalTranscriptionJobName": "simple-dictation-medical-transcription-
job",
    "TranscriptionJobStatus": "IN_PROGRESS",
    "LanguageCode": "language-code",
    "Media": {
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
    },
    "StartTime": "2020-09-20T00:35:22.256000+00:00",
    "CreationTime": "2020-09-20T00:35:22.218000+00:00",
    "Specialty": "PRIMARYCARE",
    "Type": "DICTATION"
  }
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的[批次轉錄概觀](#)。

#### 範例 2：轉錄儲存為音訊檔案的臨床醫師與病患對話

以下 `start-medical-transcription-job` 範例會轉錄包含臨床醫師與病患對話的音訊檔案。您可以在 `OutputBucketName` 參數中指定轉錄輸出的位置。

```
aws transcribe start-medical-transcription-job \
  --cli-input-json file://mysecondfile.json
```

`mysecondfile.json` 的內容：

```
{
  "MedicalTranscriptionJobName": "simple-dictation-medical-transcription-job",
  "LanguageCode": "language-code",
  "Specialty": "PRIMARYCARE",
  "Type": "CONVERSATION",
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
  }
}
```

輸出：

```
{
  "MedicalTranscriptionJob": {
    "MedicalTranscriptionJobName": "simple-conversation-medical-
transcription-job",
    "TranscriptionJobStatus": "IN_PROGRESS",
    "LanguageCode": "language-code",
    "Media": {
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
    },
    "StartTime": "2020-09-20T23:19:49.965000+00:00",
    "CreationTime": "2020-09-20T23:19:49.941000+00:00",
    "Specialty": "PRIMARYCARE",
    "Type": "CONVERSATION"
  }
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的[批次轉錄概觀](#)。

### 範例 3：轉錄臨床醫師與病患對話的多聲道音訊檔案

以下 `start-medical-transcription-job` 範例會轉錄音訊檔案中每個聲道的音訊，並將每個聲道的個別轉錄合併成單一轉錄輸出。您可在 `OutputBucketName` 參數中指定轉錄輸出的位置。

```
aws transcribe start-medical-transcription-job \
  --cli-input-json file://mythirdfile.json
```

`mythirdfile.json` 的內容：

```
{
  "MedicalTranscriptionJobName": "multichannel-conversation-medical-
transcription-job",
  "LanguageCode": "language-code",
  "Specialty": "PRIMARYCARE",
  "Type": "CONVERSATION",
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
  },
  "Settings": {
    "ChannelIdentification": true
  }
}
```

```
}
}
```

輸出：

```
{
  "MedicalTranscriptionJob": {
    "MedicalTranscriptionJobName": "multichannel-conversation-medical-
transcription-job",
    "TranscriptionJobStatus": "IN_PROGRESS",
    "LanguageCode": "language-code",
    "Media": {
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
    },
    "StartTime": "2020-09-20T23:46:44.081000+00:00",
    "CreationTime": "2020-09-20T23:46:44.053000+00:00",
    "Settings": {
      "ChannelIdentification": true
    },
    "Specialty": "PRIMARYCARE",
    "Type": "CONVERSATION"
  }
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的[聲道識別](#)。

範例 4：轉錄臨床醫生與病患對話的音訊檔案，並識別轉錄輸出中的發言者

以下 `start-medical-transcription-job` 範例會轉錄音訊檔案，並標記轉錄輸出中每一位發言者的語音。您可在 `OutputBucketName` 參數中指定轉錄輸出的位置。

```
aws transcribe start-medical-transcription-job \
  --cli-input-json file://myfourthfile.json
```

`myfourthfile.json` 的內容：

```
{
  "MedicalTranscriptionJobName": "speaker-id-conversation-medical-
transcription-job",
  "LanguageCode": "language-code",
  "Specialty": "PRIMARYCARE",
  "Type": "CONVERSATION",
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",
```

```

"Media": {
  "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
},
"Settings":{
  "ShowSpeakerLabels": true,
  "MaxSpeakerLabels": 2
}
}

```

輸出：

```

{
  "MedicalTranscriptionJob": {
    "MedicalTranscriptionJobName": "speaker-id-conversation-medical-
transcription-job",
    "TranscriptionJobStatus": "IN_PROGRESS",
    "LanguageCode": "language-code",
    "Media": {
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
    },
    "StartTime": "2020-09-21T18:43:37.265000+00:00",
    "CreationTime": "2020-09-21T18:43:37.157000+00:00",
    "Settings": {
      "ShowSpeakerLabels": true,
      "MaxSpeakerLabels": 2
    },
    "Specialty": "PRIMARYCARE",
    "Type": "CONVERSATION"
  }
}

```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的[識別發言者](#)。

**範例 5：轉錄儲存為音訊檔案且最多有兩個轉錄替代選項的醫學對話**

以下 `start-medical-transcription-job` 範例會從單一音訊檔案建立最多兩個替代轉錄。每個轉錄都有相關聯的可信度等級。根據預設，Amazon Transcribe 會傳回最高可信度等級的轉錄。您可以指定 Amazon Transcribe 傳回可信度較低的其他轉錄。您可在 `OutputBucketName` 參數中指定轉錄輸出的位置。

```

aws transcribe start-medical-transcription-job \
  --cli-input-json file://myfifthfile.json

```

myfifthfile.json 的內容：

```
{
  "MedicalTranscriptionJobName": "alternatives-conversation-medical-
transcription-job",
  "LanguageCode": "language-code",
  "Specialty": "PRIMARYCARE",
  "Type": "CONVERSATION",
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
  },
  "Settings": {
    "ShowAlternatives": true,
    "MaxAlternatives": 2
  }
}
```

輸出：

```
{
  "MedicalTranscriptionJob": {
    "MedicalTranscriptionJobName": "alternatives-conversation-medical-
transcription-job",
    "TranscriptionJobStatus": "IN_PROGRESS",
    "LanguageCode": "language-code",
    "Media": {
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
    },
    "StartTime": "2020-09-21T19:09:18.199000+00:00",
    "CreationTime": "2020-09-21T19:09:18.171000+00:00",
    "Settings": {
      "ShowAlternatives": true,
      "MaxAlternatives": 2
    },
    "Specialty": "PRIMARYCARE",
    "Type": "CONVERSATION"
  }
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的[替代轉錄](#)。

範例 6：轉錄最多有兩個替代轉錄的醫學聽寫的音訊檔案

以下 `start-medical-transcription-job` 範例會轉錄音訊檔案，並使用詞彙篩選器來為任何不希望出現的字詞加上遮罩。您可以在 `OutputBucketName` 參數中指定轉錄輸出的位置。

```
aws transcribe start-medical-transcription-job \  
  --cli-input-json file://mysixthfile.json
```

`mysixthfile.json` 的內容：

```
{  
  "MedicalTranscriptionJobName": "alternatives-conversation-medical-  
transcription-job",  
  "LanguageCode": "language-code",  
  "Specialty": "PRIMARYCARE",  
  "Type": "DICTATION",  
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",  
  "Media": {  
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"  
  },  
  "Settings": {  
    "ShowAlternatives": true,  
    "MaxAlternatives": 2  
  }  
}
```

輸出：

```
{  
  "MedicalTranscriptionJob": {  
    "MedicalTranscriptionJobName": "alternatives-dictation-medical-  
transcription-job",  
    "TranscriptionJobStatus": "IN_PROGRESS",  
    "LanguageCode": "language-code",  
    "Media": {  
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"  
    },  
    "StartTime": "2020-09-21T21:01:14.592000+00:00",  
    "CreationTime": "2020-09-21T21:01:14.569000+00:00",  
    "Settings": {  
      "ShowAlternatives": true,  
      "MaxAlternatives": 2  
    },  
    "Specialty": "PRIMARYCARE",
```

```
    "Type": "DICTATION"
  }
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的[替代轉錄](#)。

#### 範例 7：使用自訂詞彙以更準確的方式轉錄醫學聽寫的音訊檔案

以下 `start-medical-transcription-job` 範例會轉錄音訊檔案，並使用您先前建立的醫學自訂詞彙來提高轉錄準確度。您可在 `OutputBucketName` 參數中指定轉錄輸出的位置。

```
aws transcribe start-transcription-job \  
  --cli-input-json file://myseventhfile.json
```

`mysixthfile.json` 的內容：

```
{  
  "MedicalTranscriptionJobName": "vocabulary-dictation-medical-transcription-  
job",  
  "LanguageCode": "language-code",  
  "Specialty": "PRIMARYCARE",  
  "Type": "DICTATION",  
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",  
  "Media": {  
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"  
  },  
  "Settings": {  
    "VocabularyName": "cli-medical-vocab-1"  
  }  
}
```

輸出：

```
{  
  "MedicalTranscriptionJob": {  
    "MedicalTranscriptionJobName": "vocabulary-dictation-medical-  
transcription-job",  
    "TranscriptionJobStatus": "IN_PROGRESS",  
    "LanguageCode": "language-code",  
    "Media": {  
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"  
    },  
  },  
}
```



```
"StartTime": "2020-09-21T21:17:27.045000+00:00",
"CreationTime": "2020-09-21T21:17:27.016000+00:00",
"Settings": {
  "VocabularyName": "cli-medical-vocab-1"
},
"Specialty": "PRIMARYCARE",
"Type": "DICTATION"
}
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的[醫學自訂詞彙](#)。

- 如需 API 詳細資訊，請參閱AWS CLI 命令參考[StartMedicalTranscriptionJob](#)中的。

## JavaScript

適用於 JavaScript (v3) 的開發套件

### Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

建立用戶端。

```
import { TranscribeClient } from "@aws-sdk/client-transcribe";
// Set the AWS Region.
const REGION = "REGION"; //e.g. "us-east-1"
// Create an Amazon Transcribe service client object.
const transcribeClient = new TranscribeClient({ region: REGION });
export { transcribeClient };
```

開始醫學轉錄作業。

```
// Import the required AWS SDK clients and commands for Node.js
import { StartMedicalTranscriptionJobCommand } from "@aws-sdk/client-transcribe";
import { transcribeClient } from "./libs/transcribeClient.js";

// Set the parameters
```

```
export const params = {
  MedicalTranscriptionJobName: "MEDICAL_JOB_NAME", // Required
  OutputBucketName: "OUTPUT_BUCKET_NAME", // Required
  Specialty: "PRIMARYCARE", // Required. Possible values are 'PRIMARYCARE'
  Type: "JOB_TYPE", // Required. Possible values are 'CONVERSATION' and
  'DICTATION'
  LanguageCode: "LANGUAGE_CODE", // For example, 'en-US'
  MediaFormat: "SOURCE_FILE_FORMAT", // For example, 'wav'
  Media: {
    MediaFileUri: "SOURCE_FILE_LOCATION",
    // 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. For example,
    // "https://transcribe-demo.s3-REGION.amazonaws.com/hello_world.wav"
  },
};

export const run = async () => {
  try {
    const data = await transcribeClient.send(
      new StartMedicalTranscriptionJobCommand(params)
    );
    console.log("Success - put", data);
    return data; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};

run();
```

- 如需詳細資訊，請參閱 [《AWS SDK for JavaScript 開發人員指南》](#)。
- 如需 API 詳細資訊，請參閱 AWS SDK for JavaScript API 參考 [StartMedicalTranscriptionJob](#) 中的。


如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 [搭配 AWS SDK 使用此服務](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

## 搭 StartStreamTranscriptionAsync 配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用 StartStreamTranscriptionAsync。

## C++

## 適用於 C++ 的 SDK

 Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
int main() {
    Aws::SDKOptions options;

    Aws::InitAPI(options);
    {
        //TODO(User): Set to the region of your AWS account.
        const Aws::String region = Aws::Region::US_WEST_2;

        //Load a profile that has been granted AmazonTranscribeFullAccess AWS
        managed permission policy.
        Aws::Client::ClientConfiguration config;
#ifdef _WIN32
        // ATTENTION: On Windows with the AWS C++ SDK, this example only runs if
        the SDK is built
        // with the curl library.
        // For more information, see the accompanying ReadMe.
        // For more information, see "Building the SDK for Windows with curl".
        // https://docs.aws.amazon.com/sdk-for-cpp/v1/developer-guide/setup-
        windows.html
        //TODO(User): Update to the location of your .crt file.
        config.caFile = "C:/curl/bin/curl-ca-bundle.crt";
#endif
        config.region = region;

        TranscribeStreamingServiceClient client(config);
        StartStreamTranscriptionHandler handler;
        handler.SetOnErrorCallback(
            [](const Aws::Client::AWSError<TranscribeStreamingServiceErrors>
            &error) {
                std::cerr << "ERROR: " + error.GetMessage() << std::endl;
            });
    }
}
```

```

    //SetTranscriptEventCallback called for every 'chunk' of file
    transcribed.
    // Partial results are returned in real time.
    handler.SetTranscriptEventCallback([](const TranscriptEvent &ev) {
        for (auto &&r: ev.GetTranscript().GetResults()) {
            if (r.GetIsPartial()) {
                std::cout << "[partial] ";
            }
            else {
                std::cout << "[Final] ";
            }
            for (auto &&alt: r.GetAlternatives()) {
                std::cout << alt.GetTranscript() << std::endl;
            }
        }
    });

    StartStreamTranscriptionRequest request;
    request.SetMediaSampleRateHertz(SAMPLE_RATE);
    request.SetLanguageCode(LanguageCode::en_US);
    request.SetMediaEncoding(
        MediaEncoding::pcm); // wav and aiff files are PCM formats.
    request.SetEventStreamHandler(handler);

    auto OnStreamReady = [](AudioStream &stream) {
        Aws::FStream file(FILE_NAME, std::ios_base::in |
std::ios_base::binary);
        if (!file.is_open()) {
            std::cerr << "Failed to open " << FILE_NAME << '\n';
        }
        std::array<char, BUFFER_SIZE> buf;
        int i = 0;
        while (file) {
            file.read(&buf[0], buf.size());

            if (!file)
                std::cout << "File: only " << file.gcount() << " could be
read"
                    << std::endl;

            Aws::Vector<unsigned char> bits{buf.begin(), buf.end()};
            AudioEvent event(std::move(bits));
            if (!stream) {
                std::cerr << "Failed to create a stream" << std::endl;
            }
        }
    };

```

```

        break;
    }
    //The std::basic_istream::gcount() is used to count the
characters in the given string. It returns
    //the number of characters extracted by the last read()
operation.
    if (file.gcount() > 0) {
        if (!stream.WriteAudioEvent(event)) {
            std::cerr << "Failed to write an audio event" <<
std::endl;
            break;
        }
    }
    else {
        break;
    }
    std::this_thread::sleep_for(std::chrono::milliseconds(
        25)); // Slow down because we are streaming from a
file.
    }
    if (!stream.WriteAudioEvent(
        AudioEvent())) {
        // Per the spec, we have to send an empty event (an event
without a payload) at the end.
        std::cerr << "Failed to send an empty frame" << std::endl;
    }
    else {
        std::cout << "Successfully sent the empty frame" <<
std::endl;
    }
    stream.flush();
    stream.Close();
};

    Aws::Utils::Threading::Semaphore signaling(0 /*initialCount*/, 1 /
*maxCount*/);
    auto OnResponseCallback = [&signaling](
        const TranscribeStreamingServiceClient * /*unused*/,
        const Model::StartStreamTranscriptionRequest & /*unused*/,
        const Model::StartStreamTranscriptionOutcome &outcome,
        const std::shared_ptr<const Aws::Client::AsyncCallerContext> & /
*unused*/) {

        if (!outcome.IsSuccess()) {

```

```
        std::cerr << "Transcribe streaming error "
                << outcome.GetError().GetMessage() << std::endl;
    }

    signaling.Release();
};

std::cout << "Starting..." << std::endl;
client.StartStreamTranscriptionAsync(request, OnStreamReady,
OnResponseCallback,
                                     nullptr /*context*/);
    signaling.WaitOne(); // Prevent the application from exiting until we're
done.
    std::cout << "Done" << std::endl;
}

Aws::ShutdownAPI(options);

return 0;
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for C++ API 參考[StartStreamTranscriptionAsync](#)中的。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱[搭配 AWS SDK 使用此服務](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

## 搭配 StartTranscriptionJob 配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用 StartTranscriptionJob。

動作範例是大型程式的程式碼摘錄，必須在內容中執行。您可以在下列程式碼範例的內容中看到此動作：

- [建立和優化自訂詞彙](#)
- [轉錄音訊並取得工作資料](#)

## .NET

### AWS SDK for .NET

#### Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
/// <summary>
/// Start a transcription job for a media file. This method returns
/// as soon as the job is started.
/// </summary>
/// <param name="jobName">A unique name for the transcription job.</param>
/// <param name="mediaFileUri">The URI of the media file, typically an Amazon
S3 location.</param>
/// <param name="mediaFormat">The format of the media file.</param>
/// <param name="languageCode">The language code of the media file, such as
en-US.</param>
/// <param name="vocabularyName">Optional name of a custom vocabulary.</
param>
/// <returns>A TranscriptionJob instance with information on the new job.</
returns>
public async Task<TranscriptionJob> StartTranscriptionJob(string jobName,
string mediaFileUri,
MediaFormat mediaFormat, LanguageCode languageCode, string?
vocabularyName)
{
    var response = await _amazonTranscribeService.StartTranscriptionJobAsync(
        new StartTranscriptionJobRequest()
        {
            TranscriptionJobName = jobName,
            Media = new Media()
            {
                MediaFileUri = mediaFileUri
            },
            MediaFormat = mediaFormat,
            LanguageCode = languageCode,
            Settings = vocabularyName != null ? new Settings()
            {
```

```
        VocabularyName = vocabularyName
    } : null
    });
    return response.TranscriptionJob;
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for .NET API 參考[StartTranscriptionJob](#)中的。

## CLI

### AWS CLI

#### 範例 1：轉錄音訊檔案

以下 `start-transcription-job` 範例會轉錄您的音訊檔案。

```
aws transcribe start-transcription-job \  
  --cli-input-json file://myfile.json
```

`myfile.json` 的內容：

```
{  
  "TranscriptionJobName": "cli-simple-transcription-job",  
  "LanguageCode": "the-language-of-your-transcription-job",  
  "Media": {  
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-  
file-name.file-extension"  
  }  
}
```

如需詳細資訊，請參閱 [Amazon Transcribe 開發人員指南中的入門 \(AWS 命令列界面\)](#)。

#### 範例 2：轉錄多聲道音訊檔案

以下 `start-transcription-job` 範例會轉錄您的多聲道音訊檔案。

```
aws transcribe start-transcription-job \  
  --cli-input-json file://mysecondfile.json
```



mysecondfile.json 的內容：

```
{
  "TranscriptionJobName": "cli-channelid-job",
  "LanguageCode": "the-language-of-your-transcription-job",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-
file-name.file-extension"
  },
  "Settings":{
    "ChannelIdentification":true
  }
}
```

輸出：

```
{
  "TranscriptionJob": {
    "TranscriptionJobName": "cli-channelid-job",
    "TranscriptionJobStatus": "IN_PROGRESS",
    "LanguageCode": "the-language-of-your-transcription-job",
    "Media": {
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-
file-name.file-extension"
    },
    "StartTime": "2020-09-17T16:07:56.817000+00:00",
    "CreationTime": "2020-09-17T16:07:56.784000+00:00",
    "Settings": {
      "ChannelIdentification": true
    }
  }
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的[轉錄多聲道音訊](#)。

範例 3：轉錄音訊檔案並識別不同的發言者

以下 start-transcription-job 範例會轉錄音訊檔案，並識別轉錄輸出中的發言者。

```
aws transcribe start-transcription-job \
  --cli-input-json file://mythirdfile.json
```

mythirdfile.json 的內容：

```
{
  "TranscriptionJobName": "cli-speakerid-job",
  "LanguageCode": "the-language-of-your-transcription-job",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-
file-name.file-extension"
  },
  "Settings":{
    "ShowSpeakerLabels": true,
    "MaxSpeakerLabels": 2
  }
}
```

輸出：

```
{
  "TranscriptionJob": {
    "TranscriptionJobName": "cli-speakerid-job",
    "TranscriptionJobStatus": "IN_PROGRESS",
    "LanguageCode": "the-language-of-your-transcription-job",
    "Media": {
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-
file-name.file-extension"
    },
    "StartTime": "2020-09-17T16:22:59.696000+00:00",
    "CreationTime": "2020-09-17T16:22:59.676000+00:00",
    "Settings": {
      "ShowSpeakerLabels": true,
      "MaxSpeakerLabels": 2
    }
  }
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的[識別發言者](#)。

**範例 4：轉錄音訊檔案，並在轉錄輸出中為任何不希望出現的字詞加上遮罩**

以下 start-transcription-job 範例會轉錄音訊檔案，並使用您先前建立的詞彙篩選器來為任何不希望出現的字詞加上遮罩。

```
aws transcribe start-transcription-job \
```

```
--cli-input-json file://myfourthfile.json
```

myfourthfile.json 的內容：

```
{
  "TranscriptionJobName": "cli-filter-mask-job",
  "LanguageCode": "the-language-of-your-transcription-job",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-
file-name.file-extension"
  },
  "Settings":{
    "VocabularyFilterName": "your-vocabulary-filter",
    "VocabularyFilterMethod": "mask"
  }
}
```

輸出：

```
{
  "TranscriptionJob": {
    "TranscriptionJobName": "cli-filter-mask-job",
    "TranscriptionJobStatus": "IN_PROGRESS",
    "LanguageCode": "the-language-of-your-transcription-job",
    "Media": {
      "MediaFileUri": "s3://Amazon-S3-Prefix/your-media-file.file-
extension"
    },
    "StartTime": "2020-09-18T16:36:18.568000+00:00",
    "CreationTime": "2020-09-18T16:36:18.547000+00:00",
    "Settings": {
      "VocabularyFilterName": "your-vocabulary-filter",
      "VocabularyFilterMethod": "mask"
    }
  }
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的[篩選轉錄](#)。

範例 5：轉錄音訊檔案，並移除轉錄輸出中任何不希望出現的字詞

以下 start-transcription-job 範例會轉錄音訊檔案，並使用您先前建立的詞彙篩選器來為任何不希望出現的字詞加上遮罩。

```
aws transcribe start-transcription-job \  
  --cli-input-json file://myfifthfile.json
```

myfifthfile.json 的內容：

```
{  
  "TranscriptionJobName": "cli-filter-remove-job",  
  "LanguageCode": "the-language-of-your-transcription-job",  
  "Media": {  
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-  
file-name.file-extension"  
  },  
  "Settings": {  
    "VocabularyFilterName": "your-vocabulary-filter",  
    "VocabularyFilterMethod": "remove"  
  }  
}
```

輸出：

```
{  
  "TranscriptionJob": {  
    "TranscriptionJobName": "cli-filter-remove-job",  
    "TranscriptionJobStatus": "IN_PROGRESS",  
    "LanguageCode": "the-language-of-your-transcription-job",  
    "Media": {  
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-  
file-name.file-extension"  
    },  
    "StartTime": "2020-09-18T16:36:18.568000+00:00",  
    "CreationTime": "2020-09-18T16:36:18.547000+00:00",  
    "Settings": {  
      "VocabularyFilterName": "your-vocabulary-filter",  
      "VocabularyFilterMethod": "remove"  
    }  
  }  
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的[篩選轉錄](#)。

範例 6：使用自訂詞彙以更準確的方式轉錄音訊檔案

以下 `start-transcription-job` 範例會轉錄音訊檔案，並使用您先前建立的詞彙篩選器來為任何不希望出現的字詞加上遮罩。

```
aws transcribe start-transcription-job \  
  --cli-input-json file://mysixthfile.json
```

`mysixthfile.json` 的內容：

```
{  
  "TranscriptionJobName": "cli-vocab-job",  
  "LanguageCode": "the-language-of-your-transcription-job",  
  "Media": {  
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-  
file-name.file-extension"  
  },  
  "Settings": {  
    "VocabularyName": "your-vocabulary"  
  }  
}
```

輸出：

```
{  
  "TranscriptionJob": {  
    "TranscriptionJobName": "cli-vocab-job",  
    "TranscriptionJobStatus": "IN_PROGRESS",  
    "LanguageCode": "the-language-of-your-transcription-job",  
    "Media": {  
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-  
file-name.file-extension"  
    },  
    "StartTime": "2020-09-18T16:36:18.568000+00:00",  
    "CreationTime": "2020-09-18T16:36:18.547000+00:00",  
    "Settings": {  
      "VocabularyName": "your-vocabulary"  
    }  
  }  
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的[篩選轉錄](#)。

範例 7：識別音訊檔案的語言並進行轉錄

以下 `start-transcription-job` 範例會轉錄音訊檔案，並使用您先前建立的詞彙篩選器來為任何不希望出現的字詞加上遮罩。

```
aws transcribe start-transcription-job \  
  --cli-input-json file://myseventhfile.json
```

`myseventhfile.json` 的內容：

```
{  
  "TranscriptionJobName": "cli-identify-language-transcription-job",  
  "IdentifyLanguage": true,  
  "Media": {  
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-  
file-name.file-extension"  
  }  
}
```

輸出：

```
{  
  "TranscriptionJob": {  
    "TranscriptionJobName": "cli-identify-language-transcription-job",  
    "TranscriptionJobStatus": "IN_PROGRESS",  
    "Media": {  
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-  
file-name.file-extension"  
    },  
    "StartTime": "2020-09-18T22:27:23.970000+00:00",  
    "CreationTime": "2020-09-18T22:27:23.948000+00:00",  
    "IdentifyLanguage": true  
  }  
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的[識別語言](#)。

#### 範例 8：轉錄包含已修訂的個人身分識別資訊的音訊檔案

以下 `start-transcription-job` 範例會轉錄音訊檔案，並修訂轉錄輸出中的任何個人身分識別資訊。

```
aws transcribe start-transcription-job \  
  --cli-input-json file://myseventhfile.json
```

```
--cli-input-json file://myeighthfile.json
```

myeighthfile.json 的內容：

```
{
  "TranscriptionJobName": "cli-redaction-job",
  "LanguageCode": "language-code",
  "Media": {
    "MediaFileUri": "s3://Amazon-S3-Prefix/your-media-file.file-extension"
  },
  "ContentRedaction": {
    "RedactionOutput": "redacted",
    "RedactionType": "PII"
  }
}
```

輸出：

```
{
  "TranscriptionJob": {
    "TranscriptionJobName": "cli-redaction-job",
    "TranscriptionJobStatus": "IN_PROGRESS",
    "LanguageCode": "language-code",
    "Media": {
      "MediaFileUri": "s3://Amazon-S3-Prefix/your-media-file.file-extension"
    },
    "StartTime": "2020-09-25T23:49:13.195000+00:00",
    "CreationTime": "2020-09-25T23:49:13.176000+00:00",
    "ContentRedaction": {
      "RedactionType": "PII",
      "RedactionOutput": "redacted"
    }
  }
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的[自動內容修訂](#)。

**範例 9：**產生包含已修訂個人身分識別資訊 (PII) 的文字記錄和未修訂的文字記錄

以下 `start-transcription-job` 範例會產生音訊檔案的兩個轉錄，一個包含已修訂的個人身分識別資訊，另一個則不包含任何修訂。

```
aws transcribe start-transcription-job \  
  --cli-input-json file://myninthfile.json
```

myninthfile.json 的內容：

```
{  
  "TranscriptionJobName": "cli-redaction-job-with-unredacted-transcript",  
  "LanguageCode": "language-code",  
  "Media": {  
    "MediaFileUri": "s3://Amazon-S3-Prefix/your-media-file.file-extension"  
  },  
  "ContentRedaction": {  
    "RedactionOutput": "redacted_and_unredacted",  
    "RedactionType": "PII"  
  }  
}
```

輸出：

```
{  
  "TranscriptionJob": {  
    "TranscriptionJobName": "cli-redaction-job-with-unredacted-transcript",  
    "TranscriptionJobStatus": "IN_PROGRESS",  
    "LanguageCode": "language-code",  
    "Media": {  
      "MediaFileUri": "s3://Amazon-S3-Prefix/your-media-file.file-  
extension"  
    },  
    "StartTime": "2020-09-25T23:59:47.677000+00:00",  
    "CreationTime": "2020-09-25T23:59:47.653000+00:00",  
    "ContentRedaction": {  
      "RedactionType": "PII",  
      "RedactionOutput": "redacted_and_unredacted"  
    }  
  }  
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的[自動內容修訂](#)。

範例 10：使用您先前建立的自訂語言模型來轉錄音訊檔案。

以下 start-transcription-job 範例會使用您先前建立的自訂語言模型來轉錄音訊檔案。



```
aws transcribe start-transcription-job \  
  --cli-input-json file://mytenthfile.json
```

mytenthfile.json 的內容：

```
{  
  "TranscriptionJobName": "cli-clm-2-job-1",  
  "LanguageCode": "language-code",  
  "Media": {  
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.file-extension"  
  },  
  "ModelSettings": {  
    "LanguageModelName": "cli-clm-2"  
  }  
}
```

輸出：

```
{  
  "TranscriptionJob": {  
    "TranscriptionJobName": "cli-clm-2-job-1",  
    "TranscriptionJobStatus": "IN_PROGRESS",  
    "LanguageCode": "language-code",  
    "Media": {  
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.file-extension"  
    },  
    "StartTime": "2020-09-28T17:56:01.835000+00:00",  
    "CreationTime": "2020-09-28T17:56:01.801000+00:00",  
    "ModelSettings": {  
      "LanguageModelName": "cli-clm-2"  
    }  
  }  
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的[使用自訂語言模型改善特定領域的轉錄準確性](#)。

- 如需 API 詳細資訊，請參閱AWS CLI 命令參考[StartTranscriptionJob](#)中的。

## Java

### 適用於 Java 2.x 的 SDK

#### Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
public class TranscribeStreamingDemoApp {
    private static final Region REGION = Region.US_EAST_1;
    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 static Subscription currentSubscription;

    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);

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();
}
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Java 2.x API 參考[StartTranscriptionJob](#)中的。

## JavaScript

適用於 JavaScript (v3) 的開發套件

### Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

開始轉錄作業。

```
// Import the required AWS SDK clients and commands for Node.js
import { StartTranscriptionJobCommand } from "@aws-sdk/client-transcribe";
import { transcribeClient } from "./libs/transcribeClient.js";

// Set the parameters
export const params = {
  TranscriptionJobName: "JOB_NAME",
  LanguageCode: "LANGUAGE_CODE", // For example, 'en-US'
  MediaFormat: "SOURCE_FILE_FORMAT", // For example, 'wav'
  Media: {
    MediaFileUri: "SOURCE_LOCATION",
    // For example, "https://transcribe-demo.s3-REGION.amazonaws.com/
hello_world.wav"
  },
  OutputBucketName: "OUTPUT_BUCKET_NAME"
};

export const run = async () => {
  try {
    const data = await transcribeClient.send(
      new StartTranscriptionJobCommand(params)
    );
    console.log("Success - put", data);
    return data; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

建立用戶端。

```
import { TranscribeClient } from "@aws-sdk/client-transcribe";
// Set the AWS Region.
const REGION = "REGION"; //e.g. "us-east-1"
// Create an Amazon Transcribe service client object.
const transcribeClient = new TranscribeClient({ region: REGION });
export { transcribeClient };
```

- 如需詳細資訊，請參閱 [《AWS SDK for JavaScript 開發人員指南》](#)。

- 如需 API 詳細資訊，請參閱 AWS SDK for JavaScript API 參考 [StartTranscriptionJob](#) 中的。

## Python

### 適用於 Python (Boto3) 的 SDK

#### Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執程式碼範例儲存庫](#)。

```
def start_job(
    job_name,
    media_uri,
    media_format,
    language_code,
    transcribe_client,
    vocabulary_name=None,
):
    """
    Starts a transcription job. This function returns as soon as the job is
    started.
    To get the current status of the job, call get_transcription_job. The job is
    successfully completed when the job status is 'COMPLETED'.

    :param job_name: The name of the transcription job. This must be unique for
        your AWS account.
    :param media_uri: The URI where the audio file is stored. This is typically
        in an Amazon S3 bucket.
    :param media_format: The format of the audio file. For example, mp3 or wav.
    :param language_code: The language code of the audio file.
        For example, en-US or ja-JP
    :param transcribe_client: The Boto3 Transcribe client.
    :param vocabulary_name: The name of a custom vocabulary to use when
    transcribing
        the audio file.
    :return: Data about the job.
    """
    try:
        job_args = {
            "TranscriptionJobName": job_name,
```



```
        "Media": {"MediaFileUri": media_uri},
        "MediaFormat": media_format,
        "LanguageCode": language_code,
    }
    if vocabulary_name is not None:
        job_args["Settings"] = {"VocabularyName": vocabulary_name}
    response = transcribe_client.start_transcription_job(**job_args)
    job = response["TranscriptionJob"]
    logger.info("Started transcription job %s.", job_name)
except ClientError:
    logger.exception("Couldn't start transcription job %s.", job_name)
    raise
else:
    return job
```

- 如需 API 的詳細資訊，請參閱AWS 開發套件[StartTranscriptionJob](#)中的 Python (博托 3) API 參考。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱[搭配 AWS SDK 使用此服務](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

## 搭UpdateVocabulary配 AWS 開發套件或 CLI 使用

下列程式碼範例會示範如何使用UpdateVocabulary。

動作範例是大型程式的程式碼摘錄，必須在內容中執行。您可以在下列程式碼範例的內容中看到此動作：

- [建立和優化自訂詞彙](#)

## .NET

### AWS SDK for .NET

#### Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
/// <summary>
/// Update a custom vocabulary with new values. Update overwrites all
existing information.
/// </summary>
/// <param name="languageCode">The language code of the vocabulary.</param>
/// <param name="phrases">Phrases to use in the vocabulary.</param>
/// <param name="vocabularyName">Name for the vocabulary.</param>
/// <returns>The state of the custom vocabulary.</returns>
public async Task<VocabularyState> UpdateCustomVocabulary(LanguageCode
languageCode,
    List<string> phrases, string vocabularyName)
{
    var response = await _amazonTranscribeService.UpdateVocabularyAsync(
        new UpdateVocabularyRequest()
        {
            LanguageCode = languageCode,
            Phrases = phrases,
            VocabularyName = vocabularyName
        });
    return response.VocabularyState;
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for .NET API 參考[UpdateVocabulary](#)中的。

## CLI

### AWS CLI

使用新用語更新自訂詞彙。

以下 `update-vocabulary` 範例會以您提供的新用語覆寫用來建立自訂詞彙的用語。先決條件：若要取代自訂詞彙中的用語，您需有包含新用語的檔案。

```
aws transcribe update-vocabulary \  
  --vocabulary-file-uri s3://DOC-EXAMPLE-BUCKET/Amazon-S3-Prefix/custom-  
vocabulary.txt \  
  --vocabulary-name custom-vocabulary \  
  --language-code language-code
```

輸出：

```
{  
  "VocabularyName": "custom-vocabulary",  
  "LanguageCode": "language",  
  "VocabularyState": "PENDING"  
}
```

如需詳細資訊，請參閱《Amazon Transcribe 開發人員指南》中的[自訂詞彙](#)。

- 如需 API 詳細資訊，請參閱AWS CLI 命令參考[UpdateVocabulary](#)中的。

## Python

適用於 Python (Boto3) 的 SDK

### Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
def update_vocabulary(  
    vocabulary_name, language_code, transcribe_client, phrases=None,  
    table_uri=None  
):  
    """  
    Updates an existing custom vocabulary. The entire vocabulary is replaced with  
    the contents of the update.  
  
    :param vocabulary_name: The name of the vocabulary to update.
```

```
:param language_code: The language code of the vocabulary.
:param transcribe_client: The Boto3 Transcribe client.
:param phrases: A list of comma-separated phrases to include in the
vocabulary.
:param table_uri: A table of phrases and pronunciation hints to include in
the
                vocabulary.
"""
try:
    vocab_args = {"VocabularyName": vocabulary_name, "LanguageCode":
language_code}
    if phrases is not None:
        vocab_args["Phrases"] = phrases
    elif table_uri is not None:
        vocab_args["VocabularyFileUri"] = table_uri
    response = transcribe_client.update_vocabulary(**vocab_args)
    logger.info("Updated custom vocabulary %s.", response["VocabularyName"])
except ClientError:
    logger.exception("Couldn't update custom vocabulary %s.",
vocabulary_name)
    raise
```

- 如需 API 的詳細資訊，請參閱AWS 開發套件[UpdateVocabulary](#)中的 Python (博托 3) API 參考。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱[搭配 AWS SDK 使用此服務](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

## 使用軟體開發套件 Amazon Transcribe 的案例 AWS

下列程式碼範例說明如何使 AWS 用 SDK 在 Amazon Transcribe 中實作常見案例。這些案例會展示如何在 Amazon Transcribe 中呼叫多個函數來完成特定工作。每個案例都包含一個連結 GitHub，您可以在其中找到如何設定和執程式碼的指示。

### 範例

- [使用開發套件建立和優化 Amazon Transcribe 自訂詞彙 AWS](#)
- [使 Transcribe 開發套件使用 Amazon Transcribe 音訊和取得任務資料 AWS](#)

## 使用開發套件建立和優化 Amazon Transcribe 自訂詞彙 AWS

以下程式碼範例顯示做法：

- 將音訊檔案上傳至 Amazon S3。
- 執行 Amazon Transcribe 工作以轉錄檔案並取得結果。
- 建立和優化自訂詞彙以提高轉錄準確性。
- 使用自訂詞彙執行任務並取得結果。

### Python

適用於 Python (Boto3) 的 SDK

#### Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

轉錄包含 Lewis Carroll 閱讀 Jabberwocky 的音訊檔案。先建立包含 Amazon Transcribe 動作的函數。

```
def start_job(
    job_name,
    media_uri,
    media_format,
    language_code,
    transcribe_client,
    vocabulary_name=None,
):
    """
    Starts a transcription job. This function returns as soon as the job is
    started.
    To get the current status of the job, call get_transcription_job. The job is
    successfully completed when the job status is 'COMPLETED'.

    :param job_name: The name of the transcription job. This must be unique for
        your AWS account.
    :param media_uri: The URI where the audio file is stored. This is typically
        in an Amazon S3 bucket.
```

```
:param media_format: The format of the audio file. For example, mp3 or wav.
:param language_code: The language code of the audio file.
                       For example, en-US or ja-JP
:param transcribe_client: The Boto3 Transcribe client.
:param vocabulary_name: The name of a custom vocabulary to use when
transcribing
                       the audio file.

:return: Data about the job.
"""
try:
    job_args = {
        "TranscriptionJobName": job_name,
        "Media": {"MediaFileUri": media_uri},
        "MediaFormat": media_format,
        "LanguageCode": language_code,
    }
    if vocabulary_name is not None:
        job_args["Settings"] = {"VocabularyName": vocabulary_name}
    response = transcribe_client.start_transcription_job(**job_args)
    job = response["TranscriptionJob"]
    logger.info("Started transcription job %s.", job_name)
except ClientError:
    logger.exception("Couldn't start transcription job %s.", job_name)
    raise
else:
    return job

def get_job(job_name, transcribe_client):
    """
    Gets details about a transcription job.

    :param job_name: The name of the job to retrieve.
    :param transcribe_client: The Boto3 Transcribe client.
    :return: The retrieved transcription job.
    """
    try:
        response = transcribe_client.get_transcription_job(
            TranscriptionJobName=job_name
        )
        job = response["TranscriptionJob"]
        logger.info("Got job %s.", job["TranscriptionJobName"])
    except ClientError:
```

```
        logger.exception("Couldn't get job %s.", job_name)
        raise
    else:
        return job

def delete_job(job_name, transcribe_client):
    """
    Deletes a transcription job. This also deletes the transcript associated with
    the job.

    :param job_name: The name of the job to delete.
    :param transcribe_client: The Boto3 Transcribe client.
    """
    try:
        transcribe_client.delete_transcription_job(TranscriptionJobName=job_name)
        logger.info("Deleted job %s.", job_name)
    except ClientError:
        logger.exception("Couldn't delete job %s.", job_name)
        raise

def create_vocabulary(
    vocabulary_name, language_code, transcribe_client, phrases=None,
    table_uri=None
):
    """
    Creates a custom vocabulary that can be used to improve the accuracy of
    transcription jobs. This function returns as soon as the vocabulary
    processing
    is started. Call get_vocabulary to get the current status of the vocabulary.
    The vocabulary is ready to use when its status is 'READY'.

    :param vocabulary_name: The name of the custom vocabulary.
    :param language_code: The language code of the vocabulary.
        For example, en-US or nl-NL.
    :param transcribe_client: The Boto3 Transcribe client.
    :param phrases: A list of comma-separated phrases to include in the
    vocabulary.
    :param table_uri: A table of phrases and pronunciation hints to include in
    the
        vocabulary.
```

```
:return: Information about the newly created vocabulary.
"""
try:
    vocab_args = {"VocabularyName": vocabulary_name, "LanguageCode":
language_code}
    if phrases is not None:
        vocab_args["Phrases"] = phrases
    elif table_uri is not None:
        vocab_args["VocabularyFileUri"] = table_uri
    response = transcribe_client.create_vocabulary(**vocab_args)
    logger.info("Created custom vocabulary %s.", response["VocabularyName"])
except ClientError:
    logger.exception("Couldn't create custom vocabulary %s.",
vocabulary_name)
    raise
else:
    return response

def get_vocabulary(vocabulary_name, transcribe_client):
    """
    Gets information about a custom vocabulary.

    :param vocabulary_name: The name of the vocabulary to retrieve.
    :param transcribe_client: The Boto3 Transcribe client.
    :return: Information about the vocabulary.
    """
    try:
        response =
transcribe_client.get_vocabulary(VocabularyName=vocabulary_name)
        logger.info("Got vocabulary %s.", response["VocabularyName"])
    except ClientError:
        logger.exception("Couldn't get vocabulary %s.", vocabulary_name)
        raise
    else:
        return response

def update_vocabulary(
    vocabulary_name, language_code, transcribe_client, phrases=None,
    table_uri=None
):
```



```
"""
Updates an existing custom vocabulary. The entire vocabulary is replaced with
the contents of the update.

:param vocabulary_name: The name of the vocabulary to update.
:param language_code: The language code of the vocabulary.
:param transcribe_client: The Boto3 Transcribe client.
:param phrases: A list of comma-separated phrases to include in the
vocabulary.
:param table_uri: A table of phrases and pronunciation hints to include in
the
                vocabulary.
"""
try:
    vocab_args = {"VocabularyName": vocabulary_name, "LanguageCode":
language_code}
    if phrases is not None:
        vocab_args["Phrases"] = phrases
    elif table_uri is not None:
        vocab_args["VocabularyFileUri"] = table_uri
    response = transcribe_client.update_vocabulary(**vocab_args)
    logger.info("Updated custom vocabulary %s.", response["VocabularyName"])
except ClientError:
    logger.exception("Couldn't update custom vocabulary %s.",
vocabulary_name)
    raise

def list_vocabularies(vocabulary_filter, transcribe_client):
    """
    Lists the custom vocabularies created for this AWS account.

    :param vocabulary_filter: The returned vocabularies must contain this string
in
                            their names.
    :param transcribe_client: The Boto3 Transcribe client.
    :return: The list of retrieved vocabularies.
    """
    try:
        response =
transcribe_client.list_vocabularies(NameContains=vocabulary_filter)
        vocabs = response["Vocabularies"]
        next_token = response.get("NextToken")
```

```

while next_token is not None:
    response = transcribe_client.list_vocabularies(
        NameContains=vocabulary_filter, NextToken=next_token
    )
    vocabs += response["Vocabularies"]
    next_token = response.get("NextToken")
logger.info(
    "Got %s vocabularies with filter %s.", len(vocabs), vocabulary_filter
)
except ClientError:
    logger.exception(
        "Couldn't list vocabularies with filter %s.", vocabulary_filter
    )
    raise
else:
    return vocabs

```

```

def delete_vocabulary(vocabulary_name, transcribe_client):
    """
    Deletes a custom vocabulary.

    :param vocabulary_name: The name of the vocabulary to delete.
    :param transcribe_client: The Boto3 Transcribe client.
    """
    try:
        transcribe_client.delete_vocabulary(VocabularyName=vocabulary_name)
        logger.info("Deleted vocabulary %s.", vocabulary_name)
    except ClientError:
        logger.exception("Couldn't delete vocabulary %s.", vocabulary_name)
        raise

```

呼叫包裝函數，在沒有自訂詞彙的情況下轉錄音訊，然後使用不同版本的自訂詞彙來轉錄音訊，以查看改善的結果。

```

def usage_demo():
    """Shows how to use the Amazon Transcribe service."""
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

```

```
s3_resource = boto3.resource("s3")
transcribe_client = boto3.client("transcribe")

print("-" * 88)
print("Welcome to the Amazon Transcribe demo!")
print("-" * 88)

bucket_name = f"jabber-bucket-{time.time_ns()}"
print(f"Creating bucket {bucket_name}.")
bucket = s3_resource.create_bucket(
    Bucket=bucket_name,
    CreateBucketConfiguration={
        "LocationConstraint": transcribe_client.meta.region_name
    },
)
media_file_name = ".media/Jabberwocky.mp3"
media_object_key = "Jabberwocky.mp3"
print(f"Uploading media file {media_file_name}.")
bucket.upload_file(media_file_name, media_object_key)
media_uri = f"s3://{bucket.name}/{media_object_key}"

job_name_simple = f"Jabber-{time.time_ns()}"
print(f"Starting transcription job {job_name_simple}.")
start_job(
    job_name_simple,
    f"s3://{bucket.name}/{media_object_key}",
    "mp3",
    "en-US",
    transcribe_client,
)
transcribe_waiter = TranscribeCompleteWaiter(transcribe_client)
transcribe_waiter.wait(job_name_simple)
job_simple = get_job(job_name_simple, transcribe_client)
transcript_simple = requests.get(
    job_simple["Transcript"]["TranscriptFileUri"]
).json()
print(f"Transcript for job {transcript_simple['jobName']}:")
print(transcript_simple["results"]["transcripts"][0]["transcript"])

print("-" * 88)
print(
    "Creating a custom vocabulary that lists the nonsense words to try to "
    "improve the transcription."
)
```

```
vocabulary_name = f"Jabber-vocabulary-{{time.time_ns()}}"
create_vocabulary(
    vocabulary_name,
    "en-US",
    transcribe_client,
    phrases=[
        "brillig",
        "slithy",
        "borogoves",
        "mome",
        "raths",
        "Jub-Jub",
        "frumious",
        "manxome",
        "Tumtum",
        "uffish",
        "whiffling",
        "tulgey",
        "thou",
        "frabjous",
        "callooh",
        "callay",
        "chortled",
    ],
)
vocabulary_ready_waiter = VocabularyReadyWaiter(transcribe_client)
vocabulary_ready_waiter.wait(vocabulary_name)

job_name_vocabulary_list = f"Jabber-vocabulary-list-{{time.time_ns()}}"
print(f"Starting transcription job {job_name_vocabulary_list}.")
start_job(
    job_name_vocabulary_list,
    media_uri,
    "mp3",
    "en-US",
    transcribe_client,
    vocabulary_name,
)
transcribe_waiter.wait(job_name_vocabulary_list)
job_vocabulary_list = get_job(job_name_vocabulary_list, transcribe_client)
transcript_vocabulary_list = requests.get(
    job_vocabulary_list["Transcript"]["TranscriptFileUri"]
).json()
print(f"Transcript for job {transcript_vocabulary_list['jobName']}:")
```

```

print(transcript_vocabulary_list["results"]["transcripts"][0]["transcript"])

print("-" * 88)
print(
    "Updating the custom vocabulary with table data that provides additional
"
    "pronunciation hints."
)
table_vocab_file = "jabber-vocabulary-table.txt"
bucket.upload_file(table_vocab_file, table_vocab_file)
update_vocabulary(
    vocabulary_name,
    "en-US",
    transcribe_client,
    table_uri=f"s3://{bucket.name}/{table_vocab_file}",
)
vocabulary_ready_waiter.wait(vocabulary_name)

job_name_vocab_table = f"Jabber-vocab-table-{time.time_ns()}"
print(f"Starting transcription job {job_name_vocab_table}.")
start_job(
    job_name_vocab_table,
    media_uri,
    "mp3",
    "en-US",
    transcribe_client,
    vocabulary_name=vocabulary_name,
)
transcribe_waiter.wait(job_name_vocab_table)
job_vocab_table = get_job(job_name_vocab_table, transcribe_client)
transcript_vocab_table = requests.get(
    job_vocab_table["Transcript"]["TranscriptFileUri"]
).json()
print(f"Transcript for job {transcript_vocab_table['jobName']}:")
print(transcript_vocab_table["results"]["transcripts"][0]["transcript"])

print("-" * 88)
print("Getting data for jobs and vocabularies.")
jabber_jobs = list_jobs("Jabber", transcribe_client)
print(f"Found {len(jabber_jobs)} jobs:")
for job_sum in jabber_jobs:
    job = get_job(job_sum["TranscriptionJobName"], transcribe_client)
    print(
        f"\t{job['TranscriptionJobName']}, {job['Media']['MediaFileUri']}, "

```

```
        f"{job['Settings'].get('VocabularyName')}"
    )

    jabber_vocabs = list_vocabularies("Jabber", transcribe_client)
    print(f"Found {len(jabber_vocabs)} vocabularies:")
    for vocab_sum in jabber_vocabs:
        vocab = get_vocabulary(vocab_sum["VocabularyName"], transcribe_client)
        vocab_content = requests.get(vocab["DownloadUri"]).text
        print(f"\t{vocab['VocabularyName']} contents:")
        print(vocab_content)

    print("-" * 88)
    print("Deleting demo jobs.")
    for job_name in [job_name_simple, job_name_vocabulary_list,
                    job_name_vocab_table]:
        delete_job(job_name, transcribe_client)
    print("Deleting demo vocabulary.")
    delete_vocabulary(vocabulary_name, transcribe_client)
    print("Deleting demo bucket.")
    bucket.objects.delete()
    bucket.delete()
    print("Thanks for watching!")
```

- 如需 API 的詳細資訊，請參閱《適用於 Python (Boto3) 的 AWS SDK API 參考資料》中的下列主題。
  - [CreateVocabulary](#)
  - [DeleteTranscriptionJob](#)
  - [DeleteVocabulary](#)
  - [GetTranscriptionJob](#)
  - [GetVocabulary](#)
  - [ListVocabularies](#)
  - [StartTranscriptionJob](#)
  - [UpdateVocabulary](#)

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱[搭配 AWS SDK 使用此服務](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

## 使 Transcribe 開發套件使用 Amazon Transcribe 音訊和取得任務資料 AWS

下列程式碼範例示範如何：

- 使用 Amazon Transcribe 開始轉錄作業。
- 等候 工作完成。
- 取得儲存文字記錄的 URI。

如需詳細資訊，請參閱 [《開始使用 Amazon Transcribe》](#)。

### Java

適用於 Java 2.x 的 SDK

#### Note

還有更多關於 GitHub。尋找完整範例，並了解如何在 [AWS 設定和執行程式碼範例儲存庫](#)。

轉錄 PCM 檔案。

```
/**
 * To run this AWS code example, ensure that you have set up your development
 * environment, including your AWS credentials.
 *
 * For information, see this documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */

public class TranscribeStreamingDemoFile {
    private static final Region REGION = Region.US_EAST_1;
    private static TranscribeStreamingAsyncClient client;

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

        final String USAGE = "\n" +
            "Usage:\n" +
```

```
        "    <file> \n\n" +
        "Where:\n" +
        "    file - the location of a PCM file to transcribe. In this
example, ensure the PCM file is 16 hertz (Hz). \n";

    if (args.length != 1) {
        System.out.println(USAGE);
        System.exit(1);
    }

    String file = args[0];
    client = TranscribeStreamingAsyncClient.builder()
        .region(REGION)
        .build();

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

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

private static InputStream getStreamFromFile(String file) {
    try {
        File inputFile = new File(file);
        InputStream audioStream = new FileInputStream(inputFile);
        return audioStream;

    } catch (FileNotFoundException e) {
        throw new RuntimeException(e);
    }
}

private static StartStreamTranscriptionRequest getRequest(Integer
mediaSampleRateHertz) {
    return StartStreamTranscriptionRequest.builder()
        .languageCode(LanguageCode.EN_US)
        .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 static class AudioStreamPublisher implements Publisher<AudioStream> {
    private final InputStream inputStream;
    private static Subscription currentSubscription;

    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);

    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();  
}  
}
```

轉錄來自電腦麥克風的串流音訊。

```
public class TranscribeStreamingDemoApp {  
    private static final Region REGION = Region.US_EAST_1;  
    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 static Subscription currentSubscription;

    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);

    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();
    }
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Java 2.x API 參考》中的下列主題。
  - [GetTranscriptionJob](#)
  - [StartTranscriptionJob](#)

## Python

適用於 Python (Boto3) 的 SDK

### Note

還有更多關於 GitHub。尋找完整範例，並了解如何在[AWS 設定和執行程式碼範例儲存庫](#)。

```
import time
import boto3

def transcribe_file(job_name, file_uri, transcribe_client):
    transcribe_client.start_transcription_job(
        TranscriptionJobName=job_name,
        Media={"MediaFileUri": file_uri},
        MediaFormat="wav",
        LanguageCode="en-US",
    )

    max_tries = 60
    while max_tries > 0:
```





## 範例

- [建置 Amazon Transcribe 應用程式](#)
- [建置 Amazon Transcribe 串流應用程式](#)
- [使用 AWS SDK 將文本轉換為語音並返回文本](#)

## 建置 Amazon Transcribe 應用程式

下列程式碼範例示範如何使用 Amazon Transcribe 在瀏覽器中轉錄和顯示語音錄音。

### JavaScript

適用於 JavaScript (v3) 的開發套件

建立使用 Amazon Transcribe 的應用程式，在瀏覽器中轉錄和顯示語音錄音。應用程式使用兩個 Amazon Simple Storage Service (Amazon S3) 儲存貯體，一個負責支援應用程式程式碼，另一個負責存放文字記錄。應用程式會使用 Amazon Cognito 使用者集區來對您的使用者進行身分驗證。經過驗證的使用者具有 AWS Identity and Access Management (IAM) 許可以存取所需 AWS 服務。

有關如何設置和運行的完整源代碼和說明，請參閱中的完整示例[GitHub](#)。

此範例也可在 [AWS SDK for JavaScript v3 開發人員指南](#)中取得。

此範例中使用的服務

- Amazon Cognito Identity
- Amazon S3
- Amazon Transcribe

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱[搭配 AWS SDK 使用此服務](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

## 建置 Amazon Transcribe 串流應用程式

下面的程式碼範例說明如何建置可即時記錄、轉錄和翻譯直播音訊並透過電子郵件傳送結果的應用程式。

## JavaScript

### 適用於 JavaScript (v3) 的開發套件

說明如何使用 Amazon Transcribe 建置應用程式，該應用程式可即時記錄、轉錄和翻譯直播音訊，並可使用 Amazon Simple Email Service (Amazon SES) 透過電子郵件傳送結果。

有關如何設置和運行的完整源代碼和說明，請參閱中的完整示例[GitHub](#)。

此範例中使用的服務

- Amazon Comprehend
- Amazon SES
- Amazon Transcribe
- Amazon Translate

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱[搭配 AWS SDK 使用此服務](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

## 使用 AWS SDK 將文本轉換為語音並返回文本

以下程式碼範例顯示做法：

- 使用 Amazon Polly 將純文字 (UTF-8) 輸入檔案合成至音訊檔案中。
- 將音訊檔案上傳至 Amazon S3 儲存貯體。
- 使用 Amazon Transcribe 將音訊檔案轉換為文字。
- 顯示文字。

## Rust

### 適用於 Rust 的 SDK

使用 Amazon Polly 將純文字 (UTF-8) 輸入檔案合成至音訊檔案中，將音訊檔案上傳至 Amazon S3 儲存貯體，使用 Amazon Transcribe 將該音訊檔案轉換為文字，然後顯示文字。

有關如何設置和運行的完整源代碼和說明，請參閱中的完整示例[GitHub](#)。

此範例中使用的服務

- Amazon Polly

- Amazon S3
- Amazon Transcribe

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱[搭配 AWS SDK 使用此服務](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

# Amazon Transcribe 中的安全性

雲端安全是 AWS 最重視的一環。身為 AWS 的客戶，您將能從資料中心和網路架構中獲益，這些都是專為最重視安全的組織而設計的。

安全是 AWS 與您共同肩負的責任。[共同責任模型](#)將其描述為雲端的安全性和雲端中的安全性：

- 雲端本身的安全：AWS負責保護AWS在AWS 雲端。AWS也會提供您可以安全使用的服務。第三方稽核人員會定期測試和驗證我們安全性的有效性，作為 [AWS 合規計劃](#)的一部分。若要了解適用於 Amazon Transcribe 的合規計劃，請參閱[合規計劃的 AWS 服務範圍](#)。
- 雲端內部的安全：您的責任取決於所使用的AWS服務。您也必須對其他因素負責，包括資料的機密性、您的公司的要求和適用法律和法規。

本文件有助於您了解如何在使用 Amazon Transcribe 時套用共同責任模型。下列主題說明如何將 Amazon Transcribe 設定為達到您的安全及合規目標。您也將了解如何使用其他AWS服務來監控並保護Amazon Transcribe源。

## 主題

- [的 Identity and Access Management Amazon Transcribe](#)
- [Amazon Transcribe 中的資料保護](#)
- [監控 Amazon Transcribe](#)
- [符合性驗證 Amazon Transcribe](#)
- [Amazon Transcribe 中的恢復能力](#)
- [Amazon Transcribe 中的基礎設施安全](#)
- [Amazon Transcribe 中的漏洞分析和](#)管理
- [Amazon Transcribe 的安全最佳實務](#)

## 的 Identity and Access Management Amazon Transcribe

AWS Identity and Access Management (IAM) 可協助系統管理員安全地控制 AWS 資源存取權。AWS 服務 IAM 管理員控制哪些人可以通過身份驗證 (登入) 和授權 (具有權限) 來使用 Amazon Transcribe 資源。IAM 是您可以使用的 AWS 服務，無需額外付費。

## 主題

- [物件](#)
- [使用身分驗證](#)
- [使用政策管理存取權](#)
- [如何與 IAM Amazon Transcribe 搭配使用](#)
- [預防跨服務混淆代理人](#)
- [Amazon Transcribe 身分型政策範例](#)
- [疑難排解 Amazon Transcribe 身分和存取](#)

## 物件

您使用 AWS Identity and Access Management (IAM) 的方式會有所不同，具體取決於您在進行的工作 Amazon Transcribe。

**服務使用者** — 如果您使用 Amazon Transcribe 服務執行工作，則管理員會為您提供所需的認證和權限。當您使用更多 Amazon Transcribe 功能來完成工作時，您可能需要其他權限。了解存取的管理方式可協助您向管理員請求正確的許可。若您無法存取 Amazon Transcribe 中的某項功能，請參閱 [疑難排解 Amazon Transcribe 身分和存取](#)。

**服務管理員** — 如果您負責公司的 Amazon Transcribe 資源，您可能擁有完整的存取權 Amazon Transcribe。決定您的服務使用者應該存取哪些 Amazon Transcribe 功能和資源是您的工作。接著，您必須將請求提交給您的 IAM 管理員，來變更您服務使用者的許可。檢閱此頁面上的資訊，了解 IAM 的基本概念。若要進一步瞭解貴公司如何搭配使用 IAM Amazon Transcribe，請參閱 [如何與 IAM Amazon Transcribe 搭配使用](#)。

**IAM 管理員**：如果您是 IAM 管理員，建議您掌握如何撰寫政策以管理 Amazon Transcribe 存取權的詳細資訊。若要檢視可在 IAM 中使用的 Amazon Transcribe 基於身分的政策範例，請參閱 [Amazon Transcribe 身分型政策範例](#)

## 使用身分驗證

驗證是您 AWS 使用身分認證登入的方式。您必須以 IAM 使用者身分或假設 IAM 角色進行驗證 (登入 AWS)。AWS 帳戶根使用者

您可以使用透過 AWS 身分識別來源提供的認證，以聯合身分識別身分登入。AWS IAM Identity Center (IAM 身分中心) 使用者、貴公司的單一登入身分驗證，以及您的 Google 或 Facebook 登入資料都是聯合身分識別的範例。您以聯合身分登入時，您的管理員先前已設定使用 IAM 角色的聯合身分。當您使 AWS 用同盟存取時，您會間接擔任角色。

根據您的使用者類型，您可以登入 AWS Management Console 或 AWS 存取入口網站。如需有關登入的詳細資訊 AWS，請參閱《AWS 登入 使用指南》AWS 帳戶中的[如何登入](#)您的。

如果您 AWS 以程式設計方式存取，請 AWS 提供軟體開發套件 (SDK) 和命令列介面 (CLI)，以使用您的認證以加密方式簽署要求。如果您不使用 AWS 工具，則必須自行簽署要求。如需使用建議的方法自行簽署請求的詳細資訊，請參閱 IAM 使用者指南中的[簽署 AWS API 請求](#)。

無論您使用何種身分驗證方法，您可能都需要提供額外的安全性資訊。例如，AWS 建議您使用多重要素驗證 (MFA) 來增加帳戶的安全性。如需更多資訊，請參閱 AWS IAM Identity Center 使用者指南中的[多重要素驗證](#)和 IAM 使用者指南中的[在 AWS 中使用多重要素驗證 \(MFA\)](#)。

## AWS 帳戶 根使用者

當您建立時 AWS 帳戶，您會從一個登入身分開始，該身分可完整存取該帳戶中的所有資源 AWS 服務和資源。此身分稱為 AWS 帳戶 root 使用者，可透過使用您用來建立帳戶的電子郵件地址和密碼登入來存取。強烈建議您不要以根使用者處理日常任務。保護您的根使用者憑證，並將其用來執行只能由根使用者執行的任務。如需這些任務的完整清單，了解需以根使用者登入的任務，請參閱 IAM 使用者指南中的[需要根使用者憑證的任務](#)。

## 聯合身分

最佳作法是要求人類使用者 (包括需要系統管理員存取權的使用者) 使用與身分識別提供者的同盟，才能使用臨時登入資料進行存取 AWS 服務。

聯合身分識別是來自企業使用者目錄的使用者、Web 身分識別提供者、Identity Center 目錄，或使用透過身分識別來源提供的認證進行存取 AWS 服務的任何使用者。AWS Directory Service 同盟身分存取時 AWS 帳戶，他們會假設角色，而角色則提供臨時認證。

對於集中式存取權管理，我們建議您使用 AWS IAM Identity Center。您可以在 IAM Identity Center 中建立使用者和群組，也可以連線並同步到自己身分識別來源中的一組使用者和群組，以便在所有應用程式 AWS 帳戶 和應用程式中使用。如需 IAM Identity Center 的詳細資訊，請參閱 AWS IAM Identity Center 使用者指南中的[什麼是 IAM Identity Center ?](#)。

## IAM 使用者和群組

[IAM 使用者](#)是您內部的身分，具 AWS 帳戶 有單一人員或應用程式的特定許可。建議您盡可能依賴暫時憑證，而不是擁有建立長期憑證 (例如密碼和存取金鑰) 的 IAM 使用者。但是如果特定使用案例需要擁有長期憑證的 IAM 使用者，建議您輪換存取金鑰。如需更多資訊，請參閱 [IAM 使用者指南](#)中的為需要長期憑證的使用案例定期輪換存取金鑰。



[IAM 群組](#)是一種指定 IAM 使用者集合的身分。您無法以群組身分簽署。您可以使用群組來一次為多名使用者指定許可。群組可讓管理大量使用者許可的程序變得更為容易。例如，您可以擁有一個名為 IAMAdmins 的群組，並給予該群組管理 IAM 資源的許可。

使用者與角色不同。使用者只會與單一人員或應用程式建立關聯，但角色的目的是在由任何需要它的人員取得。使用者擁有永久的長期憑證，但角色僅提供暫時憑證。如需進一步了解，請參閱IAM 使用者指南中的[建立 IAM 使用者 \(而非角色\) 的時機](#)。

## IAM 角色

[IAM 角色](#)是您 AWS 帳戶 內部具有特定許可的身分。它類似 IAM 使用者，但不與特定的人員相關聯。您可以[切換角色，在中暫時擔任 IAM 角色](#)。AWS Management Console 您可以透過呼叫 AWS CLI 或 AWS API 作業或使用自訂 URL 來擔任角色。如需使用角色的方法詳細資訊，請參閱 IAM 使用者指南中的[使用 IAM 角色](#)。

使用暫時憑證的 IAM 角色在下列情況中非常有用：

- 聯合身分使用者存取 — 如需向聯合身分指派許可，請建立角色，並為角色定義許可。當聯合身分進行身分驗證時，該身分會與角色建立關聯，並獲授予由角色定義的許可。如需有關聯合角色的相關資訊，請參閱 [IAM 使用者指南](#)中的為第三方身分提供者建立角色。如果您使用 IAM Identity Center，則需要設定許可集。為控制身分驗證後可以存取的內容，IAM Identity Center 將許可集與 IAM 中的角色相關聯。如需有關許可集的資訊，請參閱 AWS IAM Identity Center 使用者指南中的[許可集](#)。
- 暫時 IAM 使用者許可 – IAM 使用者或角色可以擔任 IAM 角色來暫時針對特定任務採用不同的許可。
- 跨帳戶存取權：您可以使用 IAM 角色，允許不同帳戶中的某人 (信任的主體) 存取您帳戶的資源。角色是授予跨帳戶存取權的主要方式。但是，對於某些策略 AWS 服務，您可以將策略直接附加到資源 (而不是使用角色作為代理)。若要了解跨帳戶存取角色和以資源為基礎的政策之間的差異，請參閱 IAM 使用者指南中的[IAM 中的跨帳戶資源存取](#)。
- 跨服務訪問 — 有些 AWS 服務 使用其他 AWS 服務功能。例如，當您在服務中進行呼叫時，該服務通常會在 Amazon EC2 中執行應用程式或將物件儲存在 Amazon Simple Storage Service (Amazon S3) 中。服務可能會使用呼叫主體的許可、使用服務角色或使用服務連結角色來執行此作業。
- 轉寄存取工作階段 (FAS) — 當您使用 IAM 使用者或角色在中執行動作時 AWS，您會被視為主體。使用某些服務時，您可能會執行某個動作，進而在不同服務中啟動另一個動作。FAS 會使用主體呼叫的權限 AWS 服務，並結合要求 AWS 服務 向下游服務發出要求。只有當服務收到需要與其 AWS 服務 他資源互動才能完成的請求時，才會發出 FAS 請求。在此情況下，您必須具有執行這兩個動作的許可。如需提出 FAS 請求時的策略詳細資訊，請參閱 [《轉發存取工作階段》](#)。
- 服務角色 – 服務角色是服務擔任的 [IAM 角色](#)，可代表您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需詳細資訊，請參閱 IAM 使用者指南中的[建立角色以委派許可給 AWS 服務服務](#)。



- 服務連結角色 — 服務連結角色是連結至 AWS 服務服務可以擔任代表您執行動作的角色。服務連結角色會顯示在您的中，AWS 帳戶且屬於服務所有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。
- 在 Amazon EC2 上執行的應用程式 — 您可以使用 IAM 角色來管理在 EC2 執行個體上執行的應用程式以及發出 AWS CLI 或 AWS API 請求的臨時登入資料。這是在 EC2 執行個體內儲存存取金鑰的較好方式。若要將 AWS 角色指派給 EC2 執行個體並提供給其所有應用程式，請建立連接至執行個體的執行個體設定檔。執行個體設定檔包含該角色，並且可讓 EC2 執行個體上執行的程式取得暫時憑證。如需詳細資訊，請參閱 IAM 使用者指南中的[利用 IAM 角色來授予許可給 Amazon EC2 執行個體上執行的應用程式](#)。

如需了解是否要使用 IAM 角色或 IAM 使用者，請參閱 IAM 使用者指南中的[建立 IAM 角色 \(而非使用者\) 的時機](#)。

## 使用政策管理存取權

您可以透過 AWS 過建立原則並將其附加至 AWS 身分識別或資源來控制中的存取。原則是一個物件 AWS，當與身分識別或資源相關聯時，會定義其權限。AWS 當主參與者 (使用者、root 使用者或角色工作階段) 提出要求時，評估這些原則。政策中的許可決定是否允許或拒絕請求。大多數原則會 AWS 以 JSON 文件的形式儲存在中。如需 JSON 政策文件結構和內容的詳細資訊，請參閱 IAM 使用者指南中的[JSON 政策概觀](#)。

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

預設情況下，使用者和角色沒有許可。若要授予使用者對其所需資源執行動作的許可，IAM 管理員可以建立 IAM 政策。然後，管理員可以將 IAM 政策新增至角色，使用者便能擔任這些角色。

IAM 政策定義該動作的許可，無論您使用何種方法來執行操作。例如，假設您有一個允許 `iam:GetRole` 動作的政策。具有該原則的使用者可以從 AWS Management Console AWS CLI、或 AWS API 取得角色資訊。

### 身分型政策

身分型政策是可以附加到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。若要了解如何建立身分類型政策，請參閱 IAM 使用者指南中的[建立 IAM 政策](#)。

身分型政策可進一步分類成內嵌政策或受管政策。內嵌政策會直接內嵌到單一使用者、群組或角色。受管理的策略是獨立策略，您可以將其附加到您的 AWS 帳戶。受管政策包括 AWS 受管政策和客戶管理

的策略。如需了解如何在受管政策及內嵌政策間選擇，請參閱 IAM 使用者指南中的[在受管政策和內嵌政策間選擇](#)。

## 資源型政策

資源型政策是連接到資源的 JSON 政策文件。資源型政策的最常見範例是 IAM 角色信任政策和 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權限。對於附加政策的資源，政策會定義指定的主體可以對該資源執行的動作以及在何種條件下執行的動作。您必須在資源型政策中[指定主體](#)。主參與者可以包括帳戶、使用者、角色、同盟使用者或。AWS 服務

資源型政策是位於該服務中的內嵌政策。您無法在以資源為基礎的政策中使用 IAM 的 AWS 受管政策。

## 存取控制清單 (ACL)

存取控制清單 (ACL) 可控制哪些主體 (帳戶成員、使用者或角色) 擁有存取某資源的許可。ACL 類似於資源型政策，但它們不使用 JSON 政策文件格式。

Amazon S3 和 Amazon VPC 是支援 ACL 的服務範例。AWS WAF 如需進一步了解 ACL，請參閱 Amazon Simple Storage Service 開發人員指南中的[存取控制清單 \(ACL\) 概觀](#)。

## 其他政策類型

AWS 支援其他較不常見的原則類型。這些政策類型可設定較常見政策類型授予您的最大許可。

- 許可界限 – 許可範圍是一種進階功能，可供您設定身分型政策能授予 IAM 實體 (IAM 使用者或角色) 的最大許可。您可以為實體設定許可界限。所產生的許可會是實體的身分型政策和其許可界限的交集。會在 Principal 欄位中指定使用者或角色的資源型政策則不會受到許可界限限制。所有這類政策中的明確拒絕都會覆寫該允許。如需許可界限的詳細資訊，請參閱 IAM 使用者指南中的[IAM 實體許可界限](#)。
- 服務控制策略 (SCP) — SCP 是 JSON 策略，用於指定中組織或組織單位 (OU) 的最大權限。AWS Organizations 是一種用於分組和集中管理您企業擁有的多個 AWS 帳戶的服務。若您啟用組織中的所有功能，您可以將服務控制政策 (SCP) 套用到任何或所有帳戶。SCP 限制成員帳戶中實體的權限，包括每個 AWS 帳戶根使用者帳戶。如需 Organizations 和 SCP 的詳細資訊，請參閱 AWS Organizations 使用者指南中的[SCP 運作方式](#)。
- 工作階段政策 – 工作階段政策是一種進階政策，您可以在透過編寫程式的方式建立角色或聯合使用者的暫時工作階段時，作為參數傳遞。所產生工作階段的許可會是使用者或角色的身分型政策和工作

階段政策的交集。許可也可以來自資源型政策。所有這類政策中的明確拒絕都會覆寫該允許。如需詳細資訊，請參閱 IAM 使用者指南中的 [工作階段政策](#)。

## 多種政策類型

將多種政策類型套用到請求時，其結果形成的許可會更為複雜、更加難以理解。要了解如何在涉及多個政策類型時 AWS 確定是否允許請求，請參閱《IAM 使用者指南》中的 [政策評估邏輯](#)。

## 如何與 IAM Amazon Transcribe 搭配使用

在您使用 IAM 管理存取權限之前 Amazon Transcribe，請先了解哪些 IAM 功能可搭配使用 Amazon Transcribe。

IAM 您可以搭配使用的功能 Amazon Transcribe

IAM 特徵	Amazon Transcribe 支持
<a href="#">身分型政策</a>	是
<a href="#">資源型政策</a>	否
<a href="#">政策動作</a>	是
<a href="#">政策資源</a>	是
<a href="#">政策條件索引鍵 (服務特定)</a>	是
<a href="#">ACL</a>	否
<a href="#">ABAC(政策中的標籤)</a>	部分
<a href="#">臨時憑證</a>	是
<a href="#">主體許可</a>	是
<a href="#">服務角色</a>	是
<a href="#">服務連結角色</a>	否

若要深入瞭解其他 AWS 服務如何 Amazon Transcribe 與大部分 IAM 功能搭配使用，請參閱 IAM 使用者指南 IAM 中的使用 AWS [服務](#)。

## 適用於 Amazon Transcribe 的身分型政策

支援身分型政策 是

身分型政策是可以連接到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。若要了解如何建立身分類型政策，請參閱《IAM 使用者指南》中的 [建立 IAM 政策](#)。

使用 IAM 身分型政策，您可以指定允許或拒絕的動作和資源，以及在何種條件下允許或拒絕動作。您無法在身分型政策中指定主體，因為這會套用至連接的使用者或角色。如要了解您在 JSON 政策中使用的所有元素，請參閱《IAM 使用者指南》中的 [IAM JSON 政策元素參考](#)。

以身分識別為基礎的原則範例 Amazon Transcribe

若要檢視以 Amazon Transcribe 身為基礎的原則範例，請參閱 [Amazon Transcribe 身分型政策範例](#)

## Amazon Transcribe 內的資源型政策

支援以資源基礎的政策 否

資源型政策是附加到資源的 JSON 政策文件。資源型政策的最常見範例是 IAM 角色信任政策和 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權限。對於附加政策的資源，政策會定義指定的主體可以對該資源執行的動作以及在何種條件下執行的動作。您必須在資源型政策中 [指定主體](#)。主參與者可以包括帳戶、使用者、角色、同盟使用者或。AWS 服務

如需啟用跨帳戶存取權，您可以指定在其他帳戶內的所有帳戶或 IAM 實體，作為資源型政策的主體。新增跨帳戶主體至資源型政策，只是建立信任關係的一半。當主體和資源位於不同時 AWS 帳戶，受信任帳戶中的 IAM 管理員也必須授與主體實體 (使用者或角色) 權限，才能存取資源。其透過將身分型政策連接到實體來授與許可。不過，如果資源型政策會為相同帳戶中的主體授予存取，這時就不需要額外的身分型政策。如需詳細資訊，請參閱 IAM 使用者指南 [中的 IAM 中的跨帳戶資源存取](#)。

## 適用於 Amazon Transcribe 的政策動作

支援政策動作 是

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

JSON 政策的 Action 元素描述您可以用來允許或拒絕政策中存取的動作。原則動作通常與關聯的 AWS API 作業具有相同的名稱。有一些例外狀況，例如沒有相符的 API 操作的僅限許可動作。也有一些作業需要政策中的多個動作。這些額外的動作稱為相依動作。

政策會使用動作來授予執行相關聯動作的許可。

若要查看 Amazon Transcribe 動作清單，請參閱服務授權參考 Amazon Transcribe 中 [所定義的動作](#)。

正在 Amazon Transcribe 使用的策略動作在動作之前使用前transcribe綴。如需在單一陳述式中指定多個動作，請用逗號分隔。

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

您也可以使用萬用字元 (\*) 來指定多個動作。例如，若要指定開頭是 List 文字的所有動作，請包含以下動作：

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

若要檢視以 Amazon Transcribe 身為基礎的原則範例，請參閱 [Amazon Transcribe 身分型政策範例](#)

## Amazon Transcribe 的政策資源

支援政策資源 是

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

Resource JSON 政策元素可指定要套用動作的物件。陳述式必須包含 Resource 或 NotResource 元素。最佳實務是使用其 [Amazon Resource Name \(ARN\)](#) 來指定資源。您可以針對支援特定資源類型的動作 (稱為資源層級許可) 來這麼做。

對於不支援資源層級許可的動作 (例如列出操作)，請使用萬用字元 (\*) 來表示陳述式適用於所有資源。

```
"Resource": "*"
```

若要查看 Amazon Transcribe 資源類型及其 ARN 的清單，請參閱服務授權參考 Amazon Transcribe 中 [所定義的資源](#)。若要了解您可以使用哪些動作指定每個資源的 ARN，請參閱 [Amazon Transcribe 定義的動作](#)。

若要檢視以 Amazon Transcribe 身分為基礎的原則範例，請參閱 [Amazon Transcribe 身分型政策範例](#)

## 的政策條件索引鍵 Amazon Transcribe

支援服務特定政策條件金鑰 是

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

Condition 元素 (或 Condition 區塊) 可讓您指定使陳述式生效的條件。Condition 元素是選用項目。您可以建立使用 [條件運算子](#) 的條件運算式 (例如等於或小於)，來比對政策中的條件和請求中的值。

若您在陳述式中指定多個 Condition 元素，或是在單一 Condition 元素中指定多個索引鍵，AWS 會使用邏輯 AND 操作評估他們。如果您為單一條件索引鍵指定多個值，請使用邏輯 OR 運算來 AWS 評估條件。必須符合所有條件，才會授與陳述式的許可。

您也可以指定條件時使用預留位置變數。例如，您可以只在使用者使用其 IAM 使用者名稱標記時，將存取資源的許可授予該 IAM 使用者。如需更多資訊，請參閱 IAM 使用者指南中的 [IAM 政策元素：變數和標籤](#)。

AWS 支援全域條件金鑰和服務特定條件金鑰。若要查看所有 AWS 全域條件金鑰，請參閱《IAM 使用者指南》中的 [AWS 全域條件內容金鑰](#)。



若要查看 Amazon Transcribe 條件索引鍵清單，請參閱服務授權參考 Amazon Transcribe 中的 [條件金鑰](#)。若要瞭解您可以使用條件索引鍵的動作和資源，請參閱 [定義的動作 Amazon Transcribe](#)。

若要檢視以 Amazon Transcribe 身分為基礎的原則範例，請參閱 [Amazon Transcribe 身分型政策範例](#)

## ACL 在 Amazon Transcribe

支援 ACL	否
--------	---

存取控制清單 (ACL) 可控制哪些主體 (帳戶成員、使用者或角色) 擁有存取某資源的許可。ACL 類似於資源型政策，但它們不使用 JSON 政策文件格式。

## 阿巴克與 Amazon Transcribe

支援 ABAC (政策中的標籤)	部分
------------------	----

屬性型存取控制 (ABAC) 是一種授權策略，可根據屬性來定義許可。在中 AWS，這些屬性稱為標籤。您可以將標籤附加到 IAM 實體 (使用者或角色) 和許多 AWS 資源。為實體和資源加上標籤是 ABAC 的第一步。您接著要設計 ABAC 政策，允許在主體的標籤與其嘗試存取的資源標籤相符時操作。

ABAC 在成長快速的環境中相當有幫助，並能在政策管理變得繁瑣時提供協助。

如需根據標籤控制存取，請使用 `aws:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 條件索引鍵，在政策的 [條件元素](#) 中，提供標籤資訊。

如果服務支援每個資源類型的全部三個條件金鑰，則對該服務而言，值為 Yes。如果服務僅支援某些資源類型的全部三個條件金鑰，則值為 Partial。

如需 ABAC 的詳細資訊，請參閱 IAM 使用者指南中的 [什麼是 ABAC?](#)。如要查看含有設定 ABAC 步驟的教學課程，請參閱 IAM 使用者指南中的 [使用屬性型存取控制 \(ABAC\)](#)。

如需標記 Amazon Transcribe 資源的更多資訊，請參閱 [標記 資源](#)。如需標籤型存取控制的詳細資訊，請參閱 [使用標籤控制 AWS 資源的存取](#)。

## 將臨時憑證與 Amazon Transcribe 搭配使用

支援臨時憑證	是
--------	---

當您使用臨時憑據登錄時，某些 AWS 服務 不起作用。如需其他資訊，包括哪些 AWS 服務 與臨時登入資料 [搭配 AWS 服務 使用](#)，請參閱 IAM 使用者指南中的 IAM。

如果您使用除了使用者名稱和密碼以外的任何方法登入，則您正在 AWS Management Console 使用臨時認證。例如，當您 AWS 使用公司的單一登入 (SSO) 連結存取時，該程序會自動建立暫時認證。當您以使用者身分登入主控台，然後切換角色時，也會自動建立臨時憑證。如需切換角色的詳細資訊，請參閱 IAM 使用者指南中的 [切換至角色 \(主控台\)](#)。

您可以使用 AWS CLI 或 AWS API 手動建立臨時登入資料。然後，您可以使用這些臨時登入資料來存取 AWS。AWS 建議您動態產生臨時登入資料，而不是使用長期存取金鑰。如需詳細資訊，請參閱 [IAM 中的暫時性安全憑證](#)。

## Amazon Transcribe 的跨服務主體權限

支援轉寄存取工作階段 (FAS) 是

當您使用 IAM 使用者或角色在中執行動作時 AWS，您會被視為主體。使用某些服務時，您可能會執行某個動作，進而在不同服務中啟動另一個動作。FAS 會使用主體呼叫的權限 AWS 服務，並結合要求 AWS 服務 向下游服務發出要求。只有當服務收到需要與其 AWS 服務 他資源互動才能完成的請求時，才會發出 FAS 請求。在此情況下，您必須具有執行這兩個動作的許可。如需提出 FAS 請求時的政策詳細資訊，請參閱 [《轉發存取工作階段》](#)。

## 的服務角色 Amazon Transcribe

支援服務角色 是

服務角色是服務擔任的 [IAM 角色](#)，可代您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需詳細資訊，請參閱 IAM 使用者指南中的 [建立角色以委派許可給 AWS 服務服務](#)。

### Warning

變更服務角色的權限可能會中斷 Amazon Transcribe 功能。只有在 Amazon Transcribe 提供指引時才編輯服務角色。



## Amazon Transcribe的服務連結角色

支援服務連結角色。

否

服務連結角色是一種連結至 AWS 服務服務可以擔任代表您執行動作的角色。服務連結角色會顯示在您的中，AWS 帳戶 且屬於服務所有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。

Amazon Transcribe 不支援服務連結角色。

如需建立或管理其他服務之服務連結角色的詳細資訊，請參閱[使用 IAM的AWS 服務](#)。在表格中尋找服務，其中包含服務連結角色欄中的 Yes。選擇是連結，以檢視該服務的服務連結角色文件。

## 預防跨服務混淆代理人

混淆的副手是由不同實體強制執行動作的實體（服務或帳戶）。這種類型的模擬可能發生跨帳戶和跨服務。

為了預防混代理人，AWS提供的工具可協助您保護所有服務的資料，而這些服務主體已獲得您的資料AWS 帳戶。本節重點介紹特定於跨服務混淆的副預防Amazon Transcribe；但是，您可以在IAM用戶指南的[混淆副問題](#)部分中了解有關此主題的更多信息。

若要限制存取資源Amazon Transcribe的權IAM限，建議您[aws:SourceAccount](#)在資源策略中使用全域條件內容索引鍵[aws:SourceArn](#)。

如果同時使用這兩個全域條件內容索引鍵，且aws:SourceArn值包含AWS 帳戶 ID，則AWS 帳戶在相同政策陳述式中使用時，aws:SourceArn必須使用相同的AWS 帳戶 ID。aws:SourceAccount

如果您想要僅允許一個資源與跨服務存取相關聯，請使用 aws:SourceArn。如果您想要將其中的任何資源AWS 帳戶與跨服務存取相關聯，請使用aws:SourceAccount。

### Note

防混混代理人問題最有效的方法是使用條件內容金鑰，以及其中包含資源的完整 ARN。aws:SourceArn如果不知道完整 ARN，或者如果您指定了多個資源，請使用這些aws:SourceArn全域條件內容金鑰，同時使用萬用字元 (\*) 表示 ARN 的未知部分。例如：arn:aws:transcribe::123456789012:\*

如需假設角色原則的範例，其中顯示如何避免混淆的副問題，請參閱[預防政策混淆代理人](#)。

## Amazon Transcribe 身分型政策範例

根據預設，使用者和角色不具備建立或修改 Amazon Transcribe 資源的權限。他們也無法使用 AWS Management Console、AWS Command Line Interface (AWS CLI) 或 AWS API 執行任務。若要授與使用者對其所需資源執行動作的許可，IAM 管理員可以建立 IAM 政策建立 IAM 政策。然後，管理員可以將 IAM 政策新增至角色，使用者可以擔任這些角色。

若要了解如何使用這些範例 JSON 政策文件建立 IAM 身分型政策，請參閱《IAM 使用者指南》中的[建立 IAM 政策](#)。

如需 Amazon Transcribe 所定義之動作和資源類型的詳細資訊，包括每種資源類型的 ARN 格式，請參閱《服務授權參考》中 [Amazon Transcribe 適用的動作、資源和條件金鑰](#)。

### 主題

- [政策最佳實務](#)
- [使用 AWS Management Console](#)
- [IAM 角色所需的許可](#)
- [Amazon S3 加密金鑰所需的權限](#)
- [允許使用者檢視他們自己的許可](#)
- [AWS KMS 加密內容政策](#)
- [預防政策混淆代理人](#)
- [根據標籤檢視轉錄任務](#)

### 政策最佳實務

身分型政策會判斷您帳戶中的某個人員是否可以建立、存取或刪除 Amazon Transcribe 資源。這些動作可能會讓您的 AWS 帳戶產生費用。當您建立或編輯身分型政策時，請遵循下列準則及建議事項：

- 開始使用 AWS 受管政策並朝向最低權限許可的目標邁進 – 若要開始授予許可給使用者和工作負載，請使用 AWS 受管政策，這些政策會授予許可給許多常用案例。它們可在您的 AWS 帳戶中使用。我們建議您定義特定於使用案例的 AWS 客戶管理政策，以便進一步減少許可。如需詳細資訊，請參閱 IAM 使用者指南中的 [AWS 受管政策](#) 或 [任務職能的 AWS 受管政策](#)。
- 套用最低權限許可 – 設定 IAM 政策的許可時，請僅授予執行任務所需的許可。為實現此目的，您可以定義在特定條件下可以對特定資源採取的動作，這也稱為最低權限許可。如需使用 IAM 套用許可的詳細資訊，請參閱 IAM 使用者指南中的 [IAM 中的政策和許可](#)。

- 使用 IAM 政策中的條件進一步限制存取權 – 您可以將條件新增至政策，以限制動作和資源的存取。例如，您可以撰寫政策條件，指定必須使用 SSL 傳送所有請求。您也可以使用條件來授予對服務動作的存取權，前提是透過特定 AWS 服務 (例如 AWS CloudFormation) 使用條件。如需詳細資訊，請參閱《IAM 使用者指南》中的 [IAM JSON 政策元素：條件](#)。
- 使用 IAM Access Analyzer 驗證 IAM 政策，確保許可安全且可正常運作 – IAM Access Analyzer 驗證新政策和現有政策，確保這些政策遵從 IAM 政策語言 (JSON) 和 IAM 最佳實務。IAM Access Analyzer 提供 100 多項政策檢查及切實可行的建議，可協助您編寫安全且實用的政策。如需詳細資訊，請參閱 IAM 使用者指南中的 [IAM Access Analyzer 政策驗證](#)。
- 需要多重要素驗證 (MFA) – 如果存在需要 AWS 帳戶中 IAM 使用者或根使用者的情況，請開啟 MFA 提供額外的安全性。若要在呼叫 API 操作時請求 MFA，請將 MFA 條件新增至您的政策。如需詳細資訊，請參閱 [IAM 使用者指南](#) 中的設定 MFA 保護的 API 存取。

有關 IAM 中最佳實務的詳細資訊，請參閱 IAM 使用者指南中的 [IAM 安全最佳實務](#)。

## 使用 AWS Management Console

若要存取 Amazon 轉錄主控台，您必須擁有最基本的一組許可。這些許可必須允許您列出和檢視您 AWS 帳戶中 Amazon Transcribe 資源的詳細資訊。如果您建立比最基本必要許可更嚴格的身分型政策，則對於具有該政策的實體 (使用者或角色) 而言，主控台就無法如預期運作。

對於僅呼叫 AWS CLI 或 AWS API 的使用者，您不需要允許其最基本主控台許可。反之，只需允許存取符合他們嘗試執行之 API 操作的動作就可以了。

若要確保實體 (使用者和角色) 可以使用 [AWS Management Console](#)，請將下列其中一個 AWS 受管理的原則附加至這些實體。

- `AmazonTranscribeFullAccess`：授予建立、讀取、更新、刪除和執行所有 Amazon Transcribe 資源的完整存取權。它還允許訪問存 Amazon S3 儲桶名稱 `transcribe` 中的存儲桶。
- `AmazonTranscribeReadOnlyAccess`：授予 Amazon Transcribe 資源的唯讀存取權，以便您取得和列出轉錄工作和自訂字彙。

### Note

您可以登入政策名稱 `IAMAWS Management Console` 並搜尋政策名稱，來檢閱受管許可政策政策範例。搜尋「轉錄」會傳回上述 (`AmazonTranscribeReadOnly` 和 `AmazonTranscribeFullAccess`) 兩個政策。

您也可以建立自訂 IAM 政策，授予執行 Amazon Transcribe API 動作的相關許可。您可以將這些自訂政策連接至需要這些許可的實體。

## IAM角色所需的許可

如果您建立要呼叫的IAM角色Amazon Transcribe，則該角色必須具有存取Amazon S3值區的權限。如果適用，也KMS key必須使用來加密值區的內容。如需政策範例，請參閱下列各節範例。

### 信任政策

您用來提出轉錄要求的IAM實體必須具有信任原則，Amazon Transcribe才能擔任該角色。請使用下列Amazon Transcribe信任政策。請注意，如果您在啟用通話後分析的情況下提出即時通話分析要求，則必須使用「即時通話分析的信任政策」。

### 信任政策Amazon Transcribe

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "transcribe.amazonaws.com"
        ]
      },
      "Action": [
        "sts:AssumeRole"
      ],
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "111122223333"
        },
        "StringLike": {
          "aws:SourceArn": "arn:aws:transcribe:us-west-2:111122223333:*"
        }
      }
    }
  ]
}
```

### 即時通話分析的信任政策

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "transcribe.streaming.amazonaws.com"
        ]
      },
      "Action": [
        "sts:AssumeRole"
      ],
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "111122223333"
        },
        "StringLike": {
          "aws:SourceArn": "arn:aws:transcribe:us-west-2:111122223333:*"
        }
      }
    }
  ]
}
```

## Amazon S3輸入桶政策

下列原則授予IAM角色存取指定輸入值區中檔案的權限。

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "s3:GetObject",
      "s3:ListBucket"
    ],
    "Resource": [
      "arn:aws:s3:::DOC-EXAMPLE-INPUT-BUCKET",
      "arn:aws:s3:::DOC-EXAMPLE-INPUT-BUCKET/*"
    ]
  }
}
```

```
}
```

## Amazon S3輸出桶政策

下列原則授予IAM角色權限，可將檔案寫入指定的輸出值區。

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "s3:PutObject"
    ],
    "Resource": [
      "arn:aws:s3:::DOC-EXAMPLE-OUTPUT-BUCKET/*"
    ]
  }
}
```

## Amazon S3加密金鑰所需的權限

如果您使用KMS key加密Amazon S3值區，請在KMS key政策中包含下列內容。如此可讓 Amazon Transcribe 存取儲存貯體的內容。如需有關允許存取的詳細資訊KMS keys，請參閱AWS KMS開發人員指南KMS key中的[允許外部AWS 帳戶存取](#)。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
      },
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": "arn:aws:kms:us-west-2:111122223333:key/KMS-Example-KeyId"
    }
  ]
}
```

## 允許使用者檢視他們自己的許可

此範例會示範如何建立政策，允許 IAM 使用者檢視連接到他們使用者身分的內嵌及受管政策。此政策包含在主控台上，或是使用 AWS CLI 或 AWS API 透過編寫程式的方式完成此動作的許可。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsForUser",
        "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": "*"
    }
  ]
}
```

## AWS KMS加密內容政策

下列原則會授與IAM角色「ExampleRole」權限，以針對此特定項目使用「AWS KMS解密」和「加密」作業KMS key。此原則僅適用於至少具有一個加密內容配對的要求，在本例中為color:indigoBlue ""。如需AWS KMS加密內容的詳細資訊，請參閱[AWS KMS 加密內容](#)。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
      },
      "Action": [
        "kms:Decrypt",
        "kms:DescribeKey",
        "kms:Encrypt",
        "kms:GenerateDataKey*",
        "kms:ReEncrypt*"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:EncryptionContext:color": "indigoBlue"
        }
      }
    }
  ]
}
```

## 預防政策混淆代理人

以下是假設角色原則的範例，其中顯示如何使用aws:SourceArn和使aws:SourceAccount用Amazon Transcribe來防止混淆的副問題。有關混淆副預防的更多信息，請參閱[預防跨服務混淆代理人](#)。

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



```

    "Principal": {
      "Service": "transcribe.amazonaws.com"
    },
    "Action": [
      "sts:AssumeRole",
    ],
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "111122223333"
      },
      "StringLike": {
        "aws:SourceArn": "arn:aws:transcribe:us-west-2:111122223333:*"
      }
    }
  }
]
}

```

## 根據標籤檢視轉錄任務

您可以在身分型政策中使用條件，來根據標籤控制存取 Amazon Transcribe 資源。此範例會示範範範範範範範例會示範範範範範範範例建立政策範例，示範範範例不過，只有在轉錄作業標籤Owner具有該使用者的使用者名稱值時，才會授予許可。此政策也會授予使用完成此動作的必要許可AWS Management Console。

您可以將此政策連接至您帳戶中的IAM實體。如果名為的角色test-role嘗試檢視轉錄工作，則轉錄工作必須加上標籤Owner=test-role或owner=test-role (條件索引鍵名稱不區分大小寫)，否則會拒絕存取。如需詳細資訊，請參閱《IAM使用者指南》中的 [IAMJSON 政策元素：條件](#)。

若要取得有關中標籤的更多資訊Amazon Transcribe，請參閱[標記 資源](#)。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ListTranscriptionJobsInConsole",
      "Effect": "Allow",
      "Action": "transcribe:ListTranscriptionJobs",
      "Resource": "*"
    },
    {
      "Sid": "ViewTranscriptionJobsIfOwner",

```

```
        "Effect": "Allow",
        "Action": "transcribe:GetTranscriptionJobs",
        "Resource": "arn:aws:transcribe:*:*:transcription-job/*",
        "Condition": {
            "StringEquals": {"aws:ResourceTag/Owner": "${aws:username}"}
        }
    ]
}
```

## 疑難排解 Amazon Transcribe 身分和存取

使用下列資訊來診斷並修正使用和 AWS Identity and Access Management (IAM) 時可能會遇到的 Amazon Transcribe 常見問題。

### 主題

- [我未獲授權，不得在 Amazon Transcribe 中執行動作](#)
- [我沒有授權執行 iam : PassRole](#)
- [我想允許我以外的人訪 AWS 帳戶 問我的 Amazon Transcribe 資源](#)

### 我未獲授權，不得在 Amazon Transcribe 中執行動作

如果您收到錯誤，告知您未獲授權執行動作，您的政策必須更新，允許您執行動作。

下列範例錯誤會在 mateojackson IAM 使用者嘗試使用主控台檢視一個虛構 *my-example-widget* 資源的詳細資訊，但卻無虛構 *transcribe:GetWidget* 許可時發生。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
transcribe:GetWidget on resource: my-example-widget
```

在此情況下，必須更新 mateojackson 使用者的政策，允許使用 *transcribe:GetWidget* 動作存取 *my-example-widget* 資源。

如果您需要協助，請聯絡您的 AWS 系統管理員。您的管理員提供您的簽署憑證。

### 我沒有授權執行 iam : PassRole

如果您收到錯誤，告知您未獲授權執行 *iam:PassRole* 動作，您的政策必須更新，允許您將角色傳遞給 Amazon Transcribe。

有些 AWS 服務 允許您將現有角色傳遞給該服務，而不是建立新的服務角色或服務連結角色。如需執行此作業，您必須擁有將角色傳遞至該服務的許可。

名為 marymajor 的 IAM 使用者嘗試使用主控台在 Amazon Transcribe 中執行動作時，發生下列範例錯誤。但是，動作要求服務具備服務角色授予的許可。Mary 沒有將角色傳遞至該服務的許可。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

在這種情況下，Mary 的政策必須更新，允許她執行 iam:PassRole 動作。

如果您需要協助，請聯絡您的 AWS 系統管理員。您的管理員提供您的簽署憑證。

## 我想允許我以外的人訪 AWS 帳戶 問我的 Amazon Transcribe 資源

您可以建立一個角色，讓其他帳戶中的使用者或您組織外部的人員存取您的資源。您可以指定要允許哪些信任物件取得該角色。針對支援基於資源的政策或存取控制清單 (ACL) 的服務，您可以使用那些政策來授予人員存取您的資源的許可。

如需進一步了解，請參閱以下內容：

- 若要瞭解是否 Amazon Transcribe 支援這些功能，請參閱 [如何與 IAM Amazon Transcribe 搭配使用](#)。
- 若要了解如何提供您所擁有資源 AWS 帳戶 的存取權，請參閱 [《IAM 使用者指南》中的另一個您擁有 AWS 帳戶 的 IAM 使用者提供存取權限](#)。
- 若要了解如何向第三方提供對資源的存取權 AWS 帳戶，請參閱 [IAM 使用者指南中的提供第三方 AWS 帳戶 擁有的存取權](#)。
- 如需了解如何透過聯合身分提供存取權，請參閱 IAM 使用者指南中的 [將存取權提供給在外部進行身分驗證的使用者 \(聯合身分\)](#)。
- 若要了解跨帳戶存取使用角色和以資源為基礎的政策之間的差異，請參閱 IAM 使用者指南中的 [IAM 中的跨帳戶資源存取](#)。

## Amazon Transcribe 中的資料保護

AWS [共同的責任模型](#) 適用於 Amazon Transcribe 中的資料保護。如此模型所述，AWS 負責保護執行所有 AWS 雲端 的全球基礎設施。您負責維護在此基礎設施上託管內容的控制權。您也必須負責您所使用 AWS 服務 的安全組態和管理任務。如需有關資料隱私權的更多相關資訊，請參閱 [資料隱私權常](#)

[見問答集](#)。如需有關歐洲資料保護的相關資訊，請參閱 AWS 安全性部落格上的 [AWS 共同的責任模型](#) 和 [GDPR](#) 部落格文章。

基於資料保護目的，建議您使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 保護 AWS 帳戶憑證，並設定個人使用者。如此一來，每個使用者都只會獲得授與完成其任務所必須的許可。我們也建議您採用下列方式保護資料：

- 每個帳戶均要使用多重要素驗證 (MFA)。
- 使用 SSL/TLS 與 AWS 資源通訊。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 使用 AWS CloudTrail 設定 API 和使用者活動日誌記錄。
- 使用 AWS 加密解決方案，以及 AWS 服務內的所有預設安全控制項。
- 使用進階的受管安全服務（例如 Amazon Macie），協助探索和保護儲存在 Amazon S3 的敏感資料。
- 如果您在透過命令列介面或 API 存取 AWS 時，需要 FIPS 140-2 驗證的加密模組，請使用 FIPS 端點。如需有關 FIPS 和 FIPS 端點的更多相關資訊，請參閱 [聯邦資訊處理標準 \(FIPS\) 140-2 概觀](#)。

我們強烈建議您絕對不要將客戶的電子郵件地址等機密或敏感資訊，放在標籤或自由格式的文字欄位中，例如 Name(名稱) 欄位。這包括當您使用 Amazon Transcribe 或使用主控台、API、AWS CLI 或 AWS 開發套件的其他 AWS 服務。您在標籤或自由格式文字欄位中輸入的任何資料都可能用於計費或診斷日誌。如果您提供外部伺服器的 URL，我們強烈建議請勿在驗證您對該伺服器請求的 URL 中包含憑證資訊。

## 網際網路流量隱私權

Amazon Transcribe 的 Amazon Virtual Private Cloud (Amazon VPC) 端點是 VPC 內的邏輯實體，僅允許連接到 Amazon Transcribe。Amazon VPC 會將請求路由到 Amazon Transcribe 並將回應路由傳回 VPC。如需詳細資訊，請參閱 [AWS PrivateLink 概念](#)。如需有關搭配 Amazon Transcribe 使用 Amazon VPC 端點的詳細資訊，請參閱 [Amazon Transcribe 和介面 VPC 端點 \(AWS PrivateLink\)](#)。

## 資料加密

資料加密是指無需保護傳輸中和靜態資料。在傳輸過程中，您可以使用 Amazon S3 託管密鑰或 KMS keys 靜態密鑰以及標準傳輸層安全性 ( TLS ) 來保護您的數據。

## 靜態加密

Amazon Transcribe 使用預設 Amazon S3 金鑰 (SSE-S3) 來進行 Amazon S3 儲存貯體中的成績單的伺服器端加密。

使用此 [StartTranscriptionJob](#) 作業時，您可以指定自己的項目 KMS key 來加密轉錄工作的輸出。

Amazon Transcribe 使用透過預設金鑰加密的 Amazon EBS 磁碟區。

## 傳輸中加密

Amazon Transcribe 將 TLS 1.2 與 AWS 憑證搭配使用，以加密傳輸中的資料。這包括流轉錄。

## 金鑰管理

Amazon Transcribe 與一起 KMS keys 為您的資料提供增強的加密。使用 Amazon S3，您可以在建立轉錄工作時加密輸入媒體。與集成 AWS KMS 允許對 [StartTranscriptionJob](#) 請求的輸出進行加密。

如果未指定 KMS key，轉錄工作的輸出會使用預設 Amazon S3 金鑰加密 (SSE-S3)。

如需詳細資訊 AWS KMS，請參閱 [AWS Key Management Service 開發人員指南](#)。

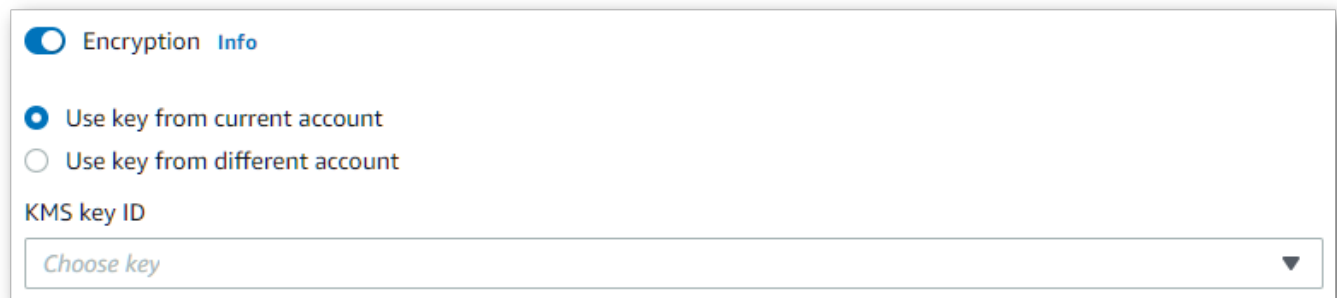
使用金鑰管理 AWS Management Console

若要加密轉錄工作的輸出，您可以選擇使用正在提 KMS key 出要求的輸 KMS key 出，或使用另一個 AWS 帳戶。AWS 帳戶

如果未指定 KMS key，轉錄工作的輸出會使用預設 Amazon S3 金鑰加密 (SSE-S3)。

若要啟用輸出加密：

1. 在 Output data (輸出資料) 下方，選擇 Encryption (加密)。



Encryption Info

Use key from current account

Use key from different account

KMS key ID

Choose key ▼

2. 選擇 KMS key 是否來自 AWS 帳戶您目前正在使用的或來自不同的 AWS 帳戶。如果要使用當前密鑰 AWS 帳戶，請從 KMS keyID 中選擇密鑰。如果您使用不同的金鑰 AWS 帳戶，您必須輸入金鑰

的 ARN。若要使用不同的金鑰AWS 帳戶，呼叫 `kms:Encrypt` 者必須具有 KMS key。如需詳細資訊，請參閱[建立金鑰原則](#)。

## 使用 API 進行金鑰管理

若要搭配 API 使用輸出加密，您必須 KMS key 使用 [StartCallAnalyticsJobStartMedicalTranscriptionJob](#)、或 [StartTranscriptionJob](#) 作業的 `OutputEncryptionKMSKeyId` 參數來指定。

如果使用位於目前的金鑰AWS 帳戶，您可以使用下列四種方式 KMS key 之一來指定您的金鑰：

1. 使用 KMS key ID 本身。例如：1234abcd-12ab-34cd-56ef-1234567890ab。
2. 使用別名作為 KMS key ID。例如：alias/ExampleAlias。
3. 使用 Amazon Resource Name (ARN)。KMS key 例如：arn:aws:kms:region:account-ID:key/1234abcd-12ab-34cd-56ef-1234567890ab。
4. 使用 ARN 作為 KMS key 別名。例如：arn:aws:kms:region:account-ID:alias/ExampleAlias。

如果使用與目前不同 AWS 帳戶的金鑰AWS 帳戶，您可以使用以下兩種方式 KMS key 之一來指定您的金鑰：

1. 使用 ARN 作為 KMS key 識別碼。例如：arn:aws:kms:region:account-ID:key/1234abcd-12ab-34cd-56ef-1234567890ab。
2. 使用 ARN 作為 KMS key 別名。例如：arn:aws:kms:region:account-ID:alias/ExampleAlias。

請注意，提出請求的實體必須得到許可才能使用指定的 KMS key。

## AWS KMS 加密內容

AWS KMS 加密上下文是純文本，非秘密鍵：值對的映射。此對映代表其他已驗證的資料 (稱為加密內容配對)，可為您的資料提供額外的安全層。Amazon Transcribe 需要對稱加密金鑰，才能將轉錄輸出加密到客戶指定的 Amazon S3 儲存貯體中。若要進一步，請參閱[無對稱金鑰](#)，請參閱 AWS KMS。

建立加密內容配對時，請勿包含敏感資訊。加密內容不會保密 — 在 CloudTrail 記錄檔中會以純文字顯示 (因此您可以使用它來識別和分類您的加密作業)。

您的加密內容配對可以包含特殊字元，例如底線 (`_`)、破折號 (`-`)、斜線 (`/`, `\`) 和冒號 (`:`)。

**i** Tip

將加密內容配對中的值與要加密的資料相關聯會很有用。雖然不是必要的，但我們建議您使用與加密內容相關的非敏感中繼資料，例如檔案名稱、標頭值或未加密的資料庫欄位。

若要搭配 API 使用輸出加密，請在 [StartTranscriptionJob](#) 作業中設定 `KMSEncryptionContext` 參數。為了提供輸出加密作業的加密內容，`OutputEncryptionKMSKeyId` 參數必須參照對稱 KMS key ID。

您可以將 [AWS KMS 條件金鑰](#) 與 IAM 原則搭配使用，KMS key 根據 [密碼編譯作業要求中使用的加密內容來控制對稱加密的存取](#)。如需加密內容原則的範例，請參閱 [AWS KMS 加密內容政策](#)。

使用加密內容是選用操作，但建議您採用。如需，請參閱 [加密內容](#)。

## 選擇不使用您的資料來改善服務

根據預設，會 Amazon Transcribe 儲存並使用已處理的語音輸入，以開發服務並持續改善您的使用體驗。您可以使用退出政策選擇 AWS Organizations 退出使用您的內容 Amazon Transcribe 來開發和改進。如需如何 [退出政策](#)。

## 監控 Amazon Transcribe

監控是維持其他 AWS 解決方案的可靠性、可用性和效能的 Amazon Transcribe 重要組成部分。AWS 提供下列監控工具來監視 Amazon Transcribe、在發生錯誤時回報，並在適當時自動採取行動：

- Amazon CloudWatch 即時監控您的 AWS 資源和執行 AWS 的應用程式。您可以收集和追蹤指標、建立自訂儀表板，以及設定警示，在特定指標達到您指定的閾值時通知您或採取動作。例如，您可以在 Amazon EC2 執行個體上 CloudWatch 追蹤 CPU 使用率或其他指標，並在需要時自動啟動新執行個體。
- Amazon CloudWatch Logs 可以監控、儲存和存取 Amazon EC2 執行個體和其他來源的記錄檔。CloudTrail CloudWatch Logs 可以監視記錄檔中的資訊，並在符合特定臨界值時通知您。您也可以將日誌資料存檔在高耐用性的儲存空間。
- AWS CloudTrail 擷取由您或代表您進行的 API 呼叫和相關事件，AWS 帳戶 並將記錄檔傳送至您指定的 Amazon S3 值區。您可以識別呼叫的使用者和帳戶 AWS、進行呼叫的來源 IP 位址，以及呼叫發生的時間。

如需詳細資訊，請參閱 [Amazon CloudWatch 使用者指南](#)。



Amazon EventBridge是一種使用事件將應用程式元件連接在一起的無伺服器服務，讓您更輕鬆地建置可擴充的事件驅動型應用程式。EventBridge 提供來自您自己的應用程式、軟體即服務 (SaaS) 應用程式的即時資料串流，以及 AWS 服務，並將資料路由至目標 (例如 Lambda)。您可以監控在服務中發生的事件，並建置事件驅動架構。如需詳細資訊，請參閱 [《Amazon EventBridge 使用者指南》](#)。

## 主題

- [Amazon Transcribe 使用監控 Amazon CloudWatch](#)
- [Amazon Transcribe 使用監控 AWS CloudTrail](#)
- [Amazon EventBridge 搭配使用 Amazon Transcribe](#)

## Amazon Transcribe 使用監控 Amazon CloudWatch

您可以 Amazon Transcribe 使用監視器 CloudWatch，它收集原始數據並將其處理為可讀的近實時指標。這些統計資料會保留 15 個月，以便您存取歷史資訊，並更清楚 Web 應用程式或服務的執行效能。您也可以設定留意特定閾值的警示，當滿足這些閾值時傳送通知或採取動作。如需詳細資訊，請參閱 [《CloudWatch 使用者指南》](#) <https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/WhatIsCloudWatch.html>。

### 搭配使用 Amazon CloudWatch 量度和維度 Amazon Transcribe

Amazon Transcribe 支援 CloudWatch 指標和維度，這些資料可協助您監控效能。支援的指標類別包括與轉錄作業相關聯的流量、錯誤、資料傳輸和延遲。支援的量度位於 AWS/Transcribe 命名空間 CloudWatch 間中。

#### Note

CloudWatch 監控指標是免費的，不會計入 CloudWatch 服務配額。

如需有關 CloudWatch 量度的詳細資訊，請參閱 [使用 Amazon CloudWatch 量度](#)。

## Amazon Transcribe 使用監控 AWS CloudTrail

Amazon Transcribe 與 AWS CloudTrail提供 AWS Identity and Access Management (IAM) 使用者或角色或服務 Amazon Transcribe 所採取之動作記錄的 AWS 服務整合。CloudTrail 會擷取所有的 API 呼叫 Amazon Transcribe。其中包括來自 Amazon Transcribe API 的呼叫 AWS Management Console 和程式碼呼叫，做為事件。藉由建立追蹤，您可以啟用持續將 CloudTrail 事件 (包括事件 Amazon



Transcribe) 傳遞至 Amazon S3 值區。如果您不建立追蹤，仍然可以在 Event history(事件歷史記錄)，檢視 CloudTrail AWS Management Console 中的最新事件。使用所收集的資訊 CloudTrail，您可以查看提出的每個要求 Amazon Transcribe、提出要求的 IP 位址、提出要求的人員、提出要求的時間，以及其他詳細資訊。

若要進一步瞭解 CloudTrail，請參閱[AWS CloudTrail 使用者指南](#)。

## Amazon Transcribe 而且 CloudTrail

CloudTrail 在您創建帳戶 AWS 帳戶 時啟用。當活動發生在中時 Amazon Transcribe，該活動會與 CloudTrail 事件歷史記錄中的其他 AWS 服務 事件一起記錄在 CloudTrail 事件中。您可以查看，搜索和下載最近的事件 AWS 帳戶。如需詳細資訊，請參閱[使用 CloudTrail 事件歷程記錄檢視事件](#)。

要獲取您的事件的持續記錄 AWS 帳戶，包括事件 Amazon Transcribe，請創建一個跟踪。追蹤是可 CloudTrail 將事件當做記錄檔傳遞至指定 Amazon S3 值區的組態。CloudTrail 記錄檔包含一或多個記錄項目。事件代表來自任何來源的單一請求。它包括所請求的動作、動作的日期和時間、請求參數等相關資訊。CloudTrail 日誌文件不是公共 API 調用的有序堆棧跟踪，因此它們不會以任何特定順序顯示。

依預設，當您在中建立系統線時 AWS Management Console，系統線會套用至所有系統線 AWS 區域。追蹤記錄來自 AWS 磁碟分割 AWS 區域 中全部的事件，並將記錄檔傳送至您指定的 Amazon S3 值區。此外，您可以設定其他，AWS 服務 以進一步分析 CloudTrail 記錄中收集的事件資料並採取行動。如需詳細資訊，請參閱：

- [建立追蹤的概觀](#)
- [CloudTrail 支援的服務與整合](#)
- [設定 Amazon SNS 通知 CloudTrail](#)
- [接收多個區域的 CloudTrail 日誌檔案及接收多個帳戶的 CloudTrail 日誌檔案](#)

CloudTrail 記錄所有 Amazon Transcribe 操作，這些操作記錄在 [API 參考](#)中。例如，[CreateVocabularyGetTranscriptionJob](#)、和[StartTranscriptionJob](#)作業會在 CloudTrail 記錄檔中產生項目。

每一筆事件或日誌專案都會包含產生請求者的資訊。此資訊可協助您判斷下列事項：

- 要求是使用根憑證還是 IAM 使用者認證
- 提出該請求時，是否使用特定 IAM 角色或聯合身分使用者的暫時安全登入資料
- 請求是否由另一個人提出 AWS 服務

如需詳細資訊，請參閱 [CloudTrail 使用者身分元素](#)。

您也可以將多個 AWS 區域 和多個 Amazon Transcribe 記錄檔彙總 AWS 帳戶 到單一 Amazon S3 儲存貯體中。如需詳細資訊，請參閱 [從多個區域接收 CloudTrail 記錄檔](#) 和 [從多個帳戶接收記錄檔](#)。

範例：Amazon Transcribe 記錄檔項目

追蹤是一種設定，可讓事件以記錄檔的形式傳遞至指定 Amazon S3 值區。CloudTrail 記錄檔包含一或多個記錄項目。事件代表來自任何來源的單一請求。它包括有關請求的操作作為操作的日期和時間，以及請求參數這樣的信息。CloudTrail 日誌文件不是公共 API 調用的有序堆棧跟踪，因此它們不會以任何特定順序顯示。

[StartTranscriptionJob](#) 和 [GetTranscriptionJob](#) 操作的呼叫會建立以下項目。

```
{
  "Records": [
    {
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "111122223333",
        "arn": "arn:aws:iam:us-west-2:111122223333:user/my-user-name",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "my-user-name"
      },
      "eventTime": "2022-03-07T15:03:45Z",
      "eventSource": "transcribe.amazonaws.com",
      "eventName": "StartTranscriptionJob",
      "awsRegion": "us-west-2",
      "sourceIPAddress": "127.0.0.1",
      "userAgent": "[]",
      "requestParameters": {
        "mediaFormat": "flac",
        "languageCode": "en-US",
        "transcriptionJobName": "my-first-transcription-job",
        "media": {
          "mediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-media-file.flac"
        }
      },
      "responseElements": {
        "transcriptionJob": {
```

```

        "transcriptionJobStatus": "IN_PROGRESS",
        "mediaFormat": "flac",
        "creationTime": "2022-03-07T15:03:44.229000-08:00",
        "transcriptionJobName": "my-first-transcription-job",
        "languageCode": "en-US",
        "media": {
            "mediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-media-file.flac"
        }
    },
    "requestID": "47B8E8D397DCE7A6",
    "eventID": "cdc4b7ed-e171-4cef-975a-ad829d4123e8",
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
},
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "111122223333",
        "arn": "arn:aws:iam:us-west-2:111122223333:user/my-user-name",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "my-user-name"
    },
    "eventTime": "2022-03-07T15:07:11Z",
    "eventSource": "transcribe.amazonaws.com",
    "eventName": "GetTranscriptionJob",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "[]",
    "requestParameters": {
        "transcriptionJobName": "my-first-transcription-job"
    },
    "responseElements": {
        "transcriptionJob": {
            "settings": {

            },
            "transcriptionJobStatus": "COMPLETED",
            "mediaFormat": "flac",
            "creationTime": "2022-03-07T15:03:44.229000-08:00",
            "transcriptionJobName": "my-first-transcription-job",
            "languageCode": "en-US",

```

```
        "media": {
            "mediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-media-file.flac"
        },
        "transcript": {
            "transcriptFileUri": "s3://DOC-EXAMPLE-BUCKET/my-first-
transcription-job.json"
        }
    },
    "requestID": "BD8798EACDD16751",
    "eventID": "607b9532-1423-41c7-b048-ec2641693c47",
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
]
```

## Amazon EventBridge 搭配使用 Amazon Transcribe

使用 Amazon EventBridge，您可以透過啟動其 AWS 服務他事件中的事件來回應 Amazon Transcribe 工作中的狀態變更。當轉錄工作變更狀態時，EventBridge 會自動將事件傳送至事件串流。您會建立規則以定義要在事件串流中監控的事件，以及發生這些事件時 EventBridge 應該採取的動作。例如，將事件路由到另一個服務 (或目標)，此服務可接著採取動作。例如，您可以設定規則，在轉錄工作成功完成時，將事件路由至 AWS Lambda 函數。若要定義 [EventBridge 規則](#)，請參閱下列各節。

您可以透過多個管道接收事件通知，包括電子郵件、[AWS Chatbot](#) 聊天通知或 [AWS Console Mobile Application](#) 推送通知。您也可以在 [主控台通知中心](#) 查看通知。如果要設置通知，則可以使用 [AWS 使用者通知](#)。AWS 使用者通知 支援彙總，這樣可以減少您在特定事件期間收到的通知數量。

### 定義 EventBridge 規則

若要定義 EventBridge 規則，請使用 [AWS Management Console](#)。定義規則時，請使用 Amazon Transcribe 做為服務名稱。如需如何建立規則的範例，請參閱 [Amazon EventBridge 規則](#)。

使用前 EventBridge，請注意以下定義：

- 事件-事件表示您其中一個轉錄作業的狀態變更。例如，工作的 `TranscriptionJobStatus` 從 `IN_PROGRESS` 變更為 `COMPLETED`。
- 目標-目標是處理事件的另一個 AWS 服務。例如，AWS Lambda 或 Amazon Simple Notification Service (Amazon SNS)。目標會收到 JSON 格式的事件。

- 規則 — 規則符合您要監視的內送事件，並 EventBridge 將其路由至一或多個目標以進行處理。如果規則將事件路由到多個目標，所有的目標都會平行處理事件。規則可自訂傳送到目標的 JSON。

Amazon EventBridge 事件會以最大的努力為基礎發出。若要取得有關在中建立和管理事件的更多資訊 EventBridge，請參閱《Amazon EventBridge 使用指南》中的[Amazon EventBridge 事件](#)。

以下是轉錄工作狀態變更 Amazon Transcribe 為 COMPLETED 或 FAILED 時所起始 EventBridge 規則的範例。

```
{
  "source": [
    "aws.transcribe"
  ],
  "detail-type": [
    "Transcribe Job State Change"
  ],
  "detail": {
    "TranscriptionJobStatus": [
      "COMPLETED",
      "FAILED"
    ]
  }
}
```

該規則包含下列欄位：

- source—事件的來源。因為 Amazon Transcribe，這總是如此 `aws.transcribe`。
- detail-type—事件詳細資訊的識別子。對於 Amazon Transcribe，這一律為 `Transcribe Job State Change`。
- detail—轉錄作業的新作業狀態。在這個範例中，規則會在工作狀態變更為 `COMPLETED` 或 `FAILED` 時啟動事件。

## Amazon Transcribe 事件

Amazon EventBridge 記錄幾個 Amazon Transcribe 事件：

- [轉錄作業事件](#)
- [語言識別事件](#)
- [呼叫分析事件](#)

- [通話分析通話後事件](#)
- [詞彙事件](#)

這些事件都包含下列共用欄位：

- `version`：事件資料的版本。這個值一律為 0。
- `id`：為事件產生的 EventBridge 唯一識別碼。
- `detail-type`：事件詳細資訊的識別子。例如 Transcribe Job State Change。
- `source`：事件的來源。因為 Amazon Transcribe 這總是如此 `aws.transcribe`。
- `account`：產生 API 呼叫的帳戶 AWS 帳戶 識別碼。
- `time`：事件傳送的日期和時間。
- `region`：提 AWS 區域 出要求的位置。
- `resources`：API 呼叫使用的資源。對於 Amazon Transcribe，此欄位永遠是空的。
- `detail`：有關事件的其他詳細資訊。
  - `FailureReason`：如果狀況或狀態變更為 FAILED，則會顯示此欄位，並描述 FAILED 狀況或狀態的原因。
  - 每個事件類型都有其他唯一欄位，顯示在 `detail` 下方。這些唯一欄位會在每個事件範例後的下列各節定義。

### 轉錄作業事件

當工作的狀態從變更 IN\_PROGRESS 為 COMPLETED 或時 FAILED，Amazon Transcribe 會產生事件。若要識別變更狀態且在目標中啟動事件的工作，請使用該事件的 `TranscriptionJobName` 欄位。Amazon Transcribe 事件包含下列資訊。如果您的轉錄作業狀態為 FAILED，則會在 `detail` 下方新增 `FailureReason` 欄位。

請注意，此事件僅適用於 [StartTranscriptionJob](#) API 操作。

```
{
  "version": "0",
  "id": "event ID",
  "detail-type": "Transcribe Job State Change",
  "source": "aws.transcribe",
  "account": "111122223333",
  "time": "timestamp",
  "region": "us-west-2",
```

```
"resources": [],
"detail": {
  "TranscriptionJobName": "my-first-transcription-job",
  "TranscriptionJobStatus": "COMPLETED" (or "FAILED")
}
}
```

- `TranscriptionJobName` : 您為轉錄作業選擇的唯一名稱。
- `TranscriptionJobStatus` : 轉錄作業的狀態。此值可以為 `COMPLETED` 或 `FAILED`。

## 語言識別事件

您啟用[自動語言識別](#)時，Amazon Transcribe 會在語言識別狀態為 `COMPLETED` 或 `FAILED` 時產生事件。若要識別變更狀態且在目標中啟動事件的工作，請使用該事件的 `JobName` 欄位。Amazon Transcribe 事件包含下列資訊。如果您的語言識別狀態為 `FAILED`，則會在 `detail` 下方新增 `FailureReason` 欄位。

請注意，此事件僅適用於包含 [LanguageIdSettings](#) 參數時的 [StartTranscriptionJob](#) API 操作。

```
{
  "version": "0",
  "id": "event ID",
  "detail-type": "Language Identification State Change",
  "source": "aws.transcribe",
  "account": "111122223333",
  "time": "timestamp",
  "region": "us-west-2",
  "resources": [],
  "detail": {
    "JobType": "TranscriptionJob",
    "JobName": "my-first-lang-id-job",
    "LanguageIdentificationStatus": "COMPLETED" (or "FAILED")
  }
}
```

- `JobType`: 對於轉錄作業，此值必須是 `TranscriptionJob`。
- `JobName` : 轉錄作業的唯一名稱。
- `LanguageIdentificationStatus` : 轉錄作業中語言識別的狀態。此值可以為 `COMPLETED` 或 `FAILED`。

## 呼叫分析事件

[通話分析](#)工作狀態從 IN\_PROGRESS 變更為 COMPLETED 或 FAILED 時，Amazon Transcribe 就會產生事件。若要識別變更狀態且在目標中啟動事件的通話分析工作，請使用該事件的 JobName 欄位。Amazon Transcribe 事件包含下列資訊。如果您的通話分析工作狀態為 FAILED，則會在 detail 下方新增 FailureReason 欄位。

請注意，此事件僅適用於 [StartCallAnalyticsJob](#) API 操作。

```
{
  "version": "0",
  "id": "event ID",
  "detail-type": "Call Analytics Job State Change",
  "source": "aws.transcribe",
  "account": "111122223333",
  "time": "timestamp",
  "region": "us-west-2",
  "resources": [],
  "detail": {
    "JobName": "my-first-analytics-job",
    "JobStatus": "COMPLETED" (or "FAILED"),
    "AnalyticsJobDetails": { // only when you enable optional features such as
      Generative Call Summarization
      "Skipped": []
    }
  }
}
```

- JobName：通話分析轉錄作業的唯一名稱。
- JobStatus：您的通話分析轉錄作業的狀態。可以是 COMPLETED 或 FAILED。
- AnalyticsJobDetails：您的呼叫分析轉錄工作的詳細信息，包括有關跳過的分析功能的信息。

## 通話分析通話後事件

[通話後分析](#)轉錄狀態從 IN\_PROGRESS 變更為 COMPLETED 或 FAILED 時，Amazon Transcribe 會產生事件。若要識別變更狀態且在目標中啟動事件的通話分析通話後工作，請使用該事件的 StreamingSessionId 欄位。

請注意，此事件僅適用於包含 [PostCallAnalyticsSettings](#) 參數時的 [StartCallAnalyticsStreamTranscription](#) API 操作。



COMPLETED 事件包含下列資訊：

```
{
  "version": "0",
  "id": "event ID",
  "detail-type": "Call Analytics Post Call Job State Change",
  "source": "aws.transcribe",
  "account": "111122223333",
  "time": "timestamp",
  "region": "us-west-2",
  "resources": [],
  "detail": {
    "StreamingSessionId": "session-id",
    "PostCallStatus": "COMPLETED",
    "Transcript": {
      "RedactedTranscriptFileUri": "s3://DOC-EXAMPLE-BUCKET/my-output-files/my-redacted-file.JSON",
      "TranscriptFileUri": "s3://DOC-EXAMPLE-BUCKET/my-output-files/my-file.JSON"
    },
    "Media": {
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-output-files/my-redacted-file.WAV",
      "RedactedMediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-output-files/my-redacted-file.WAV"
    }
  }
}
```

FAILED 事件包含下列資訊：

```
{
  "version": "0",
  "id": "event ID",
  "detail-type": "Call Analytics Post Call Job State Change",
  "source": "aws.transcribe",
  "account": "111122223333",
  "time": "timestamp",
  "region": "us-west-2",
  "resources": [],
  "detail": {
    "StreamingSessionId": "session-id",
    "PostCallStatus": "FAILED"
  }
}
```

```
}
```

- `StreamingSessionId` : 指定給您即時通話分析轉錄請求的識別號碼。
- `PostCallStatus` : 通話後通話分析轉錄的狀態。可以是 `COMPLETED` 或 `FAILED`。
- `Transcript` : 您修訂和未修訂的文字記錄的 URI。
- `Media` : 您修訂和未修訂的音訊檔案的 URI。

## 詞彙事件

當 [自訂字彙](#) 的狀態從變更 `PENDING` 為 `READY` 或時 `FAILED`，Amazon Transcribe 會產生事件。若要識別變更狀態且在目標中啟動事件的自訂詞彙工作，請使用該事件的 `VocabularyName` 欄位。Amazon Transcribe 事件包含下列資訊。如果您的自訂詞彙狀態為 `FAILED`，則會在 `detail` 下方新增 `FailureReason` 欄位。

請注意，此事件僅適用於 [CreateVocabulary](#) API 操作。

```
{
  "version": "0",
  "id": "event ID",
  "detail-type": "Vocabulary State Change",
  "source": "aws.transcribe",
  "account": "111122223333",
  "time": "timestamp",
  "region": "us-west-2",
  "resources": [],
  "detail": {
    "VocabularyName": "unique-vocabulary-name",
    "VocabularyState": "READY" (or "FAILED")
  }
}
```

- `VocabularyName` : 您自訂詞彙的唯一名稱。
- `VocabularyState` : 自訂詞彙的處理狀態。此值可以為 `READY` 或 `FAILED`。

## 符合性驗證 Amazon Transcribe

若要瞭解 AWS 服務 是否屬於特定規範遵循方案的範圍內，請參閱 [AWS 服務 遵循規範計劃](#) 方案中的，並選擇您感興趣的合規方案。如需一般資訊，請參閱 [AWS 規範計劃](#) [AWS](#)。

您可以使用下載第三方稽核報告 AWS Artifact。如需詳細資訊，請參閱[下載中的報告中的 AWS Artifact](#)。

您在使用時的合規責任取決 AWS 服務 於您資料的敏感性、公司的合規目標以及適用的法律和法規。AWS 提供下列資源以協助遵循法規：

- [安全性與合規性快速入門指南](#) — 這些部署指南討論架構考量，並提供部署以安全性和合規性 AWS 為重點的基準環境的步驟。
- [在 Amazon Web Services 上架構 HIPAA 安全性與合規性](#) — 本白皮書說明公司如何使用建立符合 HIPAA 資格的應 AWS 應用程式。

#### Note

並非所有人 AWS 服務 都符合 HIPAA 資格。如需詳細資訊，請參閱 [HIPAA 資格服務參照](#)。

- [AWS 合規資源AWS](#) — 此工作簿和指南集合可能適用於您的產業和所在地。
- [AWS 客戶合規指南](#) — 透過合規的角度瞭解共同的責任模式。這份指南總結了在多個架構 (包括美國國家標準技術研究所 (NIST)、支付卡產業安全標準委員會 (PCI) 和國際標準化組織 (ISO)) 中，保 AWS 服務 護指引並對應至安全控制的最佳實務。
- [使用AWS Config 開發人員指南中的規則評估資源](#) — 此 AWS Config 服務會評估您的資源組態符合內部實務、產業準則和法規的程度。
- [AWS Security Hub](#)— 這 AWS 服務 提供了內部安全狀態的全面視圖 AWS。Security Hub 使用安全控制，可評估您的 AWS 資源並檢查您的法規遵循是否符合安全業界標準和最佳實務。如需支援的服務和控制清單，請參閱 [Security Hub controls reference](#)。
- [Amazon GuardDuty](#) — 透過監控環境中的 AWS 帳戶可疑和惡意活動，藉此 AWS 服務 偵測您的工作負載、容器和資料的潛在威脅。GuardDuty 可協助您滿足特定合規性架構所要求的入侵偵測需求，例如 PCI DSS 等各種合規性需求。
- [AWS Audit Manager](#)— 這 AWS 服務 有助於您持續稽核您的 AWS 使用情況，以簡化您管理風險的方式，以及遵守法規和業界標準的方式。

## Amazon Transcribe 中的恢復能力

AWS 全球基礎架構是以 AWS 區域 與可用區域為中心建置的。AWS 區域 提供多個分開且隔離的實際可用區域，並以具備低延遲、高輸送量和高度備援特性的聯網相互連結。透過可用區域，您可以設計與操作的應用程式和資料庫，在可用區域之間自動容錯移轉而不會發生中斷。可用區域的可用性、容錯能力和擴充能力，均較單一或多個資料中心的傳統基礎設施還高。

如需 AWS 區域 與可用區域的詳細資訊，請參閱[AWS全球基礎架構](#)。

## Amazon Transcribe 中的基礎設施安全

作為託管服務，Amazon Transcribe受到AWS全球網絡安全的保護。如需有關 AWS 安全服務以及 AWS 如何保護基礎設施的詳細資訊，請參閱 [AWS 雲端安全](#)。若要使用基礎設施安全性的最佳實務來設計您的 AWS 環境，請參閱安全性支柱 AWS 架構良好的框架中的[基礎設施保護](#)。

您可使用 AWS 發佈的 API 呼叫，透過網路存取 Amazon Transcribe。用戶端必須支援下列項目：

- Transport Layer Security (TLS)。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 具備完美轉送私密 (PFS) 的密碼套件，例如 DHE (Ephemeral Diffie-Hellman) 或 ECDHE (Elliptic Curve Ephemeral Diffie-Hellman)。現代系統 (如 Java 7 和更新版本) 大多會支援這些模式。

此外，請求必須使用存取索引鍵 ID 和與 IAM 主體相關聯的私密存取索引鍵來簽署。或者，您可以使用 [AWS Security Token Service](#) (AWS STS) 來產生暫時安全憑證來簽署請求。

## Amazon Transcribe 中的漏洞分析和管理的

組態和 IT 控制是 AWS 與身為我們客戶的您共同的責任。如需詳細資訊，請參閱 AWS [共同的責任模型](#)。

## Amazon Transcribe 和介面 VPC 端點 (AWS PrivateLink)

您可以建立介面 VPC 端點，以在您的 VPC 與 Amazon Transcribe 間建立私有連線。介面端點由[AWS PrivateLink](#)，您可以用來私人訪問的技術Amazon Transcribe沒有網際網路閘道、NAT 裝置、VPN 連線或AWS Direct Connect連接。VPC 中的執行個體不需要公有 IP 地址，即能與 Amazon Transcribe API 通訊。您的 VPC 與 Amazon Transcribe 之間的網路流量都會在 Amazon 網路的範圍內。

每個介面端點都是由您子網路中的一或多個[彈性網絡介面](#)表示。

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的[介面 VPC 端點 \(AWS PrivateLink\)](#)。

## Amazon Transcribe VPC 端點的考量事項

在為下列項目設定介面 VPC 端點之前Amazon Transcribe，請確定您已檢閱[介面端點內容和限制](#)在Amazon VPC使用者指南。

Amazon Transcribe 支援從您的 VPC 呼叫其所有 API 動作。

## 為 Amazon Transcribe 建立介面 VPC 端點

您可以建立 VPC 端點 Amazon Transcribe 服務使用 Amazon VPC AWS Management Console 或者 AWS CLI。如需詳細資訊，請參閱 [〈建立介面端點〉](#) 在 Amazon VPC 使用者指南。

對於批次轉錄 Amazon Transcribe，使用下列服務名稱建立 VPC 端點：

- COM. 亞馬遜。##-## -2. 轉錄。

對於流轉錄 Amazon Transcribe，使用下列服務名稱建立 VPC 端點：

- COM. 亞馬遜。##-## -2. 轉錄流

如果您為端點啟用私有 DNS，則可以向以下位置發出 API 請求：Amazon Transcribe 使用其預設 DNS 名稱做為 AWS 區域，例如，`transcribestreaming.us-east-2.amazonaws.com`。

如需詳細資訊，請參閱 [〈透過介面端點存取服務〉](#) 在 Amazon VPC 使用者指南。

## 為 Amazon Transcribe 建立 VPC 端點政策

您可以將端點策略附加到您的 VPC 端點，以控制對以下內容的串流服務或批次轉錄服務的存取 Amazon Transcribe。此政策會指定下列資訊：

- 可執行動作的主體。
- 可執行的動作。
- 可供執行動作的資源。

如需詳細資訊，請參閱 [〈透過 VPC 端點控制服務的存取〉](#) 在 Amazon VPC 使用者指南。

範例：的 VPC 端點原則 Amazon Transcribe 批次轉錄動作

以下是中批次轉錄的端點原則範例 Amazon Transcribe。附加至端點後，此政策會針對所有資源上的所有主體，授予列出的 Amazon Transcribe 動作的存取權限。

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Allow",
```

```

    "Action":[
      "transcribe:StartTranscriptionJob",
      "transcribe:ListTranscriptionJobs"
    ],
    "Resource": "*"
  }
]
}

```

## 範例：的 VPC 端點原則 Amazon Transcribe 串流轉錄動作

以下是中串流轉錄的端點原則範例 Amazon Transcribe。附加至端點後，此政策會針對所有資源上的所有主體，授予列出的 Amazon Transcribe 動作的存取權限。

```

{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Allow",
      "Action": [
        "transcribe:StartStreamTranscription",
        "transcribe:StartStreamTranscriptionWebsocket"
      ],
      "Resource": "*"
    }
  ]
}

```

## 共用子網路

您無法在與您共用的子網路中建立、描述、修改或刪除 VPC 端點。不過，可以在與您共用的子網路中使用 VPC 端點。如需 VPC 共用的資訊，請參閱[與其他帳戶共用您的 VPC](#)在 Amazon Virtual Private Cloud 指南。

## Amazon Transcribe 的安全最佳實務

以下最佳實務為一般準則，並不代表完整的安全解決方案。因為這些最佳實務可能不適合或也不足夠，因此請將它們作為實用建議，而不要當作是指示。

- 使用資料加密，例如 AWS KMS 加密內容

AWS KMS加密上下文是純文本，非秘密鍵：值對的映射。此對映代表其他已驗證的資料 (稱為加密內容配對)，可為您的資料提供額外的安全層。

如需詳細資訊，請參閱[AWS KMS 加密內容](#)。

- 盡可能使用臨時登入資料

盡可能使用臨時登入資料，而不是長期登入資料，例如存取金鑰。對於需要具有程式化存取和長期登入資料的IAM使用者，我們建議您輪換存取金鑰。定期輪換長期憑證有助於您熟悉該程序。如果您遇到必須輪換憑證的情況，例如員工離職時，此功能非常有用。我們建議您使用IAM存取上次使用的資訊，以便安全地輪換和移除存取金鑰。

如需詳細資訊，請參閱[中的輪換存取金鑰和安全性最佳作法IAM](#)。

- 針對需要Amazon Transcribe存取IAM權的應用程式和AWS服務使用角色

使用IAM角色來為需存取的應用程式或服務Amazon Transcribe。使用角色時，您不必將長期登入資料 (例如密碼或存取金鑰) 分發至Amazon EC2執行個體或等AWS服務。IAM角色可以提供應用程式在對AWS資源請求時可以使用的臨時許可。

如需詳細資訊，請參閱[IAM角色和常見的角色方案：使用者、應用程式和服務](#)。

- 使用標籤型存取。

您可以使用標籤來控存取AWS 帳戶。標籤可以新增至：轉錄工作、自訂字彙、自訂字彙篩選器和自訂語言模型。Amazon Transcribe

如需詳細資訊，請參閱[標籤型存取控制](#)。

- 使用AWS監視工具

監控是維護 Amazon Transcribe 及您 AWS 解決方案安全性、可靠性、可用性和效能的重要部分。您可以使Amazon Transcribe用CloudTrail。

如需詳細資訊，請參閱[Amazon Transcribe 使用監控 AWS CloudTrail](#)。

- 啟用 AWS Config

AWS Config可以評價、稽核和評估資AWS源組態。使用時AWS Config，您可以檢閱組態中的變更以及AWS資源之間的關係。您也可依據內部準則指定的組態來判斷整體合規情況。如此一來，您就能輕鬆進行合規稽核、安全分析、變更管理和操作故障診斷的程序。

如需詳細資訊，請參閱「[什麼是AWS Config?](#)」



# Amazon Transcribe Medical

Amazon Transcribe Medical 是一種自動語音識別 (ASR) 服務，專為想要轉錄醫療相關語音的醫療專業人員而設計，例如醫生指示的筆記，藥物安全監測，遠程醫療約會或醫生-患者對話。Amazon Transcribe 醫療服務可透過即時串流 (透過麥克風) 或轉錄上傳的檔案 (批次處理) 取得。

## ⚠ Important

Amazon Transcribe 醫療並非專業醫療建議、診斷或治療的替代品。確認使用案例的相關可信度閾值，並在需要高準確性的情況下使用高可信度閾值。對於某些使用案例，應由經過適當訓練的人工審核者審查和驗證結果。Amazon Transcribe 醫療轉錄只能在患者護理情況下使用，以確保受過訓練的醫療專業人員進行準確性和健全的醫療判斷後。

Amazon Transcribe 醫療 AWS 是以共同的責任模式運作，負責保護執行「Amazon Transcribe 醫療」的基礎架構，而您必須負責管理您的資料。如需詳細資訊，請參閱[共同責任模型](#)。

Amazon Transcribe 醫療服務提供美國英語版本 (zh-TW)。

為獲得最佳效果，請使用具有 PCM 16 位元編碼的無損音訊格式，例如 FLAC 或 WAV。Amazon Transcribe Medical 支援 16,000 赫茲或更高的取樣率。

對於成績單的分析，您可以使用其他 AWS 服務，例如[Amazon Comprehend Medical](#)。

## 支持的專業

特色	分專業	音訊輸入
心脏病学	無	僅串流
神經病學	無	僅串流
腫瘤科	無	僅串流
初級保健	家庭醫學	批次, 串流
初級保健	內科	批次, 串流
初級保健	婦產科 (OB-GYN)	批次, 串流



特色	分專業	音訊輸入
初級保健	小兒科	批次, 串流
放射學	無	僅串流
泌尿外科	無	僅串流

## 區域可用性和配額

以下支援呼叫分析AWS 區域：

區域	轉錄類型
af-south-1 (開普敦)	批次
ap-east-1	批次
ap-northeast-1 (東京)	批次, 串流
ap-northeast-2 (首爾)	批次, 串流
ap-south-1 (孟買)	批次
ap-southeast-1 (新加坡)	批次
ap-southeast-2 (雪梨)	批次, 串流
ca-central-1 (加拿大, 中部)	批次, 串流
eu-central-1 (法蘭克福)	批次, 串流
eu-north-1 (斯德哥爾摩)	批次
eu-west-1 (愛爾蘭)	批次, 串流
eu-west-2	批次, 串流
eu-west-3 (巴黎)	批次

區域	轉錄類型
me-south-1 (巴林)	批次
sa-east-1	批次, 串流
us-east-1 (維吉尼亞北部)	批次, 串流
us-east-2 (俄亥俄)	批次, 串流
us-gov-east-1 (GovCloud, 美國東部)	批次, 串流
us-gov-west-1 (GovCloud, 美國西部)	批次, 串流
us-west-1	批次
us-west-2 (奧勒岡)	批次, 串流

請注意 [Amazon Transcribe](#) , Amazon Transcribe Medical、和 [呼叫分析的區域支援不同](#) 。

若要取得每個支援區域的端點，請參閱AWS一般參考中的 [服務端點](#)。

如需與您轉錄相關的配額清單，請參閱AWS一般參考資料中的 [服務配額](#)。部分配額可根據要求進行變更。如果「可調整」欄包含「是」，您可以要求增加。若要執行此作業，請選取提供的連結。

## 醫療與術語

建立醫療轉錄任務時，請指定來源檔案的語言、醫療專業和音訊類型。您輸入美式英文 (en-US) 做為語言，及 PRIMARYCARE 做為醫療專業。輸入初級醫療做為值，可讓您從下列醫療專業的來源音訊產生轉錄：

- 家庭醫學
- 內科
- 婦產科 (OB-GYN)
- 小兒科

您可以在口述和對話之間選擇您的音訊類型。為音訊檔案選擇口述，在此檔案中醫師提供了關於病患探訪或程序的報告。選擇對話的音訊檔案，這些檔案涉及醫生與病患之間的對話，或是醫生之間的對話。

若要存放轉錄任務的輸出，請選取您已建立的 Amazon S3 儲存貯體。如需 Amazon S3 值區的詳細資訊，請參閱 [入門 Amazon Simple Storage Service](#)。

以下是要在範例 JSON 中輸入的要求參數的最小數目。

```
{
  "MedicalTranscriptionJobName": "my-first-transcription-job",
  "LanguageCode": "en-US",
  "Media": {
    "MediaFileUri": "s3://path to your audio file"
  },
  "OutputBucketName": "your output bucket name",
  "Specialty": "PRIMARYCARE",
  "Type": "CONVERSATION"
}
```

Amazon Transcribe 醫療使您能夠生成替代轉錄。如需詳細資訊，請參閱 [生成替代轉錄](#)。

您也可以啟用喇叭分割或識別音訊中的頻道。如需詳細資訊，請參閱 [啟用喇叭分割](#) 及 [轉錄多聲道音訊](#)。

## 轉錄醫學術語和測量

Amazon Transcribe 醫療可以轉錄醫學術語和測量。Amazon Transcribe 口語術語的醫療輸出縮寫。例如，「血壓」會轉錄成 BP。您可以在此頁面的表格中找到 Medical 用於醫學術語和測量的慣例清單。Amazon Transcribe 「口述術語」欄位是指來源音訊中所說的術語。「輸出」欄位是指您在轉錄結果中看到的縮寫。

您可以在這裡看到來源音訊中所說的術語與轉錄輸出如何對應。

源音頻中所說的術語	輸出中使用的縮寫	範例輸出
攝氏度	C	病患的體溫為 37.4 C。
攝氏度	C	病患的體溫為 37.4 C。
華氏度	F	病患的體溫為 101 F。
公克	g	從病患抽取了 100 g 的質量。
公尺	m	病患身高 1.8 m。

源音頻中所說的術語	輸出中使用的縮寫	範例輸出
英尺	ft	病患身高 6 ft。
公斤	kg	病患的體重為 80 kg。
公斤	kg	病患的體重為 80 kg。
c c	cc	病患接受了 100 cc 的生理食鹽水。
立方公分	cc	病患接受了 100 cc 的生理食鹽水。
毫升	mL	病患排出了 100 mL 的尿液。
血壓	BP	病患的 BP 升高了
b p	BP	病患的 BP 升高了
收縮壓/舒張壓	X/Y	病患的 BP 是 120/80。
每分鐘心跳次數	BPM	病患心房顫動，心率为 160 BPM。
每分鐘心跳次數	BPM	病患心房顫動，心率为 160 BPM。
氧氣	O2	患者的 O2 飽和度 (含氧量) 為 98%。
二氧化碳	二氧化碳	患者需要呼吸輔助以增加二氧化碳。
術後	POSTOP	病患前來接受術後評估。
術後	POSTOP	病患前來接受術後評估。
電腦斷層掃描	CT Scan	顯示腦出血的病患，需要使用 CT Scan。

源音頻中所說的術語	輸出中使用的縮寫	範例輸出
脈搏 80	P 80	病患生命徵象為 P 80 , R 17 , ...
呼吸 17	R 17	病患生命徵象為 P 80 , R 17 , ...
進出	輸入/輸出	病患為 I/O 竇性心律
第五腰椎	L5	在 L4 和 L5 之間進行腰椎穿刺

## 轉錄號碼

Amazon Transcribe 醫療將數字轉錄為數字而不是單詞。例如，口述「一千兩百四十二」會轉錄為 1242。

根據以下規則轉錄數字。

規則	描述
將大於 10 的基數轉換成數字。	<ul style="list-style-type: none"> <li>「五十五」 &gt; 55</li> <li>「一百」 &gt; 100</li> <li>「一千零三十一」 &gt; 1031</li> <li>「一億二千三百四十五萬六千七百八十九」 &gt; 123,456,789</li> </ul>
當「百萬」或「億」後面沒有數字時，將基數後面接著「百萬」或「億」的數字轉換為數字後接著字詞。	<ul style="list-style-type: none"> <li>「一億」 &gt; 1 億</li> <li>「十億」 &gt; 10 億</li> <li>「二點三百萬」 &gt; 230 萬</li> </ul>
將大於 10 的序數轉換成數字。	<ul style="list-style-type: none"> <li>「四十三」 &gt; 43rd</li> <li>「二十六街」 &gt; 26th avenue</li> </ul>
將分數轉換成數字形式。	<ul style="list-style-type: none"> <li>「四分之一」 &gt; 1/4</li> <li>「十六分之三」 &gt; 3/16</li> <li>「一半」 &gt; 1/2</li> </ul>

規則	描述
	<ul style="list-style-type: none"> <li>「一百」 &gt; 1/100</li> </ul>
將一行中超過一位但不到十位的數值轉換成數字。	<ul style="list-style-type: none"> <li>「三四五」 &gt; 345</li> <li>「我的電話號碼是四二五五五五一二一二」 &gt; 4255551212</li> </ul>
小數點以「點」表示。	<ul style="list-style-type: none"> <li>「三百零三點五」 &gt; 303.5</li> <li>「三點二十三」 &gt; 3.23</li> <li>「零點四」 &gt; 0.4</li> <li>「點三」 &gt; 0.3</li> </ul>
將數字後的「百分比」轉換成百分比符號。	<ul style="list-style-type: none"> <li>「百分之二十三」 &gt; 23%</li> <li>「百分之二十三點四五」 &gt; 23.45%</li> </ul>
將數字後的字詞「元」、「美元」、「澳幣」、「AUD」或「USD」轉換成數字前的美元符號。	<ul style="list-style-type: none"> <li>「一元又十五分」 &gt; \$1.15</li> <li>「23 USD」 &gt; \$23</li> <li>「二十三澳幣」 &gt; \$23</li> </ul>
將字詞「磅」或「毫克」轉換為「lbs」或「mg」。	<ul style="list-style-type: none"> <li>「二十三磅」 &gt; 23 lbs</li> <li>「四十五毫克」 &gt; 45 mg</li> </ul>
將數字後的字詞「盧比」、「印度盧比」或「INR」轉換為數字前的盧比符號 (#)。	<ul style="list-style-type: none"> <li>「二十三盧比」 &gt; #23</li> <li>「五十盧比三十派士」 &gt; #50.30</li> </ul>
將時間轉換成數字。	<ul style="list-style-type: none"> <li>「早上七點東部標準時間」 &gt; 7 a.m. eastern standard time</li> <li>「下午一二點三十」 &gt; 12:30 p.m.</li> </ul>
將兩位數表示的年份合併為四位數。 僅適用於 20、21 和 22 世紀。	<ul style="list-style-type: none"> <li>「一九六二」 &gt; 1962</li> <li>「二零一二年」 &gt; the year is 2012</li> <li>「二零一九」 &gt; 2019</li> <li>「二一三零」 &gt; 2130</li> </ul>

規則	描述
轉換日期成為數字。	<ul style="list-style-type: none"> <li>「五月五日二零一二」 &gt; May 5th 2012</li> <li>「五月五日二零一二」 &gt; May 5 2012</li> <li>「二零一二年五月五日」 &gt; 5 May 2012</li> </ul>
使用「到」表達連續範圍數字。	<ul style="list-style-type: none"> <li>「二十三到三十七」 &gt; 23 to 37</li> </ul>

## 轉錄醫學對話

您可以使用「Amazon Transcribe 醫療」使用批次轉錄工作或即時串流，在臨床醫師與患者之間轉錄醫療對話。Batch 轉錄工作可讓您轉錄音訊檔案。為了確保 MedAmazon Transcribe ical 以最高的準確度產生轉錄結果，您必須在轉錄工作或流中指定臨床醫生的醫學專業。

您可以通過以下醫療專業轉錄臨床醫生和患者訪問：

- 心臟病學 — 僅適用於流式轉錄
- 神經病學 — 僅適用於流式轉錄
- 腫瘤學 — 僅適用於串流轉錄
- 初級保健 — 包括以下類型的醫療實踐：
  - 家庭醫學
  - 內科
  - 婦產科 (OB-GYN)
  - 小兒科
- 泌尿外科 — 僅適用於流式轉錄

您可以通過使用醫學自定義詞彙來提高轉錄準確性。有關醫學自定義詞彙如何工作的信息，請參閱[通過醫學定制詞彙提高轉錄準確性](#)。

根據預設，「Amazon Transcribe 醫療」會傳回具有最高信賴等級的轉錄。如果您想要將其設定為傳回替代轉錄，請參閱[生成替代轉錄](#)。

有關數字和醫療測量如何在轉錄輸出中顯示的信息，請參閱[轉錄號碼](#)和[轉錄醫學術語和測量](#)。

### 主題

- [轉錄醫療對話的音頻文件](#)

- [在實時流中轉錄醫療對話](#)
- [啟用喇叭分割](#)
- [轉錄多聲道音訊](#)

## 轉錄醫療對話的音頻文件

使用批量轉錄工作轉錄醫療對話的音頻文件。您可以使用它來轉錄臨床醫生與患者的對話。您可以在 [StartMedicalTranscriptionJobAPI](#) 或AWS Management Console.

當您使用 [StartMedicalTranscriptionJobAPI](#) 開始醫療轉錄工作時，請指定PRIMARYCARE為Specialty參數的值。

### AWS Management Console

轉錄臨床醫生與患者的對話 ( AWS Management Console )

若要使用AWS Management Console轉錄臨床醫生與病人對話，請建立轉錄工作，然後針對 [音訊] 輸入類型選擇 [交談]。

1. 登入 [AWS Management Console](#)。
2. 在導覽窗格中的Amazon Transcribe醫療下，選擇轉錄工作。
3. 選擇 Create job (建立任務)。
4. 在 [指定 Job 詳細資訊] 頁面的 [工作設定] 下，指定下列項目。
  - a. 名稱 — 轉錄工作的名稱。
  - b. 音訊輸入類型 — 對話
5. 對於其餘欄Amazon S3位，請指定音訊檔案的位置以及轉錄工作輸出的儲存位置。
6. 選擇 Next (下一步)。
7. 選擇 Create (建立)。

### API

使用批次轉錄工作 (API) 轉錄醫學對話

- 對於 [StartMedicalTranscriptionJobAPI](#)，請指定以下內容。
  - a. 對於MedicalTranscriptionJobName，指定在您的AWS 帳戶。



- b. 對於LanguageCode，請指定與音訊檔案中所說語言相對應的語言代碼，以及字彙篩選器的語言。
- c. 針對Media物件的MediaFileUri參數，指定您要轉錄的音訊檔案名稱。
- d. 對於Specialty，將臨床醫生在音頻文件中說話的醫學專業指定為PRIMARYCARE。
- e. 在Type，請指定 CONVERSATION。
- f. 對於OutputBucketName，指定要Amazon S3儲存轉錄結果的值區。

以下是一個示例請求，該請求使用AWS SDK for Python (Boto3)轉錄PRIMARYCARE專科醫生和患者的臨床醫生的醫學對話。

```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
job_name = "my-first-med-transcription-job"
job_uri = "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-audio-file.flac"
transcribe.start_medical_transcription_job(
    MedicalTranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'DOC-EXAMPLE-BUCKET',
    OutputKey = 'output-files/',
    LanguageCode = 'en-US',
    Specialty = 'PRIMARYCARE',
    Type = 'CONVERSATION'
)

while True:
    status = transcribe.get_medical_transcription_job(MedicalTranscriptionJobName =
job_name)
    if status['MedicalTranscriptionJob']['TranscriptionJobStatus'] in ['COMPLETED',
'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
print(status)
```

以下示例代碼顯示了臨床醫生患者對話的轉錄結果。

```
{
  "jobName": "conversation-medical-transcription-job",
  "accountId": "111122223333",
  "results": {
    "transcripts": [
      {
        "transcript": "... come for a follow up visit today..."
      }
    ],
    "items": [
      {
        ...
        "start_time": "4.85",
        "end_time": "5.12",
        "alternatives": [
          {
            "confidence": "1.0",
            "content": "come"
          }
        ],
        "type": "pronunciation"
      },
      {
        "start_time": "5.12",
        "end_time": "5.29",
        "alternatives": [
          {
            "confidence": "1.0",
            "content": "for"
          }
        ],
        "type": "pronunciation"
      },
      {
        "start_time": "5.29",
        "end_time": "5.33",
        "alternatives": [
          {
            "confidence": "0.9955",
            "content": "a"
          }
        ]
      }
    ]
  }
}
```

```
    ],
    "type": "pronunciation"
  },
  {
    "start_time": "5.33",
    "end_time": "5.66",
    "alternatives": [
      {
        "confidence": "0.9754",
        "content": "follow"
      }
    ],
    "type": "pronunciation"
  },
  {
    "start_time": "5.66",
    "end_time": "5.75",
    "alternatives": [
      {
        "confidence": "0.9754",
        "content": "up"
      }
    ],
    "type": "pronunciation"
  },
  {
    "start_time": "5.75",
    "end_time": "6.02",
    "alternatives": [
      {
        "confidence": "1.0",
        "content": "visit"
      }
    ]
  }
  ...
},
"status": "COMPLETED"
}
```

## AWS CLI

### 使用批次轉錄工作轉錄醫學對話 (AWS CLI)

- 執行下列程式碼。

```
aws transcribe start-medical-transcription-job \  
--region us-west-2 \  
--cli-input-json file://example-start-command.json
```

下列程式碼顯示的*example-start-command.json*內容：

```
{  
  "MedicalTranscriptionJobName": "my-first-med-transcription-job",  
  "Media": {  
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-audio-file.flac"  
  },  
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",  
  "OutputKey": "my-output-files/",  
  "LanguageCode": "en-US",  
  "Specialty": "PRIMARYCARE",  
  "Type": "CONVERSATION"  
}
```

## 在實時流中轉錄醫療對話

您可以使用 HTTP/2 或 [WebSocket](#) 協議轉錄醫療對話的音頻流。如需如何使用通訊 WebSocket 協定啟動串流的資訊，請參閱 [設定串 WebSocket 流](#)。若要啟動 HTTP/2 串流，請使用 [StartMedicalStreamTranscriptionAPI](#)。

您可以在以下醫療專業中轉錄流音頻：

- 心脏病学
- 神經病學
- 腫瘤科
- 初級保健

- 泌尿外科

每個醫療專業包括多種類型的程序和預約。因此，臨床醫生決定許多不同類型的筆記。使用下列範例做為指引，協助您指定 WebSocket 要求的 specialty URI 參數值或 [StartMedicalStreamTranscription](#) API 的 Specialty 參數：

- 對於電生理或超聲心動諮詢，請選擇 CARDIOLOGY。
- 對於內科腫瘤學、外科腫瘤學或放射腫瘤學諮詢，請選擇 ONCOLOGY。
- 對於向中風（短暫性缺血發作或腦血管發作）的患者提供諮詢的醫生，請選擇 NEUROLOGY。
- 有關尿失禁的諮詢，請選擇 UROLOGY。
- 對於每年檢查或緊急護理訪問，請選擇 PRIMARYCARE。
- 對於住院者住院探訪，請選擇 PRIMARYCARE。
- 有關生育，輸卵管結紮，IUD 插入或墮胎的諮詢，請選擇 PRIMARYCARE。

## AWS Management Console

### 轉錄流式醫療對話（AWS Management Console）

若要使用 AWS Management Console 即時串流轉錄臨床醫師與病患對話，請選擇轉錄醫療對話、開始串流，然後開始對著麥克風說話的選項。

1. 登入 [AWS Management Console](#)。
2. 在導覽窗格中的 Amazon Transcribe 醫療下，選擇即時轉錄。
3. 選擇 [交談]。
4. 對於醫療專業，請選擇臨床醫生的專業。
5. 選擇 Start streaming（開始串流）。
6. 對著麥克風說話。

### 在 HTTP/2 流中轉錄醫療對話

以下是 HTTP/2 請求的參數的語法。

要轉錄醫療對話的 HTTP/2 流，請使用 [StartMedicalStreamTranscription](#) API 並指定以下內容：

- LanguageCode— 語言代碼。有效值為 en-US

- MediaEncoding— 用於輸入音訊的編碼。有效值為 pcm、ogg-opus 和 flac。
- Specialty— 醫療專業人員的專業。
- Type – CONVERSATION

若要提高即時串流中特定詞彙的轉錄準確性，請使用自訂字彙。若要啟用自訂字彙，請將VocabularyName參數值設定為您要使用的自訂字彙的名稱。如需詳細資訊，請參閱[通過醫學定制詞彙提高轉錄準確性](#)。

若要標示來自不同揚聲器的語音，請將ShowSpeakerLabel參數設定為true。如需詳細資訊，請參閱[啟用喇叭分割](#)。

如需設定 HTTP/2 串流以轉錄醫療對話的詳細資訊，請參閱[設定 HTTP/2 串流](#)。

在 WebSocket 流中轉錄醫學對話

您可以使用 WebSocket 請求轉錄醫療對話。當您提出 WebSocket 要求時，您會建立預先簽署的 URI。此 URI 包含在您的應用程式和Amazon Transcribe醫療之間建立音訊串流時所需的資訊。如需建立 WebSocket 請求的詳細資訊，請參閱[設定串 WebSocket 流](#)。

請使用下列範本來建立您的預先簽署 URI。

```
GET wss://transcribestreaming.us-west-2.amazonaws.com:8443/medical-stream-  
transcription-websocket  
?language-code=languageCode  
&X-Amz-Algorithm=AWS4-HMAC-SHA256  
&X-Amz-Credential=AKIAIOSFODNN7EXAMPLE%2F20220208%2Fus-  
west-2%2Ftranscribe%2Faws4_request  
&X-Amz-Date=20220208T235959Z  
&X-Amz-Expires=300  
&X-Amz-Security-Token=security-token  
&X-Amz-Signature=Signature Version 4 signature  
&X-Amz-SignedHeaders=host  
&media-encoding=flac  
&sample-rate=16000  
&session-id=sessionId  
&specialty=medicalSpecialty  
&type=CONVERSATION  
&vocabulary-name=vocabularyName  
&show-speaker-label=boolean
```

若要提高即時串流中特定詞彙的轉錄準確性，請使用自訂字彙。若要啟用自訂字彙，請將 `vocabulary-name` 將的值設定為您要使用的自訂字彙的名稱。如需詳細資訊，請參閱 [通過醫學定制詞彙提高轉錄準確性](#)。

若要標示來自不同揚聲器的語音，請將中的 `show-speaker-label` 參數設定為 `true`。如需詳細資訊，請參閱 [啟用喇叭分割](#)。

如需建立預先簽署 URI 的詳細資訊，請參閱 [設定串 WebSocket 流](#)。

## 啟用喇叭分割

若要在「Amazon Transcribe 醫療」中啟用喇叭分割功能，請使用喇叭日記。這使您可以查看患者說了什麼以及臨床醫生在轉錄輸出中所說的內容。

當您啟用揚聲器日記功能時，「Amazon Transcribe 醫療」會為每個揚聲器的語音標示為每個揚聲器的唯一識別碼。說話是一種語音單位，通常通過沉默與其他話語分開。在批量轉錄中，臨床醫生的話語可以收到患者可以收到標籤的標籤 `spk_0` 和說話 `spk_1`。

如果一個揚聲器的話語與另一位發言者的話語重疊，Amazon Transcribe 醫療會按照開始時間在轉錄中命令他們。輸入音訊中重疊的表達用語在轉錄輸出中不會重疊。

當您使用批次轉錄工作或即時串流轉錄音訊檔案時，您可以啟用喇叭日記功能。

### 主題

- [在批次轉錄中啟用喇叭分割](#)
- [在即時串流中啟用喇叭分割](#)

## 在批次轉錄中啟用喇叭分割

您可以在批次轉錄工作中使用 [StartMedicalTranscriptionJob API](#) 或 AWS Management Console。這使您可以在臨床醫生和患者對話中對每個發言者進行分區，並確定誰在轉錄輸出中說了什麼。

### AWS Management Console

若要在 AWS Management Console 轉錄工作中啟用揚聲器日記功能，請先啟用音訊識別，然後再啟用喇叭分割。

1. 登入 [AWS Management Console](#)。
2. 在導覽窗格中的 Amazon Transcribe 醫療下，選擇轉錄工作。

3. 選擇 **Create job** (建立任務)。
4. 在 [指定工作詳細資料] 頁面上，提供轉錄工作的相關資訊。
5. 選擇 **下一步**。
6. 啟用音訊識別。
7. 對於音訊識別類型，請選擇喇叭分割。
8. 針對最大喇叭數目，請輸入您認為在音訊檔案中說話的最大喇叭數目。
9. 選擇 **建立**。

## API

### 使用批次轉錄工作 (API) 啟用喇叭分割

- 對於 [StartMedicalTranscriptionJobAPI](#)，請指定以下內容。
  - a. 對於 `MedicalTranscriptionJobName`，指定在您的中唯一的名稱AWS 帳戶。
  - b. 對於 `LanguageCode`，指定與音訊檔案中所說語言相對應的語言代碼。
  - c. 針對 `Media` 物件的 `MediaFileUri` 參數，指定您要轉錄的音訊檔案名稱。
  - d. 對於 `Specialty`，指定臨床醫生在音頻文件中說話的醫療專業。
  - e. 在 `Type`，請指定 `CONVERSATION`。
  - f. 對於 `OutputBucketName`，指定要Amazon S3儲存轉錄結果的值區。
  - g. 對於 `Settings` 物件，請指定下列項目。
    - i. `ShowSpeakerLabels - true`。
    - ii. `MaxSpeakerLabels`— 介於 2 和 10 之間的整數，表示您認為在音訊中說話的喇叭數量。

以下請求使用啟用AWS SDK for Python (Boto3)了啟用揚聲器分區的初級保健臨床醫生患者對話的批次轉錄工作。

```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
job_name = "my-first-transcription-job"
job_uri = "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
```



```
transcribe.start_medical_transcription_job(  
    MedicalTranscriptionJobName = job_name,  
    Media={  
        'MediaFileUri': job_uri  
    },  
    OutputBucketName = 'DOC-EXAMPLE-BUCKET',  
    OutputKey = 'my-output-files/',  
    LanguageCode = 'en-US',  
    Specialty = 'PRIMARYCARE',  
    Type = 'CONVERSATION',  
    OutputBucketName = 'DOC-EXAMPLE-BUCKET',  
    Settings = {'ShowSpeakerLabels': True,  
                'MaxSpeakerLabels': 2  
            }  
)  
while True:  
    status = transcribe.get_medical_transcription_job(MedicalTranscriptionJobName =  
    job_name)  
    if status['MedicalTranscriptionJob']['TranscriptionJobStatus'] in ['COMPLETED',  
    'FAILED']:  
        break  
    print("Not ready yet...")  
    time.sleep(5)  
print(status)
```

下列範例程式碼會顯示啟用喇叭分割的轉錄工作的轉錄結果。

```
{  
    "jobName": "job ID",  
    "accountId": "111122223333",  
    "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"
        },
        {
            "alternatives": [
                {

```

```
        "content": "."
      }
    ],
    "type": "punctuation"
  }
]
},
"status": "COMPLETED"
}
```

## AWS CLI

轉錄執行初級保健的臨床醫生與患者之間對話的音頻文件 ( AWS CLI )

- 執行下列程式碼。

```
aws transcribe start-transcription-job \  
--region us-west-2 \  
--cli-input-json file://example-start-command.json
```

下列程式碼顯示的*example-start-command.json*內容：

```
{  
  "MedicalTranscriptionJobName": "my-first-med-transcription-job",  
  "Media": {  
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-audio-file.flac"  
  },  
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",  
  "OutputKey": "my-output-files/",  
  "LanguageCode": "en-US",  
  "Specialty": "PRIMARYCARE",  
  "Type": "CONVERSATION",  
  "Settings": {  
    "ShowSpeakerLabels": true,  
    "MaxSpeakerLabels": 2  
  }  
}
```

```
}
```

## 在即時串流中啟用喇叭分割

若要分割喇叭並在即時串流中標示其語音，請使用AWS Management Console或串流要求。喇叭分割最適合串流中的兩個和五個喇叭。雖然 Amazon Transcribe 可以在一個串流中分割超過五個喇叭，但如果超過該數目，分割區的準確度會降低。

若要啟動 HTTP/2 要求，請使用 [StartMedicalStreamTranscriptionAPI](#)。若要啟動要 WebSocket 求，請使用預先簽署的 URI。URI 包含設定應用程式與Amazon Transcribe醫療設定雙向通訊所需的資訊。

### 在麥克風語音中啟用喇叭分割 (AWS Management Console)

您可以使用AWS Management Console來啟動臨床醫師與患者對話的即時串流，或即時對您的麥克風說出聽寫。

1. 登入 [AWS Management Console](#)。
2. 在功能窗格中，針對 [Amazon Transcribe醫療] 選擇 [即時轉錄]。
3. 在 [音訊輸入類型] 中，選擇您要轉錄的醫療語音類型。
4. 如需其他設定，請選擇「喇叭分割」。
5. 選擇開始串流以開始轉錄您的即時音訊。
6. 對著麥克風說話。

### 在 HTTP/2 串流中啟用喇叭分割

若要在醫療交談的 HTTP/2 串流中啟用喇叭分割，請使用 [StartMedicalStreamTranscriptionAPI](#) 並指定下列項目：

- 對於LanguageCode，指定對應於串流中語言的語言代碼。有效值為 en-US。
- 對於MediaSampleHertz，指定音訊的取樣率。
- 對於Specialty，指定提供者的醫療專業。
- ShowSpeakerLabel – true

如需設定 HTTP/2 串流以轉錄醫療對話的詳細資訊，請參閱[設定 HTTP/2 串流](#)。

## 在 WebSocket 要求中啟用發言者分割

若要使用 API 分割 WebSocket 串流中的喇叭，請使用下列格式建立預先簽署的 URI，以啟動 WebSocket 要求並將其設定 `show-speaker-label` 為 `true`。

```
GET wss://transcribestreaming.us-west-2.amazonaws.com:8443/medical-stream-
transcription-websocket
?language-code=languageCode
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIAIOSFODNN7EXAMPLE%2F20220208%2Fus-
west-2%2Ftranscribe%2Faws4_request
&X-Amz-Date=20220208T235959Z
&X-Amz-Expires=300
&X-Amz-Security-Token=security-token
&X-Amz-Signature=Signature Version 4 signature
&X-Amz-SignedHeaders=host
&media-encoding=flac
&sample-rate=16000
&session-id=sessionId
&specialty=medicalSpecialty
&type=CONVERSATION
&vocabulary-name=vocabularyName
&show-speaker-label=boolean
```

下列程式碼會顯示串流要求的截斷範例回應。

```
{
  "Transcript": {
    "Results": [
      {
        "Alternatives": [
          {
            "Items": [
              {
                "Confidence": 0.97,
                "Content": "From",
                "EndTime": 18.98,
                "Speaker": "0",
                "StartTime": 18.74,
                "Type": "pronunciation",
                "VocabularyFilterMatch": false
              }
            ]
          }
        ]
      }
    ]
  }
}
```

```
    },  
    {  
      "Confidence": 1,  
      "Content": "the",  
      "EndTime": 19.31,  
      "Speaker": "0",  
      "StartTime": 19,  
      "Type": "pronunciation",  
      "VocabularyFilterMatch": false  
    },  
    {  
      "Confidence": 1,  
      "Content": "last",  
      "EndTime": 19.86,  
      "Speaker": "0",  
      "StartTime": 19.32,  
      "Type": "pronunciation",  
      "VocabularyFilterMatch": false  
    },  
    ...  
    {  
      "Confidence": 1,  
      "Content": "chronic",  
      "EndTime": 22.55,  
      "Speaker": "0",  
      "StartTime": 21.97,  
      "Type": "pronunciation",  
      "VocabularyFilterMatch": false  
    },  
    ...  
    "Confidence": 1,  
    "Content": "fatigue",  
    "EndTime": 24.42,  
    "Speaker": "0",  
    "StartTime": 23.95,  
    "Type": "pronunciation",  
    "VocabularyFilterMatch": false  
  },  
  {  
    "EndTime": 25.22,  
    "StartTime": 25.22,  
    "Type": "speaker-change",  
    "VocabularyFilterMatch": false  
  },  
}
```

```

        {
            "Confidence": 0.99,
            "Content": "True",
            "EndTime": 25.63,
            "Speaker": "1",
            "StartTime": 25.22,
            "Type": "pronunciation",
            "VocabularyFilterMatch": false
        },
        {
            "Content": ".",
            "EndTime": 25.63,
            "StartTime": 25.63,
            "Type": "punctuation",
            "VocabularyFilterMatch": false
        }
    ],
    "Transcript": "From the last note she still has mild sleep deprivation and
chronic fatigue True."
}
],
"EndTime": 25.63,
"IsPartial": false,
"ResultId": "XXXXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
"StartTime": 18.74
}
]
}
}

```

Amazon Transcribe「醫療」會根據自然語音片段 (例如揚聲器變更或音訊暫停)，中斷傳入的音訊串流。系統會漸進式地將轉錄傳回您的應用程式，每次回應皆會包含更多轉錄的語音，直到整個區段轉錄完成。上述程式碼是完全轉錄語音區段的截斷範例。喇叭標籤只會針對完全轉錄的區段顯示。

下列清單顯示串流轉錄輸出中物件和參數的組織。

## Transcript

每個語音片段都有自己的Transcript物件。

## Results

每個Transcript物件都有自己的Results物件。此物件包含isPartial欄位。當其值為false時，傳回的結果會針對整個語音區段。

## Alternatives

每個Results物件都有一個Alternatives物件。

## Items

每個Alternatives物件都有自己的Items物件，其中包含轉錄輸出中每個字詞和標點符號的相關資訊。當您啟用說話者分割時，每個單字都有一個Speaker標籤，用於完全轉錄的語音片段。Amazon TranscribeMedical 使用此標籤為串流中的每個揚聲器指派唯一的整數。具有值的Type參數speaker-change表示一個人已經停止說話，另一個人即將開始。

## Transcript

每個 Items 物件都包含一個轉錄的語音區段做為Transcript欄位的值。

如需有關 WebSocket 請求的詳細資訊，請參閱[設定串 WebSocket 流](#)。

## 轉錄多聲道音訊

如果您有具有多個頻道的音訊檔案或串流，您可以使用頻道識別來轉錄每個頻道的語音。Amazon Transcribe醫療單獨從每個通道轉錄語音。它將每個通道的單獨轉錄組合到單個轉錄輸出中。

使用頻道識別來識別音訊中不同的頻道，並從每個頻道轉錄語音。在呼叫者和代理程式案例等情況下啟用此功能。使用此功能可將來電者與執行藥物安全監控的聯絡中心的錄音或串流中的代理區分開來電者。

您可以為批次處理和即時串流啟用通道識別。下列清單說明如何針對每個方法啟用此功能。

- Batch 轉錄 — AWS Management Console 和 [StartMedicalTranscriptionJobAPI](#)
- 串流轉錄 — WebSocket 串流和 [StartMedicalStreamTranscriptionAPI](#)

## 轉錄多聲道音訊檔案

當您轉錄音訊檔案時，「Amazon Transcribe醫療」會傳回每個頻道的項目清單。項目是轉錄的單字或標點符號。每個單詞都有開始時間和結束時間。如果某個頻道上的使用者透過不同頻道上的某個人說話，則每個頻道項目的開始時間和結束時間會重疊，而個人彼此交談時間也會重疊。



默認情況下，您可以轉錄具有兩個通道的音頻文件。如果您需要轉錄具有兩個以上通道的檔案，可以要求提高配額。如需要請求增加配額的相關資訊，請參閱[AWS 服務配額](#)。

若要在批次轉錄工作中轉錄多聲道音訊，請使用AWS Management Console或[StartMedicalTranscriptionJobAPI](#)。

## AWS Management Console

若要在批次轉錄工作中使用啟用頻道識別，請先啟用音訊識別，然後啟用頻道識別。AWS Management Console頻道識別是中音訊識別的子集AWS Management Console。

1. 登入 [AWS Management Console](#)。
2. 在導覽窗格中的Amazon Transcribe醫療下，選擇轉錄工作。
3. 選擇 Create job (建立任務)。
4. 在 [指定工作詳細資料] 頁面上，提供轉錄工作的相關資訊。
5. 選擇 下一步。
6. 啟用音訊識別。
7. 對於音訊識別類型，請選擇「頻道識別」。
8. 選擇 建立。

## API

### 轉錄多聲道音訊檔案 (API)

- 對於 [StartMedicalTranscriptionJobAPI](#)，請指定以下內容。
  - a. 對於TranscriptionJobName，指定您的專屬名稱AWS 帳戶。
  - b. 對於LanguageCode，指定與音訊檔案中所說語言相對應的語言代碼。有效值為 en-US。
  - c. 針Media對物件的MediaFileUri參數，指定您要轉錄之媒體檔案的名稱。
  - d. 對於Settings物件，ChannelIdentification將設定為true。

以下是使用的範例要求AWS SDK for Python (Boto3)。

```
from __future__ import print_function
import time
import boto3
```

```
transcribe = boto3.client('transcribe', 'us-west-2')
job_name = "my-first-transcription-job"
job_name = "my-first-med-transcription-job"
job_uri = "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
transcribe.start_medical_transcription_job(
    MedicalTranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'DOC-EXAMPLE-BUCKET',
    OutputKey = 'output-files/',
    LanguageCode = 'en-US',
    Specialty = 'PRIMARYCARE',
    Type = 'CONVERSATION',
    Settings = {
        'ChannelIdentification': True
    }
)
while True:
    status = transcribe.get_transcription_job(MedicalTranscriptionJobName = job_name)
    if status['MedicalTranscriptionJob']['TranscriptionJobStatus'] in ['COMPLETED',
        'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
print(status)
```

## AWS CLI

### 使用批次轉錄工作轉錄多聲道音訊檔案 (AWS CLI)

- 執行下列程式碼。

```
aws transcribe start-medical-transcription-job \
--region us-west-2 \
--cli-input-json file://example-start-command.json
```

以下是的程式碼example-start-command.json。

```
{
  "MedicalTranscriptionJobName": "my-first-med-transcription-job",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-audio-
file.flac"
  },
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",
  "OutputKey": "my-output-files/",
  "LanguageCode": "en-US",
  "Specialty": "PRIMARYCARE",
  "Type": "CONVERSATION",

  "Settings":{
    "ChannelIdentification": true
  }
}
```

下列程式碼會顯示在兩個通道上具有交談的音訊檔案的轉錄輸出。

```
{
  "jobName": "job id",
  "accountId": "111122223333",
  "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"
},
  ...
]
},
{
  "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"
    },
    ...
]
}
```

## 轉錄多聲道音訊串流

您可以使用 [StartMedicalStreamTranscription](#) API 從 HTTP/2 或 WebSocket 串流中的不同頻道轉錄音訊。

默認情況下，您可以使用兩個通道轉錄流。如果您需要轉錄具有兩個以上通道的串流，可以要求提高配額。如需請求增加配額的相關資訊，請參閱 [AWS 服務配額](#)。

### 在 HTTP/2 串流中轉錄多聲道音訊

若要轉錄 HTTP/2 串流中的多聲道音訊，請使用 [StartMedicalStreamTranscription](#) API 並指定下列項目：

- `LanguageCode`— 音訊的語言代碼。有效值為 `en-US`。
- `MediaEncoding`— 音訊的編碼。有效值為 `ogg-opus`、`flac` 和 `pcm`。
- `EnableChannelIdentification` - `true`

- `NumberOfChannels`— 您的串流音訊中頻道的數量。

如需設定 HTTP/2 串流以轉錄醫療對話的詳細資訊，請參閱[設定 HTTP/2 串流](#)。

### 轉錄 WebSocket 串流中的多聲道音訊

若要分割 WebSocket 串流中的喇叭，請使用下列格式建立預先簽署的 URI 並啟動 WebSocket 要求。指定 `enable-channel-identification` 為 `true` 和您串流中的頻道數量 `number-of-channels`。預先簽署的 URI 包含在您的應用程式和 Amazon Transcribe 醫療之間建立雙向通訊時所需的資訊。

```
GET wss://transcribestreaming.us-west-2.amazonaws.com:8443/medical-stream-
transcription-websocket
?language-code=languageCode
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIAIOSFODNN7EXAMPLE%2F20220208%2Fus-
west-2%2Ftranscribe%2Faws4_request
&X-Amz-Date=20220208T235959Z
&X-Amz-Expires=300
&X-Amz-Security-Token=security-token
&X-Amz-Signature=Signature Version 4 signature
&X-Amz-SignedHeaders=host
&media-encoding=flac
&sample-rate=16000
&session-id=sessionId
&enable-channel-identification=true
&number-of-channels=2
```

您可以在 [API 參考中找到參數定義](#)；所有 AWS API 作業的共用參數會列在「[一般參數](#)」區段中。

如需有關 WebSocket 請求的詳細資訊，請參閱[設定串 WebSocket 流](#)。

### 多頻道串流輸出

HTTP/2 和 WebSocket 請求的流轉錄的輸出是相同的。以下為範例輸出。

```
{
  "resultId": "XXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
  "startTime": 0.11,
  "endTime": 0.66,
  "isPartial": false,
```

```
"alternatives": [
  {
    "transcript": "Left.",
    "items": [
      {
        "startTime": 0.11,
        "endTime": 0.45,
        "type": "pronunciation",
        "content": "Left",
        "vocabularyFilterMatch": false
      },
      {
        "startTime": 0.45,
        "endTime": 0.45,
        "type": "punctuation",
        "content": ".",
        "vocabularyFilterMatch": false
      }
    ]
  }
],
"channelId": "ch_0"
}
```

對於每個語音區段，都有一個channelId標誌，指出該語音所屬的通道。

## 轉錄醫療聽寫

您可以使用「Amazon Transcribe醫療」使用批次轉錄工作或即時串流來轉錄臨床醫生指示的醫療筆記。Batch 轉錄工作可讓您轉錄音訊檔案。您可以在轉錄工作或串流中指定臨床醫師的Amazon Transcribe醫療專業，以確保 Medical 以最高的準確度產生轉錄結果。

您可以轉錄以下專業的醫療聽寫：

- 心臟科 — 僅適用於流式轉錄
- 神經病學 — 僅適用於流式轉錄
- 腫瘤學 — 僅適用於串流轉錄
- 初級保健 — 包括以下類型的醫療實踐：
  - 家庭醫學
  - 內科

- 婦產科 (OB-GYN)
- 小兒科
- 放射學 — 僅適用於串流轉錄
- 泌尿外科 — 僅適用於流式轉錄

您可以通過使用自定義詞彙提高轉錄準確性。有關醫學自定義詞彙如何工作的信息，請參閱[通過醫學定制詞彙提高轉錄準確性](#)。

根據預設，「Amazon Transcribe 醫療」會傳回具有最高信賴等級的轉錄。如果您想要將其設定為傳回替代轉錄，請參閱[生成替代轉錄](#)。

有關數字和醫療測量如何在轉錄輸出中顯示的信息，請參閱[轉錄號碼](#)和[轉錄醫學術語和測量](#)。

## 主題

- [轉錄醫療聽寫的音頻文件](#)
- [在實時流中轉錄醫療聽寫](#)

## 轉錄醫療聽寫的音頻文件

使用批量轉錄工作轉錄醫療對話的音頻文件。您可以使用它來轉錄臨床醫生與患者的對話。您可以在[StartMedicalTranscriptionJob](#) API 或 AWS Management Console。

當您使用 [StartMedicalTranscriptionJob](#) API 開始醫療轉錄工作時，您會指定 PRIMARYCARE 為 Specialty 參數的值。

### AWS Management Console

#### 轉錄臨床醫生與患者的對話 ( AWS Management Console )

若要使用 AWS Management Console 轉錄臨床醫生與病人對話，請建立轉錄工作，然後針對 [音訊] 輸入類型選擇 [交談]。

1. 登入 [AWS Management Console](#)。
2. 在導覽窗格中的 Amazon Transcribe 醫療下，選擇轉錄工作。
3. 選擇 Create job (建立任務)。
4. 在 [指定 Job 詳細資訊] 頁面的 [工作設定] 下，指定下列項目。
  - a. 名稱 — 轉錄工作的名稱。



- b. 音訊輸入類型 — 聽寫
5. 對於其餘欄Amazon S3位，請指定音訊檔案的位置，以及要儲存轉錄工作輸出的位置。
6. 選擇 Next (下一步)。
7. 選擇 Create (建立)。

## API

### 使用批次轉錄工作 (API) 轉錄醫學對話

- 對於 [StartMedicalTranscriptionJobAPI](#)，請指定以下內容。
  - a. 對於MedicalTranscriptionJobName，指定在您的AWS 帳戶。
  - b. 對於LanguageCode，請指定與音訊檔案中所說語言相對應的語言代碼，以及字彙篩選器的語言。
  - c. 在Media物件的MediaFileUri參數中，指定您要轉錄的音訊檔案名稱。
  - d. 對於Specialty，指定臨床醫生在音頻文件中說話的醫療專業。
  - e. 在 Type，請指定 DICTATION。
  - f. 對於OutputBucketName，指定Amazon S3儲存轉錄結果的值區。

以下是使用轉錄PRIMARYCARE專業臨床醫師的醫療聽寫的示例請求。AWS SDK for Python (Boto3)

```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe')
job_name = "my-first-med-transcription-job"
job_uri = "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-audio-file.flac"
transcribe.start_medical_transcription_job(
    MedicalTranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'DOC-EXAMPLE-BUCKET',
    OutputKey = 'my-output-files/',
    LanguageCode = 'en-US',
    Specialty = 'PRIMARYCARE',
```

```
        Type = 'DICTATION'
    )
while True:
    status = transcribe.get_medical_transcription_job(MedicalTranscriptionJobName =
    job_name)
    if status['MedicalTranscriptionJob']['TranscriptionJobStatus'] in ['COMPLETED',
    'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
print(status)
```

下列範例程式碼顯示醫療聽寫的轉錄結果。

```
{
  "jobName": "dictation-medical-transcription-job",
  "accountId": "111122223333",
  "results": {
    "transcripts": [
      {
        "transcript": "... came for a follow up visit today..."
      }
    ],
    "items": [
      {
        ...
        "start_time": "4.85",
        "end_time": "5.12",
        "alternatives": [
          {
            "confidence": "1.0",
            "content": "came"
          }
        ],
        "type": "pronunciation"
      },
      {
        "start_time": "5.12",
        "end_time": "5.29",
        "alternatives": [
```

```
        {
          "confidence": "1.0",
          "content": "for"
        }
      ],
      "type": "pronunciation"
    },
    {
      "start_time": "5.29",
      "end_time": "5.33",
      "alternatives": [
        {
          "confidence": "0.9955",
          "content": "a"
        }
      ],
      "type": "pronunciation"
    },
    {
      "start_time": "5.33",
      "end_time": "5.66",
      "alternatives": [
        {
          "confidence": "0.9754",
          "content": "follow"
        }
      ],
      "type": "pronunciation"
    },
    {
      "start_time": "5.66",
      "end_time": "5.75",
      "alternatives": [
        {
          "confidence": "0.9754",
          "content": "up"
        }
      ],
      "type": "pronunciation"
    },
    {
      "start_time": "5.75",
      "end_time": "6.02",
      "alternatives": [
```

```
        {
            "confidence": "1.0",
            "content": "visit"
        }
    ]
    ...
},
"status": "COMPLETED"
}
```

## AWS CLI

### 在批次轉錄工作中啟用喇叭分割 (AWS CLI)

- 執行下列程式碼。

```
aws transcribe start-medical-transcription-job \
--region us-west-2 \
--cli-input-json file://example-start-command.json
```

下列程式碼顯示的內容 *example-start-command.json*。

```
{
  "MedicalTranscriptionJobName": "my-first-med-transcription-job",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-audio-file.flac"
  },
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",
  "OutputKey": "my-output-files/",
  "LanguageCode": "en-US",
  "Specialty": "PRIMARYCARE",
  "Type": "DICTATION"
}
```

## 在實時流中轉錄醫療聽寫

使用 WebSocket 串流將醫療聽寫轉錄為音訊串流。您也可以使用AWS Management Console將您或其他人直接對麥克風說話的語音轉錄。

對於 HTTP/2 或 WebSocket 流，您可以轉錄以下醫療專業的音頻：

- 心脏病学
- 腫瘤科
- 神經病學
- 初級保健
- 放射學
- 泌尿外科

每個醫療專業包括多種類型的程序和預約。因此，臨床醫生決定許多不同類型的筆記。使用下列範例做為指引，協助您指定 WebSocket 要求的specialty URI 參數值或 [StartMedicalStreamTranscriptionAPI](#) 的Specialty參數：

- 對於電生理或超聲心動圖程序後的聽寫，請選擇CARDIOLOGY。
- 對於外科腫瘤學或放射腫瘤學程序後的聽寫，請選擇ONCOLOGY。
- 對於醫生口述指示腦炎診斷的筆記，請選擇NEUROLOGY。
- 對於程序筆記的聽寫以分解膀胱結石，請選擇UROLOGY。
- 對於內科諮詢後臨床醫生筆記的聽寫，請選擇PRIMARYCARE。
- 對於醫生傳達 CT 掃描，PET 掃描，MRI 或 X 光照相的結果的聽寫，請選擇RADIOLOGY。
- 對於婦科諮詢後醫生筆記的聽寫，請選擇PRIMARYCARE。

若要提高即時串流中特定詞彙的轉錄準確性，請使用自訂字彙。若要啟用自訂字彙，請vocabulary-name將的值設定為您要使用的自訂字彙的名稱。

使用AWS Management Console

若要使用AWS Management Console來轉錄醫療聽寫的串流音訊，請選擇轉錄醫療聽寫、開始串流，然後開始對著麥克風說話的選項。

## 轉錄醫療聽寫的串流音訊 ( AWS Management Console )

1. 登入 [AWS Management Console](#)。
2. 在導覽窗格的Amazon Transcribe醫療下，選擇即時轉錄。
3. 選擇「聽寫」。
4. 對於醫療專業，請選擇在流中講話的臨床醫生的醫學專業。
5. 選擇 Start streaming (開始串流)。
6. 對著麥克風說話。

### 在 HTTP/2 串流中轉錄聽寫

要轉錄醫療聽寫的 HTTP/2 流，請使用 [StartMedicalStreamTranscription](#) API 並指定以下內容：

- LanguageCode— 語言代碼。有效值為 en-US
- MediaEncoding— 用於輸入音訊的編碼。有效值為 pcm、ogg-opus 和 flac。
- Specialty— 醫療專業人員的專業。
- Type – DICTATION

如需設定 HTTP/2 串流以轉錄醫療聽寫的詳細資訊，請參閱 [設定 HTTP/2 串流](#)。

### 使用 WebSocket 串流請求轉錄醫療聽寫

若要使用要求在即時串流中轉錄醫療聽寫，WebSocket 請建立預先簽署的 URI。此 URI 包含在您的應用程式和 Amazon Transcribe 醫療之間建立音訊串流時所需的資訊。如需建立 WebSocket 請求的詳細資訊，請參閱 [設定串 WebSocket 流](#)。

使用下列範本來建立您的預先簽署 URI。

```
GET wss://transcribestreaming.us-west-2.amazonaws.com:8443/medical-stream-  
transcription-websocket  
?language-code=languageCode  
&X-Amz-Algorithm=AWS4-HMAC-SHA256  
&X-Amz-Credential=AKIAIOSFODNN7EXAMPLE%2F20220208%2Fus-  
west-2%2Ftranscribe%2Faws4_request  
&X-Amz-Date=20220208T235959Z  
&X-Amz-Expires=300  
&X-Amz-Security-Token=security-token
```

```
&X-Amz-Signature=Signature Version 4 signature  
&X-Amz-SignedHeaders=host  
&media-encoding=flac  
&sample-rate=16000  
&session-id=sessionId  
&specialty=medicalSpecialty  
&type=DICTATION  
&vocabulary-name=vocabularyName  
&show-speaker-label=boolean
```

如需建立預先簽署 URI 的詳細資訊，請參閱[設定串 WebSocket 流](#)。

## 通過醫學定制詞彙提高轉錄準確性

為了提高Amazon Transcribe醫療中的轉錄準確性，請創建和使用一個或多個自定義醫學詞彙。自訂字彙是特定領域的單字或片語的集合。此集合有助於提高 Amazon Transcribe ical 在轉錄這些單詞或短語方面的性能。

當您使用Amazon Transcribe醫療服務時，您必須對自己資料的完整性負責。請勿在自訂詞彙中輸入機密資訊、個人資訊 (PII) 或受保護的健康資訊 (PHI)。

為了獲得最佳效果，請創建單獨的小型自定義詞彙，每個詞彙都有助於轉錄特定的錄音。與創建一個用於所有錄音的大型自定義詞彙相比，您在轉錄準確性方面獲得了更大的改進。

根據預設，您的AWS 帳戶在您的中最多可有 100 個自訂語彙 自訂字彙的大小不得超過 50 KB。有關要求增加您可以使用的自訂詞彙量的資訊AWS 帳戶，請參閱[AWS服務配額](#)。

自訂詞彙提供美式英文版本 (zh-TW)。

### 主題

- [為您的醫療自定義詞彙創建文本文件](#)
- [使用文本文件創建醫學自定義詞彙](#)
- [使用醫學自定義詞彙轉錄音頻文件](#)
- [使用醫學自定義詞彙轉錄實時流](#)
- [字符集, 為,Amazon Transcribe 醫學](#)

## 為您的醫療自定義詞彙創建文本文件

若要建立自訂字彙，您可以建立 UTF-8 格式的文字檔案。在此文件中，您創建一個四列表，每列指定一個字段。每個欄位都會告訴「Amazon Transcribe 醫療」網域特定字詞的發音方式，或如何在您的轉錄中顯示這些字詞。您可以將包含這些欄位的文字檔案儲存在 Amazon S3 值區中。

### 了解如何格式化文本文件

若要建立醫療自訂字彙，請將欄名稱輸入為標題列。您可以在標題列下方輸入每個欄的值。

以下是表格中四個資料行的名稱：

- Phrase-列必需的，所需的值
- IPA-列必需的，值可以是可選的
- SoundsLike-列必需的，值可以是可選的
- DisplayAs-列必需的，值可以是可選的

建立自訂語彙時，請務必：

- 使用單一 Tab 字元分隔每一欄。Amazon Transcribe 如果您嘗試用空格或多個 Tab 字元分隔資料行，會擲回錯誤訊息。
- 確保列中的每個值後面沒有尾隨空格或空格。

請確定您為每一欄輸入的值：

- 少於 256 個字元，包括連字號
- 僅使用允許的字元集中的字元，請參閱 [字符集](#)，為 [Amazon Transcribe 醫學](#)。

### 輸入表格欄的值

下列資訊說明如何為表格的四個資料欄指定值：

- **Phrase**— 應可辨識的字詞或片語。您必須在此欄中輸入值。

如果項目是詞組，請以連字號 (-) 區隔字組。例如，輸入 **cerebral autosomal dominant arteriopathy with subcortical infarcts and leukoencephalopathy** 為 **cerebral-autosomal-dominant-arteriopathy-with-subcortical-infarcts-and-leukoencephalopathy**。



首字母縮寫或其他單字的字母應單獨發音，請在單一字母後面接著輸入點，例如 **D.N.A.** 或 **S.T.E.M.I.**。若要輸入首字母縮寫的複數形式，例如「STEM」，請使用連字號分隔「s」與首字母縮寫：「**S.T.E.M.I-s**」您可以使用大寫或小寫字母作為首字母縮寫。

Phrase 欄是必填欄。您可對輸入語言使用任何允許的字元。如需允許的字元，請參閱 [字符集, 為,Amazon Transcribe 醫學](#)。如果您未指定 DisplayAs 欄，「Amazon Transcribe 醫療」會在輸出檔案中使用 Phrase 欄的內容。

- **IPA**(必填欄，值可以是選擇性的) — 若要指定單字或片語的發音，您可以在此欄中包含 [國際拼音字母 \(IPA\)](#) 中的字元。IPA 欄位不能包含前置或結尾空格，而且您必須使用單一空格來區隔輸入中的每個音節。例如，在英文中您可將片語 **acute-respiratory-distress-syndrome** 輸入為 **ə k j u t # # s p # # ə t # # i d # s t # # s s # n d # o # m**。亦可將片語 **A.L.L.** 輸入為 **e # # l # l**。

即使您未指定 IPA 欄的內容，也必須包含空白 IPA 欄。如果您在 IPA 欄中包含值，則無法提供 SoundsLike 欄的值。

如需特定語言允許的 IPA 字元清單，請參閱 [字符集, 為,Amazon Transcribe 醫學](#)。美國英語是醫療中唯一可用的 Amazon Transcribe 語言。

- **SoundsLike**(必填欄位，值可以是選擇性的) — 您可以將單字或片語分成較小的區段，並使用該語言的標準正字法為每個區段提供發音，以模仿單字的發音方式。例如，在英文中您可以提供片語 **cerebral-autosomal-dominant-arteriopathy-with-subcortical-infarcts-and-leukoencephalopathy** 的發音提示，如下所示：**sir-e-brul-aut-o-som-ul-dah-mi-nant-ar-ter-ri-o-pa-thy-with-sub-cor-ti-cul-in-farcts-and-lewk-o-en-ce-phul-ah-pu-thy**。片語 **atrioventricular-nodal-reentrant-tachycardia** 的提示看起來像這樣：**ay-tree-o-ven-trick-u-lar-node-al-re-entr-ant-tack-ih-card-ia**。您可以使用連字號 (-) 區隔提示的每個部分。

即使您未提供 SoundsLike 欄的值，也必須包含空白 SoundsLike 欄。如果您在 SoundsLike 欄中包含值，則無法提供 IPA 欄的值。

您可對輸入語言使用任何允許的字元。針對允許使用的字元清單，請參閱 [字符集, 為,Amazon Transcribe 醫學](#)。

- **DisplayAs**(必要欄，值可以是選擇性的) — 定義單字或片語輸出時的外觀。例如，如果字組或片語為 **cerebral-autosomal-dominant-arteriopathy-with-subcortical-infarcts-and-leukoencephalopathy**，則您可以將顯示形式指定為 **cerebral autosomal dominant arteriopathy with subcortical infarcts and leukoencephalopathy**，使連字

號不會出現。如果您想要在輸出中顯示縮寫，而不是完整的術語，您也可以指定 `DisplayAs` 為 `CADASIL`。

如果您未指定 `DisplayAs` 欄，「Amazon Transcribe 醫療」會在輸出中使用輸入檔案中的 `Phrase` 欄。

您可以在 `DisplayAs` 欄中使用任何 UTF-8 字元。

您只能為 IPA 和欄中的值包含空 `DisplayAs` 格。

若要建立自訂字彙的文字檔案，請將文字檔中的每個單字或片語放在不同的行上。請使用 Tab 字元分隔欄。僅在 IPA 和 `DisplayAs` 欄中包含值。將副檔名為 `副.txt` 檔名的檔案儲存在與您使用「Amazon Transcribe 醫療」建立自訂字彙相同 AWS 區域的 Amazon S3 值區中。

如果您在 Windows 中編輯文字檔案，請確定您的檔案為 LF 格式，而非 CRLF 格式。否則，您將無法建立自訂詞彙。某些文字編輯器可讓您使用「尋找」和「取代」命令變更格式。

下列範例顯示可用來建立自訂詞彙的文字。若要從這些範例建立自訂詞彙，請將範例複製到文字編輯器中，以 Tab 字元取代 [TAB]，然後將儲存的文字檔案上傳到 Amazon S3。

```
Phrase[TAB]IPA[TAB]SoundsLike[TAB]DisplayAs
acute-respiratory-distress-syndrome[TAB][TAB][TAB]acute respiratory distress syndrome
A.L.L.[TAB]e# # 1 # 1[TAB][TAB]ALL
atrioventricular-nodal-reentrant-tachycardia[TAB][TAB]ay-tree-o-ven-trick-u-lar-node-
al-re-entr-ant-tack-ih-card-ia[TAB]
```

您可以按任何順序輸入欄。下列範例顯示自訂詞彙輸入檔案的其他有效結構。

```
Phrase[TAB]SoundsLike[TAB]IPA[TAB]DisplayAs
acute-respiratory-distress-syndrome[TAB][TAB][TAB]acute respiratory distress syndrome
A.L.L.[TAB][TAB]e# # 1 # 1[TAB]ALL
atrioventricular-nodal-reentrant-tachycardia[TAB]ay-tree-o-ven-trick-u-lar-node-al-re-
entr-ant-tack-ih-card-ia[TAB][TAB]
```

```
DisplayAs[TAB]SoundsLike[TAB]IPA[TAB]Phrase
acute respiratory distress syndrome[TAB][TAB][TAB]acute-respiratory-distress-syndrome
ALL[TAB][TAB]e# # 1 # 1[TAB]A.L.L.
[TAB]ay-tree-o-ven-trick-u-lar-node-al-re-entr-ant-tack-ih-card-ia[TAB]
[TAB]atrioventricular-nodal-reentrant-tachycardia
```

為了便於閱讀，下表以 html 格式更清楚地顯示前面的例子。它們只是為了說明這些例子。

片語	IPA	SoundsLike	DisplayAs
acute-respiratory-distress-syndrome			acute respiratory distress syndrome
A.L.L.	eɪ ɛ   ɛ		ALL
atrioventricular-nodal-reentrant-tachycardia		ay-tree-o-ven-trick-ular-node-al-re-entr-ant-tack-ih-card-ia	

片語	SoundsLike	IPA	DisplayAs
acute-respiratory-distress-syndrome			acute respiratory distress syndrome
atrioventricular-nodal-reentrant-tachycardia	ay-tree-o-ven-trick-ular-node-al-re-entr-ant-tack-ih-card-ia		
A.L.L.		eɪ ɛ   ɛ	ALL

DisplayAs	SoundsLike	IPA	片語
acute respiratory distress syndrome			acute-respiratory-distress-syndrome
ALL		eɪ ɛ   ɛ	A.L.L.
	ay-tree-o-ven-trick-ular-node-al-re-entr-ant-tack-ih-card-ia		atrioventricular-nodal-reentrant-tachycardia

## 使用文本文件創建醫學自定義詞彙

若要建立自訂字彙，您必須準備好包含單字或片語集合的文字檔案。Amazon TranscribeMedical 會使用此文字檔建立自訂語彙，可用於改善這些字詞或片語的轉錄準確性。您可以使用 [CreateMedicalVocabulary](#) API 或 Amazon Transcribe 醫療主控台建立自訂字彙。

### AWS Management Console

若要使用 AWS Management Console 建立自訂字彙，請提供包含單字或片語之文字檔案的 Amazon S3 URI。

1. 登入 [AWS Management Console](#)。
2. 在導覽窗格的 MedAmazon Transcribe ical 下，選擇自訂語彙。
3. 在「名稱」中，在「字彙設定」下，選擇自訂字彙的名稱。
4. 在 Amazon S3 中指定音訊檔案或影片檔案的位置：
  - 對於 S3 字彙設定下的字彙輸入檔案位置，指定識別用於建立自訂詞彙之文字檔的 Amazon S3 URI。
  - 對於 S3 中的字彙輸入檔案位置，請選擇瀏覽 S3 以瀏覽文字檔並選擇它。
5. 選擇 [建立字彙]。

您可以在中查看自訂字彙的處理狀態 AWS Management Console。

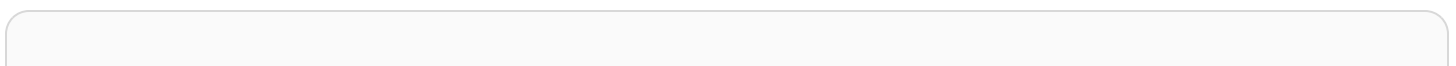
### API

#### 建立醫學自訂詞彙 (API)

- 對於 [StartTranscriptionJob](#) API，請指定以下內容。
  - a. 在 LanguageCode，請指定 en-US。
  - b. 對於 VocabularyFileUri，指 Amazon S3 定您用來定義自訂字彙的文字檔位置。
  - c. 對於 VocabularyName，指定自訂字彙的名稱。您指定的名稱在您的中必須是唯一的 AWS 帳戶。

若要查看自訂詞彙的處理狀態，請使用 [GetMedicalVocabulary](#) API。

以下為使用建立自訂語彙的範例。AWS SDK for Python (Boto3)



```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
vocab_name = "my-first-vocabulary"
response = transcribe.create_medical_vocabulary(
    VocabularyName = job_name,
    VocabularyFileUri = 's3://DOC-EXAMPLE-BUCKET/my-vocabularies/my-vocabulary-
table.txt'
    LanguageCode = 'en-US',
)

while True:
    status = transcribe.get_medical_vocabulary(VocabularyName = vocab_name)
    if status['VocabularyState'] in ['READY', 'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
print(status)
```

## AWS CLI

在批次轉錄工作中啟用喇叭分割 (AWS CLI)

- 執行下列程式碼。

```
aws transcribe create-medical-vocabulary \
--vocabulary-name my-first-vocabulary \
--vocabulary-file-uri s3://DOC-EXAMPLE-BUCKET/my-vocabularies/my-vocabulary-
file.txt \
--language-code en-US
```

## 使用醫學自定義詞彙轉錄音頻文件

使用[StartMedicalTranscriptionJob](#)或啟動AWS Management Console轉錄工作，該工作使用自訂詞彙來提高轉錄準確性。

### AWS Management Console

1. 登入 [AWS Management Console](#)。

2. 在導覽窗格的 MedAmazon Transcribe ical 下，選擇轉錄工作。
3. 選擇 Create job (建立任務)。
4. 在 [指定工作詳細資料] 頁面上，提供轉錄工作的相關資訊。
5. 選擇 下一步。
6. 在自訂下，啟用自訂字彙。
7. 在「字彙選擇」下，選擇自訂字彙。
8. 選擇 建立。

## API

使用批次轉錄工作 (API) 在音訊檔案中啟用喇叭分割

- 對於 [StartMedicalTranscriptionJob](#) API，請指定以下內容。
  - a. 對於 MedicalTranscriptionJobName，指定在您的中唯一的名稱 AWS 帳戶。
  - b. 對於 LanguageCode，請指定與音訊檔案中所說語言相對應的語言代碼，以及字彙篩選器的語言。
  - c. 針對 Media 物件的 MediaFileUri 參數，指定您要轉錄的音訊檔案名稱。
  - d. 對於 Specialty，指定臨床醫生在音頻文件中說話的醫療專業。
  - e. 對於 Type，指定音訊檔案是交談還是聽寫。
  - f. 對於 OutputBucketName，指定 Amazon S3 儲存轉錄結果的值區。
  - g. 對於 Settings 物件，請指定下列項目。
    - VocabularyName— 您的自定義詞彙的名稱。

下列請求會使 AWS SDK for Python (Boto3) 用啟動具有自訂字彙的批次轉錄工作。

```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
job_name = "my-first-med-transcription-job"
job_uri = "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
transcribe.start_medical_transcription_job(
    MedicalTranscriptionJobName = job_name,
```

```
Media = {
    'MediaFileUri': job_uri
},
OutputBucketName = 'DOC-EXAMPLE-BUCKET',
OutputKey = 'my-output-files/',
LanguageCode = 'en-US',
Specialty = 'PRIMARYCARE',
Type = 'CONVERSATION',
Settings = {
    'VocabularyName': 'example-med-custom-vocab'
}
)

while True:
    status = transcribe.get_medical_transcription_job(MedicalTranscriptionJobName =
job_name)
    if status['MedicalTranscriptionJob']['TranscriptionJobStatus'] in ['COMPLETED',
'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
print(status)
```

## 使用醫學自定義詞彙轉錄實時流

若要提高即時串流中的轉錄準確性，您可以使用 HTTP/2 或 WebSocket 串流來使用自訂詞彙。若要啟動 HTTP/2 要求，請使用 [StartMedicalStreamTranscription](#) API。您可以使用 [StartMedicalStreamTranscription](#) API 或 WebSocket 通訊協定即時使用自訂字彙。AWS Management Console

轉錄您的麥克風口說的聽寫 ( AWS Management Console )

若要使用 AWS Management Console 來轉錄醫療聽寫的串流音訊，請選擇轉錄醫療聽寫、開始串流，然後開始對著麥克風說話的選項。

轉錄醫療聽寫的串流音訊 ( AWS Management Console )

1. 登入 [AWS Management Console](#)。
2. 在導覽窗格的 MedAmazon Transcribe ical 下，選擇即時轉錄。
3. 對於醫療專業，請選擇在流中講話的臨床醫生的醫學專業。

4. 在「音訊」輸入類型中，選擇「對話」或「聽寫」。
5. 對於其他設定，請選擇「自訂字彙」。
  - 對於詞彙選擇，請選擇自訂字彙。
6. 選擇 Start streaming (開始串流)。
7. 對著麥克風說話。

准許 HTTP/2 串流中的喇叭分割

以下是 HTTP/2 請求的參數的語法。

```
POST /medical-stream-transcription HTTP/2
host: transcribestreaming.us-west-2.amazonaws.com
authorization: Generated value
x-amz-target: com.amazonaws.transcribe.Transcribe.StartMedicalStreamTranscription
x-amz-content-sha256: STREAMING-MED-AWS4-HMAC-SHA256-EVENTS
x-amz-date: 20220208T235959Z
x-amzn-transcribe-session-id: my-first-http2-med-stream
x-amzn-transcribe-language-code: en-US
x-amzn-transcribe-media-encoding: flac
x-amzn-transcribe-sample-rate: 16000
x-amzn-transcribe-vocabulary-name: my-first-med-vocab
x-amzn-transcribe-specialty: PRIMARYCARE
x-amzn-transcribe-type: CONVERSATION
x-amzn-transcribe-show-speaker-label: true
Content-type: application/vnd.amazon.eventstream
transfer-encoding: chunked
```

參數說明：

- 主機：使用AWS 區域您正在調用的更新AWS 區域（'us-west-2' 在前面的例子中）。如需有效清單AWS 區域，請參閱[AWS 區域和端點](#)。
- 授權：這是一個生成的字段。若要深入瞭解如何建立簽名，請參閱[使用簽名版本 4 簽署AWS請求](#)。
- x-amz-target: 請勿變更此欄位；請使用上述範例中顯示的內容。
- x-amz-content-sha256：這是產生的欄位。若要深入瞭解如何計算簽名，請參閱[使用簽名版本 4 簽署AWS請求](#)。
- x-amz-date：建立簽章的日期和時間。格式為年月，年 = 年 = 年，毫米 = 月，日 = 日，HH = 小時，毫米 = 分，SS = 秒，和 'T' 和 'Z' 是固定字符。如需詳細資訊，請參閱 [Signature 第 4 版中的處理日期](#)。



- x-amzn-transcribe-session-id：串流工作階段的名稱。
- x-amzn-transcribe-language-code：用於輸入音頻的編碼。如需有效值 [支援的語言和特定語言功能的清單](#)，請參閱 [StartMedicalStreamTranscription](#) 或。
- x-amzn-transcribe-media-編碼：用於輸入音頻的編碼。有效值為 pcm、ogg-opus 和 flac。
- x-amzn-transcribe-sample-頻率：輸入音頻的採樣率（以赫茲為單位）。Amazon Transcribe 支援 8,000 赫茲到 48,000 赫茲的範圍。低質量的音頻，例如電話音頻，通常約為 8,000 Hz。高品質的音訊通常介於 16,000 赫茲至 48,000 赫茲之間。請注意，您指定的取樣率必須與音訊的取樣率相符。
- x-amzn-transcribe-vocabulary-name：您要與轉錄一起使用的詞彙的名稱。
- x-amzn-transcribe-specialty：正在轉錄的醫療專業。
- x-amzn-transcribe-type：選擇這是聽寫還是對話。
- x-amzn-transcribe-show-揚聲器標籤：要啟用日記，此值必須是 true。
- content 類型：請勿變更此欄位；請使用上述範例中顯示的內容。

在 WebSocket 要求中啟用發言者分割

若要使用 API 分割 WebSocket 串流中的發言者，請使用下列格式建立預先簽署的 URI，以啟動 WebSocket 要求並設 vocabulary-name 定為自訂字彙的名稱。

```
GET wss://transcribestreaming.us-west-2.amazonaws.com:8443/medical-stream-
transcription-websocket
?language-code=en-US
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIAIOSFODNN7EXAMPLE%2F20220208%2Fus-
west-2%2Ftranscribe%2Faws4_request
&X-Amz-Date=20220208T235959Z
&X-Amz-Expires=300
&X-Amz-Security-Token=security-token
&X-Amz-Signature=Signature Version 4 signature
&X-Amz-SignedHeaders=host
&media-encoding=flac
&sample-rate=16000
&session-id=sessionId
&specialty=medicalSpecialty
&type=CONVERSATION
&vocabulary-name=vocabularyName
&show-speaker-label=boolean
```

## 字符集, 為,Amazon Transcribe 醫學

若要在「Amazon Transcribe醫療」中使用自訂字彙，請使用下列字元集。

### 英文字元集

若是英文自訂詞彙，您可以在 Phrase 和 SoundsLike 欄中使用下列字元：

- a - z
- A - Z
- ' (單引號)
- - (連字號)
- . (句點)

您可以在詞彙輸入檔案的 IPA 欄中，使用下列國際音標 (IPA) 字元。

字元	代碼	字元	代碼
au	0061 028A	w	0077
aɪ	0061 026A	z	007A
b	0062	æ	00E6
d	0064	ð	00F0
eɪ	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

字元	代碼	字元	代碼
l	006C	g	0261
ɫ	006C 0329	ɪ	026A
m	006D	ɹ	0279
n	006E	ʃ	0283
ŋ	006E 0329	ʊ	028A
oʊ	006F 028A	ʌ	028C
p	0070	ɹ	028D
s	0073	ʒ	0292
t	0074	dʒ	02A4
u	0075	tʃ	02A7
v	0076	θ	03B8

## 識別轉錄中的個人健康資訊 (PHI)

使用個人 Health 信息識別在轉錄結果中標記個人健康信息 (PHI)。通過查看標籤，您可以找到可用於識別患者的 PHI。

您可以使用即時串流或批次轉錄工作來識別 PHI。

您可以使用自己的後處理來編寫轉錄輸出中識別的 PHI。

使用個人 Health 信息識別來識別以下類型的 PHI：

- 個人 PHI:
  - 名稱 — 全名或姓氏和首字母
  - 性別
  - 年齡
  - 電話號碼

- 與患者直接相關的日期 ( 不包括年份 )
- 電子郵件地址
- 地理 PHI:
  - 實體位址
  - 郵政編碼
  - 醫療中心或診所的名稱
- 帳戶 PHI :
  - 傳真號碼
  - 社會安全號碼
  - Health 保險受益人號碼
  - 帳號 :
  - 證書或許可證號碼
- 車輛 PHI:
  - 車輛識別符 (VIN)
  - 車牌號碼
- 其他 PHI:
  - 網路統一資源位置 (URL)
  - 網際網路通訊協定 (IP) 位址號碼

Amazon Transcribe 美國 Health 保險流通與責任法案 (HIPAA) 符合 HIPAA (HIPAA) 條件的服務。如需詳細資訊，請參閱 [Amazon Transcribe Medical](#)。如需有關識別音訊檔案中 PHI 的資訊，請參閱 [識別音頻文件中的 PHI](#)。如需識別串流中 PHI 的詳細資訊，請參閱 [識別即時串流中的 PHI](#)。

## 主題

- [識別音頻文件中的 PHI](#)
- [識別即時串流中的 PHI](#)

## 識別音頻文件中的 PHI

使用批次轉錄工作來轉錄音訊檔案，並識別其中的個人健康資訊 (PHI)。當您啟用個人 Health 資訊 (PHI) 識別資訊時，Amazon Transcribe 醫療會將其在轉錄結果中識別的 PHI 加上標籤。有關 Amazon Transcribe 醫療可識別的 PHI 的信息，請參閱 [識別轉錄中的個人健康資訊 \(PHI\)](#)。

您可以使用 [StartMedicalTranscriptionJobAPI](#) 或AWS Management Console。

## AWS Management Console

若要使用AWS Management Console轉錄臨床醫生與病人對話，請建立轉錄工作，然後針對 [音訊] 輸入類型選擇 [交談]。

轉錄音訊檔案並識別其 PHI (AWS Management Console)

1. 登入 [AWS Management Console](#)。
2. 在導覽窗格的Amazon Transcribe醫療下，選擇轉錄工作。
3. 選擇 Create job (建立任務)。
4. 在 [指定 Job 詳細資訊] 頁面的 [工作設定] 下，指定下列項目。
  - a. 名稱 — 您的專屬轉錄工作的名稱AWS 帳戶。
  - b. 音訊輸入類型 — 「對話」或「聽寫」。
5. 對於其餘欄Amazon S3位，請指定音訊檔案的位置，以及要儲存轉錄工作輸出的位置。
6. 選擇 下一步。
7. 在「音訊設定」下，選擇「PHI 識別」
8. 選擇 建立。

## API

使用批次轉錄工作 (API) 轉錄音訊檔案並識別其 PHI

- 對於 [StartMedicalTranscriptionJobAPI](#)，請指定以下內容。
  - a. 對於MedicalTranscriptionJobName，指定您的唯一名稱AWS 帳戶。
  - b. 對於LanguageCode，指定與音訊檔案中所說語言相對應的語言代碼。
  - c. 針對Media物件的MediaFileUri參數，指定您要轉錄的音訊檔案名稱。
  - d. 對於Specialty，將臨床醫生在音頻文件中說話的醫學專業指定為PRIMARYCARE。
  - e. 對於 Type，指定 CONVERSATION 或 DICTATION.
  - f. 針對OutputBucketName，指定您希望在其中存放轉錄結果的儲存Amazon S3貯體。

以下是使用轉錄音訊檔案並識別患者 PHI 的範例請求。AWS SDK for Python (Boto3)

```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe')
job_name = "my-first-transcription-job"
job_uri = "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-audio-file.flac"
transcribe.start_medical_transcription_job(
    MedicalTranscriptionJobName = job_name,
    Media = {'MediaFileUri': job_uri},
    LanguageCode = 'en-US',
    ContentIdentificationType = 'PHI',
    Specialty = 'PRIMARYCARE',
    Type = 'type', # Specify 'CONVERSATION' for a medical conversation. Specify
    'DICTATION' for a medical dictation.
    OutputBucketName = 'DOC-EXAMPLE-BUCKET'
)
while True:
    status = transcribe.get_medical_transcription_job(MedicalTranscriptionJobName =
    job_name)
    if status['MedicalTranscriptionJob']['TranscriptionJobStatus'] in ['COMPLETED',
    'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
print(status)
```

以下示例代碼顯示了識別 PHI 患者的轉錄結果。

```
{
  "jobName": "my-medical-transcription-job-name",
  "accountId": "111122223333",
  "results": {
    "transcripts": [{
      "transcript": "The patient's name is Bertrand."
    }],
    "items": [{
      "start_time": "0.0",
      "end_time": "0.37",
```

```
    "alternatives": [{
      "confidence": "0.9993",
      "content": "The"
    }],
    "type": "pronunciation"
  }, {
    "start_time": "0.37",
    "end_time": "0.44",
    "alternatives": [{
      "confidence": "0.9981",
      "content": "patient's"
    }],
    "type": "pronunciation"
  }, {
    "start_time": "0.44",
    "end_time": "0.52",
    "alternatives": [{
      "confidence": "1.0",
      "content": "name"
    }],
    "type": "pronunciation"
  }, {
    "start_time": "0.52",
    "end_time": "0.92",
    "alternatives": [{
      "confidence": "1.0",
      "content": "is"
    }],
    "type": "pronunciation"
  }, {
    "start_time": "0.92",
    "end_time": "0.9989",
    "alternatives": [{
      "confidence": "1.0",
      "content": "Bertrand"
    }],
    "type": "pronunciation"
  }, {
    "alternatives": [{
      "confidence": "0.0",
      "content": "."
    }],
    "type": "punctuation"
  }],
}
```

```
    "entities": [{
      "content": "Bertrand",
      "category": "PHI*-Personal*",
      "startTime": 0.92,
      "endTime": 1.2,
      "confidence": 0.9989
    }],
  },
  "status": "COMPLETED"
}
```

## AWS CLI

### 使用批次轉錄工作轉錄音訊檔案並識別 PHI (AWS CLI)

- 執行下列程式碼。

```
aws transcribe start-medical-transcription-job \
--medical-transcription-job-name my-medical-transcription-job-name \
--language-code en-US \
--media MediaFileUri="s3://DOC-EXAMPLE-BUCKET/my-input-files/my-audio-file.flac" \
--output-bucket-name DOC-EXAMPLE-BUCKET \
--specialty PRIMARYCARE \
--type type \ # Choose CONVERSATION to transcribe a medical conversation.
               Choose DICTATION to transcribe a medical dictation.
--content-identification-type PHI
```

## 識別即時串流中的 PHI

您可以在 HTTP/2 或 WebSocket 流中識別個人 Health 信息 (PHI)。當您啟用 PHI 識別時，「Amazon Transcribe 醫療」會在轉錄結果中標示其識別的 PHI。有關 Amazon Transcribe 醫療可識別的 PHI 的信息，請參閱 [識別轉錄中的個人健康資訊 \(PHI\)](#)。

### 識別麥克風口述的聽寫中的 PHI

若要使用 AWS Management Console 來轉錄麥克風接收的語音並識別任何 PHI，請選擇聽寫作為音訊輸入類型，開始串流，然後開始對電腦上的麥克風講話。



## 若要在聽寫中使用AWS Management Console

1. 登入 [AWS Management Console](#)。
2. 在瀏覽窗格中，選擇 Real-time transcription (即時轉錄)。
3. 在「音訊輸入類型」中，選擇「聽寫」。
4. 對於其他設定，請選擇 PHI 識別。
5. 選擇開始串流並對著麥克風說話。
6. 選擇「停止串流」以結束聽寫。

## 在 HTTP/2 串流中識別 PHI

若要在啟動 PHI 識別的情況下啟動 HTTP/2 串流，請使用 [StartMedicalStreamTranscription](#) API 並指定下列項目：

- 對於 LanguageCode，指定串流中所說語言的語言代碼。對於「美式英文」，請指定 en-US。
- 對於 MediaSampleHertz，指定音訊的取樣率。
- 在 content-identification-type，請指定 PHI。

## 識別 WebSocket 串流中的 PHI

若要在啟動 PHI 識別的情況下啟動 WebSocket 串流，請使用下列格式來建立預先簽署的 URL。

```
GET wss://transcribestreaming.us-west-2.amazonaws.com:8443/medical-stream-  
transcription-websocket?  
&X-Amz-Algorithm=AWS4-HMAC-SHA256  
&X-Amz-Credential=AKIAIOSFODNN7EXAMPLE%2F20220208%2Fus-  
west-2%2Ftranscribe%2Faws4_request  
&X-Amz-Date=20220208T235959Z  
&X-Amz-Expires=300  
&X-Amz-Security-Token=security-token  
&X-Amz-Signature=Signature Version 4 signature  
&X-Amz-SignedHeaders=host  
&language-code=en-US  
&media-encoding=flac  
&sample-rate=16000  
&specialty=medical-specialty  
&content-identification-type=PHI
```

您可以在 [API 參考中找到參數定義](#)；所有AWS API 作業的共用參數會列在「[一般參數](#)」區段中。

## 生成替代轉錄

當您使用「Amazon Transcribe醫療」時，您將獲得信心水平最高的轉錄。不過，您可以設定「Amazon Transcribe醫療」以傳回其他可信度較低的轉錄。

使用替代轉錄來查看所轉錄音訊的不同解釋。例如，在使用人員能夠審核轉錄的應用程式中，您可以提供可供人員選擇的替代轉錄。

您可以使用AWS Management Console或 [StartMedicalTranscriptionJob](#) API 產生替代轉錄。

### AWS Management Console

若要使用AWS Management Console產生替代轉錄，請在設定工作時啟用替代結果。

1. 登入 [AWS Management Console](#)。
2. 在導覽窗格的 MEDIAL (Amazon Transcribe醫療) 中，選擇 Tran( 轉錄工作)。
3. 選擇 Create job (建立任務)。
4. 在 [指定工作詳細資料] 頁面上，提供轉錄工作的相關資訊。
5. 選擇 Next (下一步)。
6. 啟用替代結果。
7. 在「最大」替代方案中，輸入介於 2 到 10 之間的整數值，以取得您要在輸出中輸出的替代轉錄數目上限。
8. 選擇 Create (建立)。

### API

使用批次轉錄工作 (API) 分隔音訊檔案中每位發言者的文字

- 對於 [StartMedicalTranscriptionJob](#) API，請指定以下內容。
  - a. 對於MedicalTranscriptionJobName，指定在您的中唯一的名稱AWS 帳戶。
  - b. 對於LanguageCode，請指定與音訊檔案中所說語言相對應的語言代碼，以及字彙篩選器的語言。
  - c. 在Media物件的MediaFileUri參數中，指定要轉錄的音訊檔案的位置。
  - d. 對於Specialty，指定臨床醫生在音頻文件中說話的醫療專業。

- e. 對於Type，指定要轉錄醫學對話還是聽寫。
- f. 對於OutputBucketName，指定Amazon S3儲存轉錄結果的值區。
- g. 對於Settings物件，請指定下列項目。
  - i. ShowAlternatives – true.
  - ii. MaxAlternatives-介於 2 到 10 之間的整數，用來指出轉錄輸出中所需的替代轉錄數。

下列要求會使用啟動AWS SDK for Python (Boto3)轉錄工作，該工作最多可產生兩個替代轉錄。

```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
job_name = "my-first-transcription-job"
job_uri = s3://DOC-EXAMPLE-BUCKET/my-input-files/my-audio-file.flac
transcribe.start_medical_transcription_job(
    MedicalTranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'DOC-EXAMPLE-BUCKET',
    OutputKey = 'my-output-files/',
    LanguageCode = 'en-US',
    Specialty = 'PRIMARYCARE',
    Type = 'CONVERSATION',
    Settings = {
        'ShowAlternatives': True,
        'MaxAlternatives': 2
    }
)

while True:
    status = transcribe.get_medical_transcription_job(MedicalTranscriptionJobName =
job_name)
    if status['MedicalTranscriptionJob']['TranscriptionJobStatus'] in ['COMPLETED',
'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
```

```
print(status)
```

## AWS CLI

在音頻文件中轉錄初級保健臨床醫生和患者之間對話的音頻文件 ( AWS CLI )

- 執行下列程式碼。

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

下列程式碼顯示的內容*example-start-command.json*。

```
{  
  "MedicalTranscriptionJobName": "my-first-transcription-job",  
  "LanguageCode": "en-US",  
  "Specialty": "PRIMARYCARE",  
  "Type": "CONVERSATION",  
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",  
  "Media": {  
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-audio-  
file.flac"  
  },  
  "Settings": {  
    "ShowAlternatives": true,  
    "MaxAlternatives": 2  
  }  
}
```

## Amazon Transcribe醫療和介面 VPC 端點 (AWS PrivateLink)

您可以在 VPC 和之間建立私人連接Amazon Transcribe通過創建醫療VPC 端點介面。介面端點由[AWS PrivateLink](#)，一種使您能夠私人訪問的技術Amazon Transcribe沒有網際網路閘道、NAT 裝置、VPN 連線或AWS Direct Connect連接。VPC 中的執行個體不需要公用 IP 位址即可與之通訊Amazon Transcribe醫療 API。您的 VPC 和之間的流量Amazon Transcribe醫療不會離開亞馬遜網絡。

每個介面端點都是由您子網路中的一或多個[彈性網絡介面](#)表示。

如需詳細資訊，請參閱[介面虛擬私人雲端端點 \(AWS PrivateLink\)](#)在Amazon VPC使用者指南。

## 的注意事項Amazon Transcribe醫療 VPC 端點

在為下列項目設定介面 VPC 端點之前Amazon Transcribe醫療，確保您進行審查[介面端點內容和限制](#)在Amazon VPC使用者指南。

Amazon Transcribe醫療支援從您的 VPC 呼叫其所有 API 動作。

## 為建立介面 VPC 端點Amazon Transcribe醫療

您可以建立 VPC 端點Amazon Transcribe醫療服務使用AWS Management Console或AWS CLI。如需詳細資訊，請參閱[建立介面端點](#)在Amazon VPC使用者指南。

對於批量轉錄Amazon Transcribe醫療，請使用下列服務名稱建立 VPC 端點：

- COM. 亞馬遜。##-## -2. 轉錄

用於串流轉錄Amazon Transcribe醫療，請使用下列服務名稱建立 VPC 端點：

- COM. 亞馬遜。##-## -2. 轉錄流

如果您為端點啟用私有 DNS，則可以向以下位置發出 API 請求：Amazon Transcribe醫療使用其默認 DNS 名稱AWS 區域，例如，transcribestreaming.us-east-2.amazonaws.com。

如需詳細資訊，請參閱[透過介面端點存取服務](#)在Amazon VPC使用者指南。

## 建立 VPC 端點原則Amazon Transcribe醫學串流

您可以將端點策略附加到控制存取的 VPC 端點Amazon Transcribe醫療。此政策會指定下列資訊：

- 可執行動作的主體。
- 可執行的動作。
- 可供執行動作的資源。

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的[使用 VPC 端點控制對服務的存取](#)。

範例：的 VPC 端點原則Amazon Transcribe醫學串流轉錄動作

以下是用於串流轉錄的端點原則範例Amazon Transcribe醫療。連接到端點時，此策略會授與對列出的端點的存取權Amazon Transcribe對所有資源的所有校長的醫療行動。

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Allow",
      "Action": [
        "transcribe:StartMedicalStreamTranscription",
      ],
      "Resource": "*"
    }
  ]
}
```

範例：的 VPC 端點原則Amazon Transcribe醫療批量轉錄動作

以下是用於批次轉錄的端點原則範例Amazon Transcribe醫療。連接到端點時，此策略會授與對列出的端點的存取權Amazon Transcribe對所有資源的所有校長的醫療行動。

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Allow",
      "Action": [
        "transcribe:StartMedicalTranscriptionJob"
      ],
      "Resource": "*"
    }
  ]
}
```

## 共用子網路

您無法在與您共用的子網路中建立、描述、修改或刪除 VPC 端點。不過，可以在與您共用的子網路中使用 VPC 端點。如需 VPC 共用的資訊，請參閱[與其他帳戶共用您的 VPC](#)在Amazon Virtual Private Cloud指南。

# AWS HealthScribe

AWS HealthScribe 是符合 HIPAA 條件的全新機器學習 (ML) 功能，結合語音辨識和生成式 AI，可轉錄患者與臨床醫生的對話，並產生方便審查的臨床筆記。AWSHealthScribe 協助醫療保健軟體廠商建置臨床應用程式，以減輕文件負擔並改善諮詢體驗。該服務會自動提供豐富的對話記錄、識別發言者角色、進行對話分類、擷取醫學術語並產生初步臨床筆記。AWSHealthScribe 結合這些功能，無需整合及最佳化個別 AI 服務，讓您可以加速實作。

常見使用案例：

- 縮短文件時間 — 讓臨床醫生能夠使用 AI 生成式的臨床筆記快速完成臨床文件，這些筆記方便在您的應用程式中檢閱、調整和完成。
- 提高醫療抄寫員效率 — 為醫療抄寫員提供 AI 生成式的文字記錄和臨床筆記以及諮詢音訊，以加快文件的準備時間。
- 有效率的患者門診回憶 — 建立體驗，讓使用者可以在您的應用程式中快速回憶對話的重點。

## Important

AWS HealthScribe 產生的結果非完全精準，且由於包括音訊品質、背景噪音、發言者清晰度、醫學術語的複雜性、特定環境的語言細微差別，以及[機器學習和生成式 AI 的本質](#)等各種因素，可能並非完全準確。AWSHealthScribe 旨在作為臨床醫生和醫學抄寫的輔助工具。AWSHealthScribe 輸出須經過受過訓練的醫學專業人員實施正確的醫學判斷進行審查後，方可用於病患照護情境，包括但不限於作為電子健康記錄的一部分。AWSHealthScribe 輸出不可取代專業醫療建議、診斷或治療，且並非用於治癒、治療、緩解、預防或診斷任何疾病或健康狀況。

AWS HealthScribe 的運作模式採用共同責任模型，由 AWS 負責保護執行 AWS HealthScribe 的基礎架構，且您負責管理您的資訊。如需詳細資訊，請參閱[共同責任模型](#)。

AWS HealthScribe 提供美國東部 (維吉尼亞北部) 區域使用。

本服務提供美式英文 (en-US) 版本。為了獲得最佳結果，使用不失真的音訊格式，例如 FLAC 或以 PCM 16 位元編碼的 WAV。AWSHealthScribe 支援 16,000 Hz 或更高的取樣率。

AWS HealthScribe 目前支援一般內科和骨科專科。

AWS HealthScribe 作業會分析醫療諮詢以產生兩個 JSON 輸出檔案：一個[文字記錄](#)檔案和一個[臨床文件](#)檔案。

在文字記錄檔案中，除了標準的逐段轉錄輸出與單字時間戳記，AWS HealthScribe 還提供：

- 參與者角色偵測，因此您可以區分對話記錄中的患者語臨床醫生。
- 文字記錄分段，根據其臨床相關性 (例如閒聊、主觀、客觀等)，進行文字記錄對話分類。這可以用來顯示文字記錄的特定部分。
- 臨床實體，其中包括結構化資訊，例如藥物、醫療條件和對話中提到的治療方法。

在臨床文件檔案中，AWS HealthScribe 提供了：

- 摘要，包含臨床文件重要部分的摘要說明，例如主訴、目前病史、系統性回顧、過去病史、評估和計劃。
- 證據連結，將 AI 生成式摘要中使用的每個句子連接到原始的諮詢文字記錄，讓使用者可以更輕鬆驗證應用程式中摘要的準確性。

AWS HealthScribe 專屬的 API 操作：

- StartMedicalScribeJob
- ListMedicalScribeJobs
- GetMedicalScribeJob
- DeleteMedicalScribeJob

若要檢視 AWS HealthScribe 請求範例，請參閱[開始進行 AWS HealthScribe 作業](#)。

## 文字記錄檔案

文字記錄檔案以逐段格式提供對話的內容。

此外，還為每個對話回合提供以下見解：

- 參與者角色 — 每個參與者都被標記為臨床醫生或患者。如果對話在每個類別中有多個參與者，則會為每個參與者指定編號。例如：CLINICIAN\_1、CLINICIAN\_2 和 PATIENT\_1、PATIENT\_2。
- 區段 — 每個對話回合都會根據所識別的內容，指定給四個可能的區段之一。
  - 主觀 — 病患提供其健康問題的資訊。



- 客觀 — 臨床醫師透過身體檢查、實驗室檢驗、影像或診斷測試觀察到的資訊。
- 評估和計劃 — 與醫師的評估和治療計劃相關的資訊。
- 門診流程管理 — 與閒聊或轉診相關的資訊。
- 見解 — 擷取對話中的臨床相關實體 (ClinicalEntity)。AWSHealthScribe 可偵測所有由 [Amazon Comprehend Medical](#) 支援的臨床實體。

如需輸出的詳細資訊，請參閱[範例文字記錄輸出](#)。

## 臨床文件檔案

文件見解檔案包含臨床文件下列重要章節的摘要。

章節	描述
主訴	簡要說明患者前往臨床醫生門診的原因。
目前病史	提供患者疾病資訊的備註事項，包括參考嚴重程度、發病、症狀的時間、目前的治療和受影響的不為。
系統性回顧	病患對全身不同系統回報的症狀評估。
過去病史	詳述病患過去的醫療情形、手術及治療。
評估	供有關臨床醫生對患者健康評估的資訊的備註事項。
計劃	參考任何醫療、生活方式調整和進一步預約的注意事項。

出現在 Summary 的每個句子都包含原始諮詢文字記錄的引用，讓使用者可以更輕鬆驗證應用程式摘要的準確性。提供 AI 生成式見解可追溯性和透明度，與 Responsible AI 原則一致，如可解釋性。提供臨床醫生或醫學抄寫員這些參考以及摘要說明，有助於培養信任並鼓勵在臨床環境中安全使用 AI。

Summary 中的每個句子都隨附 EvidenceLinks，提供摘要文字記錄中相關對話的 SegmentId。

如需更詳細的輸出資訊，請參閱[範例臨床文件輸出](#)。

## 開始 AWS HealthScribe 工作

您可以使用 AWS CLI 或 AWS SDK 開始 AWS HealthScribe 工作；請參閱下列項目，取得範例。

### AWS CLI

此範例使用 [start-medical-scribe-job](#) 指令。如需詳細資訊，請參閱 [StartMedicalScribeJob](#)。

```
aws transcribe start-medical-scribe-job \  
--region us-west-2 \  
--medical-scribe-job-name my-first-medical-scribe-job \  
--media MediaFileUri=s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac \  
--output-bucket-name DOC-EXAMPLE-BUCKET \  
--DataAccessRoleArn=arn:aws:iam::111122223333:role/ExampleRole \  
--settings ShowSpeakerLabels=false,ChannelIdentification=true \  
--channel-definitions ChannelId=0,ParticipantRole=CLINICIAN \  
ChannelId=1,ParticipantRole=PATIENT
```

這是使用 [start-medical-scribe-job](#) 指令的另一個範例，以及帶其他設定的請求內文。

```
aws transcribe start-medical-scribe-job \  
--region us-west-2 \  
--cli-input-json file://filepath/my-first-medical-scribe-job.json
```

檔案 `my-first-medical-scribe-job.json` 包含以下請求內文。

```
{  
  "MedicalScribeJobName": "my-first-medical-scribe-job",  
  "Media": {  
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"  
  },  
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",  
  "DataAccessRoleArn": "arn:aws:iam::111122223333:role/ExampleRole",  
  "Settings": {  
    "ShowSpeakerLabels": false,  
    "ChannelIdentification": true  
  },  
}
```

```

"ChannelDefinitions": [
  {
    "ChannelId": 0,
    "ParticipantRole": "CLINICIAN"
  }, {
    "ChannelId": 1,
    "ParticipantRole": "PATIENT"
  }
]
}

```

## AWS SDK for Python (Boto3)

下列範例會使用 AWS SDK for Python (Boto3) 以提出 [start\\_medical\\_scribe\\_job](#) 請求。如需詳細資訊，請參閱 [StartMedicalScribeJob](#)。

```

from __future__ import print_function
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
job_name = "my-first-medical-scribe-job"
job_uri = "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
transcribe.start_medical_scribe_job(
    MedicalScribeJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'DOC-EXAMPLE-BUCKET',
    DataAccessRoleArn = 'arn:aws:iam::111122223333:role/ExampleRole',
    Settings = {
        'ShowSpeakerLabels': false,
        'ChannelIdentification': true
    },
    ChannelDefinitions = [
        {
            'ChannelId': 0,
            'ParticipantRole': 'CLINICIAN'
        }, {
            'ChannelId': 1,
            'ParticipantRole': 'PATIENT'
        }
    ]
)

```

```
while True:
    status = transcribe.get_medical_scribe_job(MedicalScribeJobName = job_name)
    if status['MedicalScribeJob']['MedicalScribeJobStatus'] in ['COMPLETED', 'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
print(status)
```

### Note

AWS Management Console 目前不支援 AWS HealthScribe 工作。

## 範例輸出

除了文字記錄之外，StartMedicalScribeJob 請求還會產生獨立的臨床文件檔案。這兩個文件都是 JSON 格式，並存放在請求中指定的輸出位置。下列是每個輸出類型的範例：

### 範例文字記錄輸出

AWS HealthScribe 文字記錄檔案 (來自StartMedicalScribeJob 請求) 採用下列格式：

```
{
  "Conversation": {
    "ConversationId": "sampleConversationUUID",
    "JobName": "sampleJobName",
    "JobType": "ASYNC",
    "LanguageCode": "en-US",
    "ClinicalInsights": [
      {
        "Attributes": [],
        "Category": "MEDICAL_CONDITION",
        "InsightId": "insightUUID1",
        "InsightType": "ClinicalEntity",
        "Spans": [
          {
            "BeginCharacterOffset": 12,
            "Content": "pain",
            "EndCharacterOffset": 15,
            "SegmentId": "uuid1"
          }
        ]
      }
    ]
  }
}
```

```
    }
  ],
  "Type": "DX_NAME"
},
{
  "Attributes": [],
  "Category": "TEST_TREATMENT_PROCEDURE",
  "InsightId": "insightUUID2",
  "InsightType": "ClinicalEntity",
  "Spans": [
    {
      "BeginCharacterOffset": 4,
      "Content": "mammogram",
      "EndCharacterOffset": 12,
      "SegmentId": "uuid2"
    }
  ],
  "Type": "TEST_NAME"
},
{
  "Attributes": [],
  "Category": "TEST_TREATMENT_PROCEDURE",
  "InsightId": "insightUUID3",
  "InsightType": "ClinicalEntity",
  "Spans": [
    {
      "BeginCharacterOffset": 15,
      "Content": "pap smear",
      "EndCharacterOffset": 23,
      "SegmentId": "uuid3"
    }
  ],
  "Type": "TEST_NAME"
},
{
  "Attributes": [],
  "Category": "MEDICATION",
  "InsightId": "insightUUID4",
  "InsightType": "ClinicalEntity",
  "Spans": [
    {
      "BeginCharacterOffset": 28,
      "Content": "phentermine",
      "EndCharacterOffset": 38,
```

```
        "SegmentId": "uuid4"
      }
    ],
    "Type": "GENERIC_NAME"
  },
  {
    "Attributes": [
      {
        "AttributeId": "attributeUUID1",
        "Spans": [
          {
            "BeginCharacterOffset": 38,
            "Content": "high",
            "EndCharacterOffset": 41,
            "SegmentId": "uuid5"
          }
        ],
        "Type": "TEST_VALUE"
      }
    ],
    "Category": "TEST_TREATMENT_PROCEDURE",
    "InsightId": "insightUUID5",
    "InsightType": "ClinicalEntity",
    "Spans": [
      {
        "BeginCharacterOffset": 14,
        "Content": "weight",
        "EndCharacterOffset": 19,
        "SegmentId": "uuid6"
      }
    ],
    "Type": "TEST_NAME"
  },
  {
    "Attributes": [],
    "Category": "ANATOMY",
    "InsightId": "insightUUID6",
    "InsightType": "ClinicalEntity",
    "Spans": [
      {
        "BeginCharacterOffset": 60,
        "Content": "heart",
        "EndCharacterOffset": 64,
        "SegmentId": "uuid7"
      }
    ]
  }
}
```

```
    }
  ],
  "Type": "SYSTEM_ORGAN_SITE"
}
],
"TranscriptItems": [
  {
    "Alternatives": [
      {
        "Confidence": 0.7925,
        "Content": "Okay"
      }
    ],
    "BeginAudioTime": 0.16,
    "EndAudioTime": 0.6,
    "Type": "PRONUNCIATION"
  },
  {
    "Alternatives": [
      {
        "Confidence": 0,
        "Content": "."
      }
    ],
    "BeginAudioTime": 0,
    "EndAudioTime": 0,
    "Type": "PUNCTUATION"
  },
  {
    "Alternatives": [
      {
        "Confidence": 1,
        "Content": "Good"
      }
    ],
    "BeginAudioTime": 0.61,
    "EndAudioTime": 0.92,
    "Type": "PRONUNCIATION"
  },
  {
    "Alternatives": [
      {
        "Confidence": 1,
        "Content": "afternoon"
      }
    ]
  }
]
```

```
    }
  ],
  "BeginAudioTime": 0.92,
  "EndAudioTime": 1.54,
  "Type": "PRONUNCIATION"
},
{
  "Alternatives": [
    {
      "Confidence": 0,
      "Content": "."
    }
  ],
  "BeginAudioTime": 0,
  "EndAudioTime": 0,
  "Type": "PUNCTUATION"
},
{
  "Alternatives": [
    {
      "Confidence": 0.9924,
      "Content": "You"
    }
  ],
  "BeginAudioTime": 1.55,
  "EndAudioTime": 1.88,
  "Type": "PRONUNCIATION"
},
{
  "Alternatives": [
    {
      "Confidence": 1,
      "Content": "lost"
    }
  ],
  "BeginAudioTime": 1.88,
  "EndAudioTime": 2.19,
  "Type": "PRONUNCIATION"
},
{
  "Alternatives": [
    {
      "Confidence": 1,
      "Content": "one"
    }
  ]
}
```



```

    }
  ],
  "BeginAudioTime": 2.19,
  "EndAudioTime": 2.4,
  "Type": "PRONUNCIATION"
},
{
  "Alternatives": [
    {
      "Confidence": 1,
      "Content": "lb"
    }
  ],
  "BeginAudioTime": 2.4,
  "EndAudioTime": 2.97,
  "Type": "PRONUNCIATION"
}
],
"TranscriptSegments": [
  {
    "BeginAudioTime": 0.16,
    "Content": "Okay.",
    "EndAudioTime": 0.6,
    "ParticipantDetails": {
      "ParticipantRole": "CLINICIAN_0"
    },
    "SectionDetails": {
      "SectionName": "SUBJECTIVE"
    },
    "SegmentId": "uuid1"
  },
  {
    "BeginAudioTime": 0.61,
    "Content": "Good afternoon.",
    "EndAudioTime": 1.54,
    "ParticipantDetails": {
      "ParticipantRole": "CLINICIAN_0"
    },
    "SectionDetails": {
      "SectionName": "OTHER"
    },
    "SegmentId": "uuid2"
  },
  {

```

```
"BeginAudioTime": 1.55,
"Content": "You lost one lb.",
"EndAudioTime": 2.97,
"ParticipantDetails": {
  "ParticipantRole": "CLINICIAN_0"
},
"SectionDetails": {
  "SectionName": "SUBJECTIVE"
},
"SegmentId": "uuid3"
},
{
  "BeginAudioTime": 2.98,
"Content": "Yeah, I think it, uh, do you feel more energy?",
"EndAudioTime": 6.95,
"ParticipantDetails": {
  "ParticipantRole": "CLINICIAN_0"
},
"SectionDetails": {
  "SectionName": "SUBJECTIVE"
},
"SegmentId": "uuid5"
},
{
  "BeginAudioTime": 6.96,
"Content": "Yes.",
"EndAudioTime": 7.88,
"ParticipantDetails": {
  "ParticipantRole": "CLINICIAN_0"
},
"SectionDetails": {
  "SectionName": "SUBJECTIVE"
},
"SegmentId": "uuid6"
},
{
  "BeginAudioTime": 7.89,
"Content": "Uh, how about craving for the carbohydrate or sugar or fat or
anything?",
"EndAudioTime": 17.93,
"ParticipantDetails": {
  "ParticipantRole": "CLINICIAN_0"
},
"SectionDetails": {
```

```
        "SectionName": "SUBJECTIVE"
      },
      "SegmentId": "uuid7"
    }
  ]
}
}
```

這是使用 [start-medical-scribe-job](#) 指令的另一個範例，以及帶其他設定的請求內文。

```
aws transcribe start-medical-scribe-job \  
--region us-west-2 \  
--cli-input-json file://filepath/my-first-medical-scribe-job.json
```

檔案 `my-first-medical-scribe-job.json` 包含以下請求內文。

```
{  
  "MedicalScribeJobName": "my-first-medical-scribe-job",  
  "Media": {  
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"  
  },  
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",  
  "DataAccessRoleArn": "arn:aws:iam::111122223333:role/ExampleRole",  
  "Settings": {  
    "ShowSpeakerLabels": false,  
    "ChannelIdentification": true  
  },  
  "ChannelDefinitions": [  
    {  
      "ChannelId": 0,  
      "ParticipantRole": "CLINICIAN"  
    }, {  
      "ChannelId": 1,  
      "ParticipantRole": "PATIENT"  
    }  
  ]  
}
```

## 臨床文件輸出範例

文件見解檔案 (來自 StartMedicalScribeJob 要求) 具有下列格式：

```
{
  "ClinicalDocumentation": {
    "Sections": [
      {
        "SectionName": "CHIEF_COMPLAINT",
        "Summary": [
          {
            "EvidenceLinks": [
              {
                "SegmentId": "uuid1"
              },
              {
                "SegmentId": "uuid2"
              },
              {
                "SegmentId": "uuid3"
              },
              {
                "SegmentId": "uuid4"
              },
              {
                "SegmentId": "uuid5"
              },
              {
                "SegmentId": "uuid6"
              }
            ],
            "SummarizedSegment": "Weight loss."
          }
        ]
      },
      {
        "SectionName": "HISTORY_OF_PRESENT_ILLNESS",
        "Summary": [
          {
            "EvidenceLinks": [
              {
                "SegmentId": "uuid7"
              }
            ],

```

```
    {
      "SegmentId": "uuid8"
    },
    {
      "SegmentId": "uuid9"
    },
    {
      "SegmentId": "uuid10"
    }
  ],
  "SummarizedSegment": "The patient is seen today for a follow-up of weight
loss."
},
{
  "EvidenceLinks": [
    {
      "SegmentId": "uuid11"
    },
    {
      "SegmentId": "uuid12"
    },
    {
      "SegmentId": "uuid13"
    }
  ],
  "SummarizedSegment": "They report feeling more energy and craving
carbohydrates, sugar, and fat."
},
{
  "EvidenceLinks": [
    {
      "SegmentId": "uuid14"
    },
    {
      "SegmentId": "uuid15"
    },
    {
      "SegmentId": "uuid16"
    }
  ],
  "SummarizedSegment": "The patient is up to date on their mammogram and pap
smear."
},
{
```

```
    "EvidenceLinks": [
      {
        "SegmentId": "uuid17"
      },
      {
        "SegmentId": "uuid18"
      },
      {
        "SegmentId": "uuid19"
      },
      {
        "SegmentId": "uuid20"
      }
    ],
    "SummarizedSegment": "The patient is taking phentermine and would like to
continue."
  }
]
},
{
  "SectionName": "REVIEW_OF_SYSTEMS",
  "Summary": [
    {
      "EvidenceLinks": [
        {
          "SegmentId": "uuid21"
        },
        {
          "SegmentId": "uuid22"
        }
      ],
      "SummarizedSegment": "Patient reports intermittent headaches, occasional
chest pains but denies any recent fevers or chills."
    },
    {
      "EvidenceLinks": [
        {
          "SegmentId": "uuid23"
        },
        {
          "SegmentId": "uuid24"
        }
      ],

```

```
        "SummarizedSegment": "No recent changes in vision, hearing, or any
respiratory complaints."
    }
]
},
{
    "SectionName": "PAST_MEDICAL_HISTORY",
    "Summary": [
        {
            "EvidenceLinks": [
                {
                    "SegmentId": "uuid25"
                },
                {
                    "SegmentId": "uuid26"
                }
            ],
            "SummarizedSegment": "Patient has a history of hypertension and was
diagnosed with Type II diabetes 5 years ago."
        },
        {
            "EvidenceLinks": [
                {
                    "SegmentId": "uuid27"
                },
                {
                    "SegmentId": "uuid28"
                }
            ],
            "SummarizedSegment": "Underwent an appendectomy in the early '90s and had a
fracture in the left arm during childhood."
        }
    ]
},
{
    "SectionName": "ASSESSMENT",
    "Summary": [
        {
            "EvidenceLinks": [
                {
                    "SegmentId": "uuid29"
                },
                {
                    "SegmentId": "uuid30"
                }
            ]
        }
    ]
}
```





依預設加密靜態資料，有助於降低保護敏感資料所涉及的營運開銷和複雜性。同時，其可讓您建置符合嚴格加密合規性和法規要求的安全應用程式。

雖然您無法停用此層加密或選取替代加密類型，但您可以在使用 AWS HealthScribe 建立作業時選擇客戶受管金鑰，以在現有的 Amazon S3 受管金鑰之上增加另一層加密：

- 客戶受管金鑰：AWS HealthScribe 支援使用您建立、擁有和管理的對稱客戶自管金鑰，以在現有 AWS 擁有的加密之上增加另二層加密。您可以完全控制此層加密，因此能執行以下任務：
  - 建立和維護金鑰政策
  - 建立和維護 IAM 政策和授權
  - 啟用和停用金鑰政策
  - 輪換金鑰密碼編譯資料
  - 新增標籤
  - 建立金鑰別名
  - 安排金鑰供刪除

如需詳細資訊，請參閱《AWS Key Management Service 開發人員指南》中的[客戶受管金鑰](#)。

#### Note

AWS HealthScribe 可免費使用 AWS 擁有的金鑰自動啟用靜態加密，以保護個人身分識別資料。但是，若是使用客戶受管金鑰，則會收取 AWS KMS 費用。如需定價的詳細資訊，請參閱 [AWS Key Management Service 定價](#)。  
如需 AWS KMS 的詳細資訊，請參閱 [什麼是 AWS Key Management Service](#)。

## 建立客戶受管金鑰

您可以使用 AWS Management Console 或 AWS KMS API 建立對稱的客戶受管金鑰。若要建立對稱客戶受管金鑰，請依照《AWS Key Management Service 開發人員指南》中[建立對稱客戶受管金鑰](#)的步驟進行。

金鑰政策會控制客戶受管金鑰的存取權限。每個客戶受管金鑰都必須只有一個金鑰政策，其中包含決定誰可以使用金鑰及其使用方式的陳述式。在建立客戶受管金鑰時，可以指定金鑰政策。如需詳細資訊，請參閱《AWS Key Management Service 開發人員指南》中的[管理客戶受管金鑰的存取](#)。

如果您要在與 [StartMedicalScribeJob](#) 請求中指定為 [DataAccessRoleArn](#) 的 IAM 角色相同的帳戶中使用金鑰，您不需要更新金鑰政策。若要在與 [DataAccessRole](#) 不同的帳戶中使用您的客戶受管金鑰，您必須在金鑰政策中信任 [DataAccessRoleArn](#)：才能執行下列動作：

- [kms:Encrypt](#)：允許使用客戶受管金鑰進行加密
- [kms:Decrypt](#)：允許使用客戶受管金鑰進行解密
- [kms:DescribeKey](#)：提供客戶受管金鑰的詳細資訊，讓 AWS HealthScribe 能夠驗證金鑰

以下是範例政策陳述式，您可以新增此範例來對 IAM 角色授予跨帳戶許可，以便使用您的客戶受管金鑰。

```
"Statement" : [
  {
    "Sid": "Allow access to the DataAccessRole for StartMedicalScribeJob",
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::111122223333:role/DataAccessRole"
    },
    "Action": [
      "kms:DescribeKey",
      "kms:Encrypt",
      "kms:Decrypt",
      "kms:GenerateDataKey"
    ],
    "Resource" : "*"
  }
]
```

無論您的客戶受管金鑰和 [DataAccessRole](#) 位於相同帳戶還是不同帳戶中，您的 [DataAccessRole](#) 都需要有許可才能使用客戶受管金鑰來執行上述動作。以下是您可以新增至 [DataAccessRole](#) 的範例政策陳述式：

```
"Statement" : [
  {
    "Sid": "Allow role to perform AWS KMS actions for customer managed key",
    "Effect": "Allow",
    "Action": [
      "kms:DescribeKey",
      "kms:Encrypt",
      "kms:Decrypt"
    ]
  }
]
```

```
  ],  
  "Resource": "*" }  
]
```

如需有關[在政策中指定許可](#)的詳細資訊，請參閱《AWS Key Management Service 開發人員指南》。  
如需有關[故障診斷金鑰存取](#)的詳細資訊，請參閱《AWS Key Management Service 開發人員指南》。

## 為 AWS HealthScribe 指定客戶受管金鑰

您可以指定客戶受管金鑰作為 StartMedicalScribeJob 請求的另一層加密。當您建立 [StartMedicalScribeJob](#) 請求時，您可以在請求中包含 [OutputEncryptionKMSKeyId](#) 欄位以指定客戶受管金鑰。

## AWS KMS 加密內容

AWS KMS 加密內容是純文字對映，非私密 key:value 對。此對映代表其他已驗證的資料 (稱為加密內容配對)，可為您的資料提供一層額外的安全性。AWSHealthScribe 需要對稱加密金鑰，才能將 AWS HealthScribe 輸出加密到客戶指定的 Amazon S3 儲存貯體中。若要了解詳細資訊，請參閱 [AWS KMS 中的非對稱金鑰](#)。

建立加密內容對時，請勿包含敏感資訊。加密內容不會保密 — 在 CloudTrail 日誌中會以純文字顯示 (因此您可以使用它以識別和分類您的加密操作)。加密內容對可以包含特殊字元，例如底線 (\_)、破折號 (-)、斜線 (/、\ ) 和冒號 (:)。

### Tip

將加密內容對中的值與要加密的資料建立關聯會很實用。雖然不是必要的舉動，但我們建議您使用與加密內容相關的非敏感中繼資料，例如檔案名稱、標頭值或未加密的資料庫欄位。若要搭配 API 使用輸出加密，請在 [StartMedicalScribeJob](#) 操作中設定 [KMSEncryptionContext](#) 參數。為了提供輸出加密操作的加密內容，[OutputEncryptionKMSKeyId](#) 參數必須參照對稱 AWS KMS 金鑰 ID。

您可以使用 [AWS KMS 條件金鑰](#) 搭配 IAM 政策，依據 [密碼編譯](#) 操作請求中使用的加密內容，控制對稱加密 AWS KMS 金鑰的存取。如需加密內容政策範例，請參閱 [AWS KMS 加密內容政策](#)。

加密內容是選用項目，但建議使用。如需詳細資訊，請參閱[加密內容](#)。

## 的文件歷史記錄 Amazon Transcribe

- 最新文件更新日期：2023 年 11 月 13 日

下表說明每個發行版本中的重要變更 Amazon Transcribe。如需有關此文件更新的通知，您可以訂閱 RSS 訂閱源。

變更	描述	日期
<a href="#">功能更新</a>	在日記中將最大揚聲器更新為 30 而不是 10。	2024年5月10日
<a href="#">章節更新</a>	更新生成呼叫摘要並新增錯誤輸出詳細資訊。	2024 年 4 月 30 日
<a href="#">章節更新</a>	自訂字彙欄的更新-IPA 和 SoundsLike.	2024 年 4 月 30 日
<a href="#">功能更新</a>	Amazon 轉錄通話分析現在支援生成式呼叫摘要。	2023 年 11 月 29 日
<a href="#">章節更新</a>	更新了新的 PII 修訂和語言識別輸出格式。	2023 年 11 月 13 日
<a href="#">功能更新</a>	日記現在可以與聲道識別結合使用。	2023 年 3 月 6 日
<a href="#">功能更新</a>	聲道識別現在可以與日記結合使用。	2023 年 3 月 6 日
<a href="#">章節更新</a>	IAM 最佳作法已更新。	2023 年 2 月 13 日
<a href="#">新的語言</a>	Amazon Transcribe 現在支持越南語和瑞典語。	2022 年 12 月 6 日
<a href="#">新功能</a>	Amazon Transcribe 現在支持實時呼叫分析。	2022 年 11 月 28 日

<a href="#">功能更新</a>	印地文和泰文現在可使用串流編輯和識別功能。	2022 年 11 月 11 日
<a href="#">章節更新</a>	新的 PII 類別可用於串流修訂和識別。	2022 年 9 月 14 日
<a href="#">章節更新</a>	自訂語言模型一節已修訂。	2022 年 6 月 18 日
<a href="#">章節更新</a>	批次語言識別現在可以識別每個音訊檔案的多種語言。	2022 年 5 月 31 日
<a href="#">指南更新</a>	Amazon Transcribe API 參考現在是一個獨立的指南。	2022 年 4 月 1 日
<a href="#">新增章節</a>	包括「Amazon Transcribe 醫療」和「Amazon Transcribe 呼叫分析」的新比較表。 Amazon Transcribe	2022 年 3 月 21 日
<a href="#">新增章節</a>	隨附新的 SDK 程式碼範例章節。	2022 年 3 月 21 日
<a href="#">功能更新</a>	通話分析現在提供通話摘要。	2022 年 3 月 21 日
<a href="#">章節更新</a>	介紹章節現在展示 Amazon Transcribe 使用案例。	2022 年 3 月 21 日
<a href="#">章節更新</a>	入門章節已更新為針對特定方法。	2022 年 3 月 21 日
<a href="#">章節更新</a>	串流一章已更新並重新架構。	2022 年 3 月 21 日
<a href="#">功能更新</a>	語言識別現在支援自訂詞彙和帶串流轉錄的自訂詞彙篩選。	2022 年 3 月 11 日
<a href="#">新事件</a>	有新的事件類型：詞彙事件。	2022 年 2 月 7 日
<a href="#">章節更新</a>	已更新自訂詞彙一節。	2022 年 1 月 20 日

<a href="#">新功能</a>	語言識別現在可以與串流轉錄搭配使用。	2021 年 11 月 23 日
<a href="#">新功能</a>	語言識別現在可以與自訂語言模型、自訂詞彙、詞彙篩選和內容修訂搭配使用。	2021 年 10 月 29 日
<a href="#">新功能</a>	Amazon Transcribe 現在支援具有串流轉錄功能的自訂語言模型。	2021 年 10 月 20 日
<a href="#">新功能</a>	Amazon Transcribe 現在可以為您的視頻文件生成字幕。	2021 年 9 月 16 日
<a href="#">新功能</a>	Amazon Transcribe 現在支援串流的 PII 編輯和識別。	2021 年 9 月 14 日
<a href="#">新功能</a>	Amazon Transcribe 現在支援 AWS KMS 加密內容，為您的資 AWS 帳戶 源提供額外的安全性。	2021 年 9 月 10 日
<a href="#">新的語言</a>	Amazon Transcribe 現在支援南非荷蘭語、丹麥語、普通話（繁體）、泰語、紐西蘭英語和南非英語。	2021 年 8 月 26 日
<a href="#">新功能</a>	Amazon Transcribe 現在支援資源標記。	2021 年 8 月 24 日
<a href="#">新功能</a>	Amazon Transcribe 現在支援批次轉錄工作的呼叫分析。	2021 年 8 月 4 日
<a href="#">新功能</a>	Amazon Transcribe 現在支援將自訂字彙與批次自訂語言模型搭配使用。	2021 年 5 月 12 日

<a href="#">新功能</a>	Amazon Transcribe 現在支援用於串流轉錄的部分結果穩定功能。	2021 年 5 月 11 日
<a href="#">新功能</a>	Amazon Transcribe 現在支持澳大利亞英語，英國英語，印地語和美國西班牙語自定義語言模型。	2021 年 3 月 19 日
<a href="#">新功能</a>	Amazon Transcribe 現在支持 OGG/OPUS 和 FLAC 編解碼器，用於流式音頻轉錄。	2020 年 11 月 24 日
<a href="#">新的語言</a>	Amazon Transcribe 添加了對意大利語和德語對流音頻轉錄的支持。	2020 年 11 月 4 日
<a href="#">AWS 區域 擴充</a>	Amazon Transcribe 現已在法蘭克福 ( eu-central-1 ) 和倫敦 ( eu-west-2 ) 提供。	2020 年 11 月 4 日
<a href="#">新功能</a>	Amazon Transcribe 在批次轉錄中新增對介面 VPC 端點的支援。	2020 年 10 月 9 日
<a href="#">新功能</a>	Amazon Transcribe 增加了對流媒體中的頻道識別的支持。	2020 年 9 月 17 日
<a href="#">新功能</a>	Amazon Transcribe 在批量轉錄中添加了對自動語言識別的支持。	2020 年 9 月 15 日
<a href="#">新功能</a>	Amazon Transcribe 增加了對流媒體中揚聲器分區的支持。	2020 年 8 月 19 日
<a href="#">新功能</a>	Amazon Transcribe 添加了對自定義語言模型的支持。	2020 年 8 月 5 日

<a href="#">新功能</a>	Amazon Transcribe 在串流中新增對介面 VPC 端點的支援。	2020 年 6 月 26 日
<a href="#">新功能</a>	Amazon Transcribe 添加了對流中詞彙過濾的支持。	2020 年 5 月 20 日
<a href="#">新功能</a>	Amazon Transcribe 添加對自動編輯個人身份信息的支持。	2020 年 2 月 26 日
<a href="#">新功能</a>	Amazon Transcribe 添加了對創建自定義單詞詞彙以從轉錄中過濾的支持。	2019 年 12 月 20 日
<a href="#">新功能</a>	Amazon Transcribe 添加對排隊轉錄作業的支持。	2019 年 12 月 19 日
<a href="#">新的語言</a>	Amazon Transcribe 增加了對海灣阿拉伯語，希伯來語，日本，馬來語，瑞士德語，泰盧固語和土耳其語的支持。	2019 年 11 月 21 日
<a href="#">AWS 區域 擴充</a>	Amazon Transcribe 現已在亞太區域 (東京) (ap-northeast-1) 推出。	2019 年 11 月 21 日
<a href="#">新功能</a>	Amazon Transcribe 添加了對替代轉錄的支持。	2019 年 11 月 20 日
<a href="#">新的語言</a>	Amazon Transcribe 增加了對荷蘭語，波斯語，印度尼西亞語，愛爾蘭英語，葡萄牙語，蘇格蘭英語，泰米爾語和威爾斯英語的支持。	2019 年 11 月 12 日
<a href="#">新的語言</a>	Amazon Transcribe 現在支援澳洲英語 (en-au) 的串流轉錄。	2019 年 10 月 25 日



<a href="#">AWS 區域 擴充</a>	Amazon Transcribe 現已在中國 ( 北京 ) ( CN-北 -1 ) 和中國 ( 寧夏 ) ( CN-西北 -1 ) 上市。	2019 年 10 月 9 日
<a href="#">新功能</a>	Amazon Transcribe 允許您提供自己的轉錄輸出文件 KMS key 來加密。如需詳細資訊，請KeyId參閱 <a href="#">StartStreamTranscription</a> API 的 <a href="#">OutputEncryptionKMS</a> 參數。	2019 年 9 月 24 日
<a href="#">新的語言</a>	Amazon Transcribe 增加了對中文 ( 普通話 ) ，簡體，中國大陸和俄語的支持。	2016 年 8 月 23 日
<a href="#">新功能</a>	Amazon Transcribe 添加了對使用該 WebSocket 協議流式音頻轉錄的支持。	2019 年 7 月 19 日
<a href="#">新功能</a>	AWS CloudTrail 現在會記錄 <a href="#">StartStreamTranscription</a> API 的事件。	2019 年 7 月 19 日
<a href="#">AWS 區域 擴充</a>	Amazon Transcribe 現已在美國西部 (加利佛尼亞北部) (us-west-1) 推出。	2019 年 6 月 27 日
<a href="#">新的語言</a>	Amazon Transcribe 添加對現代標準阿拉伯語的支持。	2019 年 5 月 28 日
<a href="#">新功能</a>	Amazon Transcribe 現在將數字單詞轉錄為美國英語的數字。例如，「四十二」就會轉錄為「42」。	2019 年 5 月 23 日

<a href="#">新的語言</a>	Amazon Transcribe 增加了對印地語和印度英語的支持。	2019 年 5 月 15 日
<a href="#">新的軟體開發套件</a>	適用於 C++ 的 AWS SDK 現在支援 Amazon Transcribe.	2019 年 5 月 8 日
<a href="#">新的語言</a>	Amazon Transcribe 增加對西班牙語的支持。	2019 年 4 月 19 日
<a href="#">AWS 區域 擴充</a>	Amazon Transcribe 現已在歐洲 (法蘭克福) (eu-central-1) 和亞太區域 (首爾) (ap-north-east-2) 推出。	2019 年 4 月 18 日
<a href="#">新的語言</a>	Amazon Transcribe 增加了對英式英語，法語和加拿大法語流轉錄的支持。	2019 年 4 月 5 日
<a href="#">新功能</a>	紅寶石 V3 的 AWS 開發套件現在支援 Amazon Transcribe	2019 年 3 月 25 日
<a href="#">新功能</a>	Amazon Transcribe 允許自定義詞彙，這是您要 Amazon Transcribe 在音頻輸入中識別的特定單詞的列表。	2019 年 3 月 25 日
<a href="#">新的語言</a>	Amazon Transcribe 增加了對德語和韓語的支持。	2019 年 3 月 22 日
<a href="#">新的語言</a>	Amazon Transcribe 現在支援美國西班牙文 (ES-US) 的串流轉錄。	2019 年 2 月 7 日
<a href="#">AWS 區域 擴充</a>	Amazon Transcribe 現已在南美洲 ( 聖保羅 ) ( sa-east-1 ) 上市。	2019 年 2 月 7 日

<a href="#">AWS 區域 擴充</a>	Amazon Transcribe 現已在亞太區域 (孟買) (AI-南 1)、亞太區域 (新加坡) (ap-南部-1)、歐洲 (倫敦) (eu-west-2) 和歐洲 (巴黎) (歐洲-西部 3) 提供。	2019 年 1 月 24 日
<a href="#">新的語言</a>	Amazon Transcribe 新增對法文、義大利文和巴西葡萄牙文的支援。	2018 年 12 月 20 日
<a href="#">新功能</a>	Amazon Transcribe 現在支持音頻流的轉錄。	2018 年 11 月 19 日
<a href="#">新的語言</a>	Amazon Transcribe 增加了對澳大利亞英語，英國英語和加拿大法語的支持。	2018 年 11 月 15 日
<a href="#">AWS 區域 擴充</a>	Amazon Transcribe 現已在加拿大 (中部) (ca-central-1) 和亞太區域 (雪梨) (ap-southeast-2) 提供。	2018 年 7 月 17 日
<a href="#">新功能</a>	您現在可以指定您自己的位置，以存放來自轉錄任務的輸出。	2018 年 7 月 11 日
<a href="#">新功能</a>	添加 AWS CloudTrail 和 Amazon CloudWatch Events 集成。	2018 年 6 月 28 日
<a href="#">新功能</a>	Amazon Transcribe 添加了對自定義詞彙的支持。	2018 年 4 月 4 日
<a href="#">新的指南</a>	這是《Amazon Transcribe 開發人員指南》的第一版。	2017 年 11 月 29 日

# AWS 詞彙表

如需最新的 AWS 術語，請參閱《AWS 詞彙表 參考》中的 [AWS 詞彙表](#)。

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。