



開發人員指南

# Amazon Transcribe



# Amazon Transcribe: 開發人員指南

Copyright © 2026 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 .....	14
字元集 .....	15
阿布哈茲文 .....	19
南非荷蘭文 .....	21
Arabic .....	22
阿斯圖里亞文 .....	23
亞塞拜然文 .....	23
亞美尼亞文 .....	24
巴什基爾文 .....	25
巴斯克文 .....	28
白俄羅斯文 .....	28
孟加拉文 .....	29
波士尼亞文 .....	31
保加利亞文 .....	32
加泰隆尼亞文 .....	33
中庫德文 .....	33
簡體中文 .....	35
繁體中文 .....	36
中文、廣式 .....	37
克羅埃西亞文 .....	37
捷克文 .....	37
丹麥文 .....	38
荷蘭文 .....	39
英文 .....	40
Estonian .....	40
波斯文 .....	40
芬蘭文 .....	42
法文 .....	42

加利西亞文 .....	43
喬治亞文 .....	44
德文 .....	45
Greek .....	46
古吉拉特文 .....	47
豪沙文 .....	49
Hebrew .....	49
北印度文 .....	50
匈牙利文 .....	53
冰島文 .....	53
印尼文 .....	54
義大利文 .....	54
日文 .....	55
卡拜爾文 .....	55
坎那達文 .....	56
哈薩克文 .....	58
盧安達文 .....	59
韓文 .....	60
吉爾吉斯文 .....	60
拉脫維亞文 .....	63
立陶宛文 .....	63
盧干達文 .....	64
馬其頓文 .....	64
馬來文 .....	67
馬來亞拉姆文 .....	67
馬爾他文 .....	69
馬拉地文 .....	70
草原馬里文 .....	72
Mongolian .....	74
書面挪威語 .....	76
歐迪亞文 .....	77
普什圖文 .....	79
Polish .....	81
葡萄牙文 .....	81
旁遮普文 .....	83
羅馬尼亞文 .....	84

俄文 .....	85
塞爾維亞文 .....	86
僧伽羅文 .....	88
斯洛伐克文 .....	91
斯洛維尼亞文 .....	91
索馬利亞文 .....	92
西班牙文 .....	92
巽他文 .....	93
史瓦西里文 .....	94
瑞典文 .....	94
他加祿語/菲律賓文 .....	95
坦米爾文 .....	95
韃靼語 .....	97
特拉古 .....	99
Thai .....	101
Turkish .....	103
烏克蘭文 .....	104
維吾爾語 .....	105
烏茲別克文 .....	108
越南文 .....	109
威爾斯文 .....	113
沃洛夫文 .....	114
祖魯文 .....	115
運作方式 .....	116
資料輸入和輸出 .....	116
媒體格式 .....	117
音訊聲道 .....	118
取樣率： .....	118
輸出 .....	118
轉錄數字 .....	122
開始使用 .....	123
註冊 AWS 帳戶 .....	123
安裝 AWS CLI 和 SDKs .....	124
設定 IAM 登入資料 .....	124
建立 儲存 Amazon S3 貯體 .....	125
建立 IAM 政策 .....	125

使用 轉錄 AWS 管理主控台 .....	127
使用 轉錄 AWS CLI .....	137
開始新的轉錄作業 .....	137
取得轉錄作業的狀態 .....	138
列出您的轉錄作業 .....	139
刪除轉錄作業 .....	140
使用 AWS SDKs 轉錄 .....	140
使用 AWS SDKs .....	152
使用 HTTP 或 WebSockets 進行轉錄 .....	153
串流轉錄 .....	156
最佳實務 .....	157
串流和部分結果 .....	157
部分結果穩定 .....	159
設定串流轉錄 .....	163
事件串流編碼 .....	176
資料框架 .....	178
工作佇列 .....	180
啟用工作佇列 .....	180
標記 資源 .....	184
標籤式存取控制 .....	185
將標籤新增至 Amazon Transcribe 資源 .....	185
分隔發言者 (日記) .....	189
在批次轉錄中對發言者進行分隔 .....	189
在串流轉錄中分隔發言者 .....	193
範例輸出 .....	195
轉錄多聲道音訊 .....	202
批次轉錄中使用聲道識別 .....	203
在串流轉錄中使用聲道識別 .....	207
範例輸出 .....	208
識別語言 .....	217
批次語言識別 .....	217
識別多語言音訊中的語言 .....	218
提高語言識別準確性 .....	219
將語言識別與其他 Amazon Transcribe 功能結合 .....	219
使用語言識別與批次轉錄 .....	220
串流語言識別 .....	226

識別多語言音訊中的語言 .....	227
透過串流媒體使用語言識別 .....	228
替代轉錄 .....	234
請求替代轉錄 .....	236
提高轉錄準確性 .....	241
自訂詞彙 .....	242
自訂詞彙表與清單 .....	243
使用資料表建立自訂詞彙 .....	243
使用清單建立自訂詞彙 .....	253
使用自訂的詞彙 .....	256
自訂語言模型 .....	263
資料來源 .....	263
訓練與調整資料 .....	264
建立自訂語言模型 .....	264
使用自訂語言模型 .....	269
篩選單字 .....	277
建立詞彙篩選條件 .....	278
建立自訂詞彙篩選 .....	279
使用自訂詞彙篩選 .....	283
在批次轉錄中使用自訂詞彙篩選 .....	256
在串流轉錄中使用自訂詞彙篩選 .....	260
偵測有毒語音 .....	291
使用有毒語音偵測 .....	292
在批次轉錄中使用有毒語音偵測 .....	292
範例輸出 .....	295
修訂文字記錄 .....	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 .....	311
編輯輸出的範例 (批次處理) .....	311
編輯流輸出範例 .....	314
PII 識別輸出範例 .....	315
建立字幕 .....	317
產生字幕檔案 .....	318
分析客服中心音訊 .....	321
常用案例 .....	321

考量事項和其他資訊 .....	322
區域可用性和配額 .....	323
通話後分析 .....	324
通話後見解 .....	324
建立類別 .....	326
開始轉錄 .....	338
通話後分析輸出 .....	346
啟用生成式通話摘要 .....	359
即時通話分析 .....	363
即時見解 .....	363
建立類別 .....	367
具有即時轉錄功能的通話後分析 .....	374
開始轉錄 .....	381
即時通話分析輸出 .....	389
轉錄 Amazon Chime 呼叫 .....	395
程式碼範例 .....	396
基本概念 .....	397
動作 .....	397
案例 .....	467
建置 Amazon Transcribe 串流應用程式 .....	467
將文字轉換為語音然後返回文字 .....	468
建立和優化自訂詞彙 .....	469
轉錄音訊並取得工作資料 .....	479
安全 .....	490
身分和存取權管理 .....	490
目標對象 .....	491
使用身分驗證 .....	491
使用政策管理存取權 .....	492
Amazon Transcribe 如何使用 IAM .....	494
預防混淆代理人 .....	498
身分型政策範例 .....	499
疑難排解 .....	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以改善服務 .....	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 .....	526
恢復能力 .....	526
基礎架構安全 .....	526
漏洞分析和管理的 .....	527
VPC 端點 (AWS PrivateLink) .....	527
共用子網路 .....	529
安全最佳實務 .....	529
Amazon Transcribe 醫療 .....	531
區域可用性和配額 .....	532
醫學專科 .....	533
轉錄醫療術語及衡量值 .....	534
轉錄數字 .....	536
轉錄醫療對話 .....	538
轉錄音訊檔案 .....	539
轉錄即時串流 .....	543
啟用發言者分隔 .....	546
轉錄多聲道音訊 .....	555
轉錄醫療聽寫 .....	562
轉錄音訊檔案 .....	563
轉錄串流醫療聽寫 .....	567
建立和使用醫療自訂詞彙 .....	570
為您的醫療自訂詞彙建立文字檔案 .....	570
使用文字檔案建立醫學自訂詞彙 .....	574
使用醫學自訂詞彙轉錄音訊檔案 .....	576
使用醫學自訂詞彙轉錄即時串流 .....	578
Amazon Transcribe Medical 的字元集 .....	580
識別文字記錄中的 PHI .....	582
識別音訊檔案中的 PHI .....	583
識別即時串流中的 PHI .....	588
產生替代轉錄 .....	589
VPC 端點 (AWS PrivateLink) .....	592
Amazon Transcribe Medical VPC 端點的考量事項 .....	592

建立 Amazon Transcribe Medical 的介面 VPC 端點 .....	592
為 Amazon Transcribe Medical 串流建立 VPC 端點政策 .....	593
共用子網路 .....	594
AWS HealthScribe .....	595
安全 .....	596
服務可用性 .....	596
技術需求 .....	596
支援的醫療專科 .....	596
工作流程 .....	597
文字記錄檔案 .....	598
臨床文件檔案 .....	599
Medical Scribe 內容 .....	599
HISTORY_AND_PHYSICAL 範本區段 .....	600
GIRPP 範本區段 .....	601
BIRP 範本區段 .....	601
SIRP 範本區段 .....	602
DAP 範本區段 .....	602
BH_SOAP 範本區段 .....	603
PH_SOAP 範本區段 .....	603
轉錄作業 .....	604
啟動 AWS HealthScribe 轉錄任務 .....	604
串流 .....	619
準則和要求 .....	619
ResourceAccessRoleArn 角色許可 .....	620
Starting AWS HealthScribe 串流轉錄 .....	621
for AWS HealthScribe 靜態資料加密 .....	638
建立客戶受管金鑰 .....	639
指定 AWS HealthScribe 的客戶受管金鑰 .....	643
AWS KMS 加密內容 .....	643
監控 AWS HealthScribe 的加密金鑰 .....	646
文件歷史紀錄 .....	649
AWS 詞彙表 .....	658
.....	dclix

# 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。您可以使用 Amazon Transcribe 做為獨立轉錄服務，或將 speech-to-text 功能新增至任何應用程式。

透過 Amazon Transcribe，您可以使用語言自訂來提高特定使用案例的準確性、篩選內容以確保客戶隱私權或適合受眾的語言、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 儲存貯體（批次）中的媒體檔案。若要查看每種類型的轉錄支援的語言，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](#)

如需此服務的簡短影片導覽，請參閱 [什麼是 Amazon Transcribe ？](#)。

如需了解詳細資訊，請參閱 [Amazon Transcribe 運作方式](#) 和 [入門 Amazon Transcribe](#)。

### Tip

Amazon Transcribe API 的相關資訊位於 [《API 參考資料》](#) 中。

## Amazon Transcribe 和 HIPAA 資格

Amazon Transcribe 涵蓋 AWS 在 HIPAA 資格和 BAA 下，這需要 BAA 客戶在使用時加密所有靜態和傳輸中的 PHI。自動 PHI 識別功能可在 Amazon Transcribe 運作的所有區域中免費使用。如需詳細資訊，請參閱 [HIPAA 資格和 BAA](#)。

## 定價

Amazon Transcribe 是一項 pay-as-you-go 服務；定價是以轉錄音訊的秒數為基礎，每月計費。

用量以 1 秒遞增方式計費，最低計費標準為 15 秒。請注意，PII 內容修訂和自訂語言模型等功能需要支付額外費用。


如需每個的成本資訊 AWS 區域，請參閱 [Amazon Transcribe 定價](#)。

## 區域可用性和配額

Amazon Transcribe 支援以下項目 AWS 區域：

區域	轉錄類型
af-south-1(開普敦)	批次，串流
ap-east-1(香港)	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，串流
ap-southeast-1(新加坡)	批次，串流
ap-southeast-2(雪梨)	批次，串流
ap-southeast-5 ( 馬來西亞 )	串流
ap-southeast-7 ( 泰國 )	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-central-2 ( 蘇黎世 )	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(倫敦)	批次，串流
eu-west-3(巴黎)	批次
me-south-1(巴林)	批次

區域	轉錄類型
mx-central-1 ( 墨西哥 )	串流
sa-east-1(聖保羅)	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(舊金山)	批次
us-west-2(奧勒岡)	批次，串流

 Important

Amazon Transcribe、[Amazon Transcribe Medical](#)和 [呼叫分析](#)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。


# Amazon Transcribe features

為了協助您決定最適合使用案例 Amazon Transcribe 的解決方案，下表提供功能比較。

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 呼叫分析</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 呼叫分析</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 醫療僅提供美式英文。

<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

支援的語言 Amazon Transcribe 列在下表中；也列出特定語言的功能。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 Amazon Transcribe 功能的完整清單，請參閱[功能摘要](#)。

在下表中，「批次」是指轉錄位於 Amazon S3 儲存貯體中的媒體檔案，而「串流」是指即時轉錄串流媒體。對於通話分析轉錄，'post-call' 是指轉錄位於 Amazon S3 儲存貯體中的媒體檔案，而 '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="#">轉錄數字</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，串流	編號	批次，串流	編號	編號	編號
<a href="#">Arabic, 海灣</a>	ar-AE	批次，串流	編號	編號	編號	編號	post-call
<a href="#">Arabic, 現代標準</a>	ar-SA	批次，串流	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

語言	語言代碼	資料輸入	轉錄數字	縮寫	自訂語言模型	修訂	通話分析 * -
<a href="#">巴斯克文</a>	eu-ES	批次，串流	編號	批次，串流	編號	編號	編號
<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，串流	串流	批次，串流	編號	編號	編號
<a href="#">中庫德文，伊朗</a>	ckb-IR	批次	編號	批次	編號	編號	編號
<a href="#">中庫德文，伊拉克</a>	ckb-IQ	批次	編號	批次	編號	編號	編號
<a href="#">中文、廣式</a>	zh-HK (yue-HK)	批次，串流	編號	編號	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，串流	編號	編號	編號	編號	編號
<a href="#">克羅埃西亞文</a>	hr-HR	批次，串流	編號	批次，串流	編號	編號	編號

語言	語言代碼	資料輸入	轉錄數字	縮寫	自訂語言模型	修訂	通話分析 * -
<a href="#">捷克文</a>	cs-CZ	批次，串流	編號	批次，串流	編號	編號	編號
<a href="#">丹麥文</a>	da-DK	批次，串流	編號	批次，串流	編號	編號	編號
<a href="#">荷蘭文</a>	nl-NL	批次，串流	編號	批次，串流	編號	編號	編號
<a href="#">英文</a> ，澳洲	en-AU	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> ，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，串流	批次，串流	批次，串流	編號	串流	post-call
<a href="#">英文</a> ，愛爾蘭	en-IE	批次，串流	批次，串流	批次，串流	編號	串流	post-call
<a href="#">英文</a> ，紐西蘭	en-NZ	批次，串流	批次，串流	批次，串流	編號	串流	編號
<a href="#">英文</a> ，蘇格蘭	en-AB	批次，串流	批次，串流	批次，串流	編號	串流	post-call
<a href="#">英文</a> ，南非	en-ZA	批次，串流	批次，串流	批次，串流	編號	串流	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，串流	批次，串流	批次，串流	編號	串流	post-call
<a href="#">Estonian</a>	et-EE	批次	編號	批次	編號	編號	編號

語言	語言代碼	資料輸入	轉錄數字	縮寫	自訂語言模型	修訂	通話分析 * -
<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, 串流	編號	編號	編號	編號	編號
<a href="#">芬蘭文</a>	fi-FI	批次, 串流	編號	批次, 串流	編號	編號	編號
<a href="#">法文</a>	fr-FR	批次, 串流	批次, 串流	批次, 串流	編號	串流	post-call, real-time
<a href="#">法文, 加拿大</a>	fr-CA	批次, 串流	批次, 串流	批次, 串流	編號	串流	post-call, real-time
<a href="#">加利西亞文</a>	gl-ES	批次, 串流	編號	批次, 串流	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	串流	post-call, real-time
<a href="#">德文, 瑞士</a>	de-CH	批次, 串流	批次, 串流	批次, 串流	編號	串流	post-call
<a href="#">Greek</a>	el-GR	批次, 串流	編號	批次, 串流	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, 串流	編號	編號	編號	編號	編號

語言	語言代碼	資料輸入	轉錄數字	縮寫	自訂語言模型	修訂	通話分析 * -
<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, 串流	編號	批次, 串流	編號	編號	編號
<a href="#">義大利文</a>	it-IT	批次, 串流	批次, 串流	批次, 串流	編號	串流	post-call, real-time
<a href="#">日文</a>	ja-JP	批次, 串流	批次, 串流	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
<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, 串流	編號	批次, 串流	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

語言	語言代碼	資料輸入	轉錄數字	縮寫	自訂語言模型	修訂	通話分析 * -
<a href="#">馬來文</a>	ms-MY	批次，串流	編號	批次，串流	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，串流	編號	批次，串流	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
<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，串流	編號	批次，串流	編號	編號	編號
<a href="#">葡萄牙文</a>	pt-PT	批次，串流	批次，串流	批次，串流	編號	串流	post-call
<a href="#">葡萄牙文, 巴西</a>	pt-BR	批次，串流	批次，串流	批次，串流	編號	串流	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，串流	編號	批次，串流	編號	編號	編號

語言	語言代碼	資料輸入	轉錄數字	縮寫	自訂語言模型	修訂	通話分析 * -
<a href="#">俄文</a>	ru-RU	批次，串流	編號	編號	編號	編號	編號
<a href="#">塞爾維亞文</a>	sr-RS	批次，串流	編號	批次，串流	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，串流	編號	批次，串流	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，串流	編號	批次，串流	編號	編號	編號
<a href="#">西班牙文</a>	es-ES	批次，串流	批次，串流	批次，串流	編號	串流	post-call
<a href="#">西班牙文，美國</a>	es-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="#">巽他文</a>	su-ID	批次	編號	批次	編號	編號	編號
<a href="#">史瓦西里文，肯亞</a>	sw-KE	批次	編號	批次	編號	編號	編號
<a href="#">史瓦西里文，蒲隆地</a>	sw-BI	批次	編號	批次	編號	編號	編號
<a href="#">史瓦西里文，盧安達</a>	sw-RW	批次	編號	批次	編號	編號	編號

語言	語言代碼	資料輸入	轉錄數字	縮寫	自訂語言模型	修訂	通話分析 * -
<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，串流	編號	批次，串流	編號	編號	編號
<a href="#">他加祿語/菲律賓文</a>	tl-PH	批次，串流	編號	批次，串流	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="#">韃靼語</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，串流	編號	批次，串流	編號	編號	編號
<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，串流	編號	批次，串流	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

語言	語言代碼	資料輸入	轉錄數字	縮寫	自訂語言模型	修訂	通話分析 * -
<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，串流	編號	批次，串流	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 SDKs：

批次轉錄	串流轉錄
<a href="#">.NET</a>	<a href="#">.NET</a> (不支援 Transcribe Medical 和 HealthScribe)
<a href="#">AWS 命令列界面 (CLI)</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>
<a href="#">PHP V3</a>	串流不支援 SDK。
<a href="#">Python Boto3</a>	<a href="#">適用於的 Python 串流 SDK Amazon Transcribe</a>
<a href="#">Ruby V3</a>	<a href="#">Ruby V3</a>

批次轉錄	串流轉錄
<a href="#">Rust</a>	<a href="#">Rust</a>

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

如需所有可用 AWS SDKs 和建置器工具的詳細資訊，請參閱 [要建置的工具 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

對於每種語言 Amazon Transcribe 支援，都有一組特定的字元 Amazon Transcribe 可以辨識。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, 國語 \(中國大陸\), 簡體字集](#)
- [中文, 國語 \(台灣\), 繁體字集](#)
- [中文、廣式 \(香港\)、傳統字元集](#)
- [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)

您也可以在此 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	ѓ	0491
и	0438	ѣ	0493
й	0439	ж	0497
к	043A	з	0499
л	043B	ќ	049B
м	043C	к	049F
н	043D	т	04A1

字元	代碼	字元	代碼
o	043E	һ	04A3
п	043F	Һ	04A5
p	0440	ƣ	04A9
c	0441	ƥ	04AB
т	0442	Ƨ	04AD
y	0443	Ʃ	04AF
ф	0444	ƪ	04B1
x	0445	Ƭ	04B3
ц	0446	Ʈ	04B5
ч	0447	ƺ	04B7
ш	0448	h	04BB
щ	0449	ё	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
ë	0451	й	04E3

字元	代碼	字元	代碼
ħ	0452	ö	04E7
í	0453	ø	04E9
ε	0454	ÿ	04EF
s	0455	ÿ	04F1
i	0456	ÿ	04F3
ï	0457	#	04F7
j	0458	ÿ	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)

您也可以在此 Phrase 欄位中使用下列 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
ë	00EB	ú	00FA

字元	代碼	字元	代碼
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
خ	062E	ن	0646

字元	代碼	字元	代碼
د	062F	ه	0647
ذ	0630	و	0648
ر	0631	س	0649
ز	0632	ص	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)

您也可以在此欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
ä	00E4	ğ	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)

您也可以在此欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
մ	0561	ւ	0574
բ	0562	յ	0575
գ	0563	և	0576
դ	0564	ժ	0577
ե	0565	ռ	0578
զ	0566	ճ	0579
է	0567	ղ	057A

字元	代碼	字元	代碼
ƀ	0568	Ɓ	057B
ƃ	0569	Ƅ	057C
Ɔ	056A	ƅ	057D
Ɔ	056B	Ɔ	057E
Ɔ	056C	Ɔ	057F
Ɔ	056D	Ɔ	0580
Ɔ	056E	Ɔ	0581
Ɔ	056F	Ɔ	0582
Ɔ	0570	Ɔ	0583
Ɔ	0571	Ɔ	0584
Ɔ	0572	Ɔ	0585
Ɔ	0573	Ɔ	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)

您也可以在此欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
а	0430	Ӏ	0459

字元	代碼	字元	代碼
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
у	0443	Ѵ	04AF
ф	0444	Ѷ	04B1

字元	代碼	字元	代碼
x	0445	ҁ	04B3
ц	0446	ц	04B5
ч	0447	ч	04B7
ш	0448	h	04BB
щ	0449	ё	04BD
ъ	044A	ё	04BF
ы	044B	#	04CA
ь	044C	ă	04D1
э	044D	ă	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	Ы	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)

您也可以在此欄位中使用下列 Unicode 字元：

字元	代碼	字元	代碼
a	0430	c	0441
б	0431	т	0442
в	0432	y	0443

字元	代碼	字元	代碼
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
ঙ	0999	†	09BF
চ	099A	†	09C0

字元	代碼	字元	代碼
ছ	099B	৞	09C1
জ	099C	৞	09C2
ঝ	099D	৞	09C3
ঞ	099E	৞	09C4
ট	099F	৞	09C7
ঠ	09A0	৞	09C8
ড	09A1	৞	09CB
ঢ	09A2	৞	09CC
ণ	09A3	৞	09CD
ত	09A4	#	09CE
থ	09A5	৞	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)

您也可以在此 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

字元	代碼	字元	代碼
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
й	0439	ш	0448
к	043A	щ	0449
л	043B	ъ	044A

字元	代碼	字元	代碼
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 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	ژ	06B5
ش	0634	ھ	06BE
ع	0639	ؤ	06C6
غ	063A	ؤ	06C7
ف	0641	ى	06CC
ق	0642	ئ	06CE
ل	0644	ه	06D5

## 中文, 國語 (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 代表使用數字的四個中文色調（簡化）。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 代表使用數字的四個中文色調（繁體）。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	□Υ
音調 2	□Υ´
音調 3	□Υ˘
音調 4	□Υ`

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
##	###`-##		
##	###˘-##´		
####	###´-##´-##`-##˘		

```
##### #`-###`-##'-##~-###` ###
```

## 中文、廣式（香港）、傳統字元集

對於中文（中文）香港自訂詞彙，Phrase 欄位可以使用下列檔案中列出的任何字元：

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

## 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)

您也可以在此 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)

您也可以在此 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	შ	10E7
ი	10D8	ჩ	10E8
კ	10D9	ც	10E9

字元	代碼	字元	代碼
œ	10DA	ß	10EA
ð	10DB	đ	10EB
б	10DC	ѳ	10EC
œ	10DD	ƒ	10ED
ǝ	10DE	б	10EE
ƒ	10DF	Ꝥ	10EF
œ	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	૧	0ABE
૩	0A98	૨	0ABF
૪	0A99	૩	0AC0
૫	0A9A	૪	0AC1
૬	0A9B	૫	0AC2
૭	0A9C	૬	0AC3
૮	0A9D	૭	0AC5
૯	0A9E	૮	0AC7
૦	0A9F	૯	0AC8
૧	0AA0	૧૦	0AC9
૨	0AA1	૧૧	0ACB

字元	代碼	字元	代碼
ᱠ	0AA2	ᱡ	0ACC
ᱢ	0AA3	ᱣ	0ACD
ᱤ	0AA4	ᱥ	0AD0
ᱦ	0AA5	ᱧ	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	ḅ	0253
y	01B4	ḁ	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	◌	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	ढि	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 對應下列字元：

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	ż	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
ఋ	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	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

## 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	ı	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	ѣ	0493
й	0439	џ	0497
к	043A	џ	0499
л	043B	ќ	049B
м	043C	ќ	049F
н	043D	ќ	04A1
о	043E	ћ	04A3
п	043F	ћ	04A5
р	0440	ќ	04A9
с	0441	ќ	04AB
т	0442	ќ	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
ഇയ്യ	0D08	മ	0D2E
ഉ	0D09	യ	0D2F
ഉയ്യ	0D0A	ര	0D30
ഊ	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	ç	010B
ì	00EC	ġ	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
॰	0901	थ	0925
ॱ	0902	द	0926
ॲ	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
ѐ	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	џ	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	ă	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	ý	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	ୠ	0B30
ଏ	0B0F	ଲ	0B32
ଏଁ	0B10	ଳ	0B33
ଓ	0B13	ଶ	0B36
ଓଁ	0B14	ଷ	0B37
କ	0B15	ସ	0B38
ଖ	0B16	ଢ	0B39
ଗ	0B17	.	0B3C
ଘ	0B18		0B3E
ଙ	0B19	^	0B3F
ଚ	0B1A	ୱ	0B40
ଛ	0B1B	୲	0B41
ଜ	0B1C	୳	0B42
ଝ	0B1D	୴	0B43
ଞ	0B1E	୵	0B47
ଟ	0B1F	୶	0B48
ଠ	0B20	୷	0B4B
ଡ	0B21	୸	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)

您也可以使用下列 Unicode 字元：

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

字元	代碼	字元	代碼
ج	062C	ٴ	0651
ح	062D	°	0652
خ	062E	#	0654
د	062F	'	0670
ذ	0630	ٲ	067C
ر	0631	پ	067E
ز	0632	ح	0681
س	0633	ح	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
ઠ	0A19	ટ	0A3E
૨	0A1A	૬	0A3F
૩	0A1B	૭	0A40
૪	0A1C	૮	0A41
૫	0A1D	૯	0A42
૬	0A1E	૦	0A47
૭	0A1F	૧	0A48
૮	0A20	૨	0A4B
૯	0A21	૩	0A4C
૦	0A22	૪	0A4D
૧	0A23	૫	0A5C
૨	0A24	૬	0A70
૩	0A25	૭	0A71
૪	0A26	૮	0A72
૫	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	ѣ	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	ж	0497
и	0438	ѝ	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	ə	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)

您也可以在此欄位中使用下列 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

## 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	ஊ	0BC1
ட	0B9F	ஊ	0BC2
ண	0BA3	ஊ	0BC6
த	0BA4	ஊ	0BC7
ந	0BA8	ஊ	0BC8

字元	代碼	字元	代碼
Ɑ	0BAA	ᄀ	0BCA
ᄂ	0BAE	ᄁ	0BCB
ᄃ	0BAF	ᄂ	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	ѣ	0C25
о	0C02	ѣ	0C26
ঃ	0C03	ѣ	0C27
๓	0C05	ѣ	0C28

字元	代碼	字元	代碼
అ	0C06	ప	0C2A
ఇ	0C07	ఫ	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
ఊ	0C32	ఐ	0C10
ఋ	0C33	ఒ	0C12
ౠ	0C35	ఓ	0C13
ౡ	0C36	ఔ	0C14
ౢ	0C37	క	0C15
ౣ	0C38	ఖ	0C16
౤	0C39	గ	0C17
౥	0C3E	ఘ	0C18
౦	0C3F	ఙ	0C19
౧	0C40	చ	0C1A
౨	0C41	ఛ	0C1B
౩	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	ค	0E25
ข	0E02	ก	0E26
ฃ	0E03	ข	0E27
ค	0E04	ค	0E28
ฅ	0E05	ฅ	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	'	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	ء	064C
أ	0623	=	064D
ؤ	0624	ـ	064E
إ	0625	ء	064F
ئ	0626	ـ	0650
ا	0627	ء	0651
ب	0628	ء	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
ʁ	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	к	043A
м	043C	у	0443
ь	044C	ж	0436
ъ	044A	ф	0493
a	0430		

## 越南字元集

Amazon Transcribe 代表使用數字的越南文六種色調。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
ý	00FD	Ý	00DD

字元	代碼	字元	代碼
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
à	1EA3	À	1EA2
ã	1EA5	Ã	1EA4
á	1EA7	Á	1EA6
ă	1EA9	Â	1EA8
â	1EAB	Ã	1EAA
â	1EAD	Â	1EAC
á	1EAF	Ã	1EAE
ă	1EB1	À	1EB0
â	1EB3	Ã	1EB2
ã	1EB5	Ã	1EB4
ă	1EB7	Ă	1EB6
ê	1EB9	Ê	1EB8
è	1EBB	È	1EBA

字元	代碼	字元	代碼
ẽ	1EBD	ẽ	1EBC
ě	1EBF	Ě	1EBE
è	1EC1	È	1EC0
ẽ	1EC3	Ě	1EC2
ẽ	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
ợ	1EE3	Ợ	1EE2

字元	代碼	字元	代碼
ұ	1EE5	Ў	1EE4
ů	1EE7	Ў	1EE6
ú	1EE9	Ў	1EE8
ù	1EEB	Ў	1EEA
ǔ	1EED	Ў	1EEC
ũ	1EEF	Ў	1EEE
ұ	1EF1	Ў	1EF0
ỳ	1EF3	Ỳ	1EF2
ʏ	1EF5	Ỳ	1EF4
ÿ	1EF7	Ỳ	1EF6
ỹ	1EF9	Ỳ	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
á	00E1	ó	00F3

字元	代碼	字元	代碼
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
ã	00E3	ë	00EB

字元	代碼	字元	代碼
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 運作方式

Amazon Transcribe 使用機器學習模型將語音轉換為文字。

除了轉錄文字外，文字記錄還包含轉錄內容的相關資料，包括每個單字或標點符號的可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：

- 批次轉錄：轉錄已上傳至 Amazon S3 儲存貯體的媒體檔案。您可以使用 [AWS CLI](#)、[AWS 管理主控台](#) 和各種 [AWS SDK](#) 進行批次轉錄。
- 串流轉錄：即時轉錄媒體串流。您可以使用 [AWS 管理主控台](#)、[HTTP/2](#)、[WebSockets](#) 和各種 [AWS SDK](#) 進行串流轉錄。

請注意，批次和串流轉錄的功能和語言支援會不同。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](#)

## API 操作可協助您入門

批次：[StartTranscriptionJob](#)

串流：[StartStreamTranscription](#), [StartStreamTranscriptionWebSocket](#)

## 資料輸入和輸出

Amazon Transcribe 會將音訊資料作為 Amazon S3 儲存貯體或媒體串流中的媒體檔案，並將其轉換為文字資料。

如果您要轉錄存放在 Amazon S3 儲存貯體中的媒體檔案，則您要執行批次轉錄。如果您要轉錄媒體串流，您要執行的是串流轉錄。這兩個流程有不同的規則和要求。

如果是批次轉錄，如果您不需要同時處理所有的轉錄任務，請使用 [工作佇列](#)。這可讓 追蹤您的轉錄任務 Amazon Transcribe ，並在有可用插槽時處理它們。

### Note

Amazon Transcribe 可能會暫時存放您的內容，以持續改善其分析模型的品質。請參閱 [Amazon Transcribe 常見問答集](#)，了解更多資訊。若要請求刪除可能已儲存的內容 Amazon Transcribe，請使用 [開啟案例支援](#)。

## 主題

- [媒體格式](#)
- [音訊聲道](#)
- [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不同，但兩者都建議使用無失真格式。請參閱下列資料表，了解詳細資訊：

	批次	串流
支援的格式	<ul style="list-style-type: none"> <li>• AMR</li> <li>• FLAC</li> <li>• M4A</li> <li>• MP3</li> <li>• MP4</li> <li>• Ogg</li> <li>• WebM</li> <li>• WAV</li> </ul>	<ul style="list-style-type: none"> <li>• FLAC</li> <li>• Ogg Opus</li> <li>• PCM 編碼</li> </ul>
建議格式	<ul style="list-style-type: none"> <li>• FLAC</li> <li>• 具備 PCM 16 位元編碼的 WAV 檔案</li> </ul>	<ul style="list-style-type: none"> <li>• FLAC</li> <li>• PCM 簽署 16 位小端音訊 (請注意，這不包括 WAV)</li> </ul>

為了獲得最佳結果，您應該使用不失真的格式，例如 FLAC 或以 PCM 16 位元編碼的 WAV。

### 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。

## 音訊聲道

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

如果您的音訊在一個聲道上包含多個發言者，而您想要在轉錄輸出中進行每個發言者的分隔和標記，則可以使用[發言者分隔 \(日記\)](#)。

如果您的音訊包含兩個不同聲道的語音，您可以使用[聲道識別](#)功能，在文字記錄中分別轉錄每個聲道。

這兩個選項都會產生文字記錄檔案。

### Note

如果您未啟用[發言者分隔](#)或[聲道識別](#)，您的文字記錄文字會以連續區段的形式提供。

## 取樣率：

使用批次轉錄作業時，您可以選擇提供取樣率，但此參數是選擇性提供。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。與批次轉錄作業相同，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。對於高保真音訊，Amazon Transcribe 支援介於 16,000 Hz 和 48,000 Hz 之間的值。

## 輸出

轉錄輸出為 JSON 格式。文字記錄的第一部分以段落形式包含文字記錄本身，後面接著每個單字和標點符號的其他資料。所提供的資料須視您請求中包含的功能。您的文字記錄至少會包含每個單字的開始時間、結束時間和可信度分數。[以下章節](#)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。

所有批次文字記錄都存放在 Amazon S3 儲存貯體中。您可以選擇將文字記錄儲存在自己的 Amazon S3 儲存貯體中，或使用安全的預設儲存貯 Amazon Transcribe 體。若要了解如何建立和使用 Amazon S3 儲存貯體的詳細資訊，請參閱[使用儲存貯體](#)。

如果您想要將文字記錄存放在您擁有的 Amazon S3 儲存貯體中，請在您的文字記錄請求中指定儲存貯體的 URI。在開始批次轉錄任務之前，請確定您授予此儲存貯體的 Amazon Transcribe 寫入許可。如果您指定自己的儲存貯體，您的文字記錄會保留在該儲存貯體中，直到您移除為止。

如果您未指定 儲存 Amazon S3 貯體，Amazon Transcribe 會使用安全的服務管理儲存貯體，並為您提供暫時 URI，供您用來下載文字記錄。請注意，臨時 URI 的有效期限為 15 分鐘。如果您使用提供的 URI 時發生 AccessDenied 錯誤，請提出 GetTranscriptionJob 請求以取得文字記錄的新臨時 URI。

如果您選擇預設儲存貯體，則會在工作到期時 (90 天) 刪除您的文字記錄。如果您想在超過到期日仍保留文字記錄，您必須下載。

串流文字記錄會透過您在串流中使用的相同方法傳回。

#### Tip

如果您想將 JSON 輸出轉換為 Word 格式的逐段文字記錄，請參閱此 [GitHub 範例 \(適用於 Python3\)](#)。此指令碼可搭配通話後分析文字記錄和啟用日記的標準批次文字記錄使用。

## 範例輸出

文字記錄以段落形式提供完整的轉錄，後面接著逐字劃分，提供每個單字和標點符號的資料。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](#)的簡單批次轉錄作業。您套用至轉錄請求的每個其他功能時，可以取得文字記錄輸出檔案中的額外資料。

基本批次文字記錄包含兩個主要部分：

1. `transcripts`：一個文字區塊中包含整個文字記錄。
2. `items`：包含 `transcripts` 部分中每個單字和標點符號的資訊。
3. `audio_segments`：音訊區段是音訊錄音的特定部分，其中包含不中斷的口語語言，且停頓或休息時間最少。此區段會擷取自然的語音流程，並在 `audio_segments` 以開始時間和結束時間擷取。音訊區段中的 `items` 元素是一系列與區段中每個項目對應的識別符。

您在轉錄請求中包含的每個其他功能都會在您的文字記錄中產生額外的資訊。

```
{
  "jobName": "my-first-transcription-job",
  "accountId": "111122223333",
  "results": {
    "transcripts": [
      {
        "transcript": "Welcome to Amazon Transcribe."
      }
    ],
    "items": [
      {
        "id": 0,
        "start_time": "0.64",
        "end_time": "1.09",
        "alternatives": [
          {
            "confidence": "1.0",
            "content": "Welcome"
          }
        ],
        "type": "pronunciation"
      },
      {
        "id": 1,
        "start_time": "1.09",
        "end_time": "1.21",
        "alternatives": [
          {
            "confidence": "1.0",
            "content": "to"
          }
        ],
        "type": "pronunciation"
      },
      {
        "id": 2,
        "start_time": "1.21",
        "end_time": "1.74",
        "alternatives": [
          {
            "confidence": "1.0",
            "content": "Amazon"
          }
        ]
      }
    ]
  }
}
```

```
    }
  ],
  "type": "pronunciation"
},
{
  "id": 3,
  "start_time": "1.74",
  "end_time": "2.56",
  "alternatives": [
    {
      "confidence": "1.0",
      "content": "Transcribe"
    }
  ],
  "type": "pronunciation"
},
{
  "id": 4,
  "alternatives": [
    {
      "confidence": "0.0",
      "content": "."
    }
  ],
  "type": "punctuation"
}
],
"audio_segments": [
  {
    "id": 0,
    "transcript": "Welcome to Amazon Transcribe.",
    "start_time": "0.64",
    "end_time": "2.56",
    "items": [
      0,
      1,
      2,
      3,
      4
    ]
  }
]
},
"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，並針對在其撰寫系統中使用案例區分的語言適當地大寫單字。

對於多數語言，數字會轉錄為該語言的單字形式。不過，對於支援轉錄數字的語言，會根據所使用的內容，以不同的方式 Amazon Transcribe 處理數字。

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：

- 支援數字轉錄的語言：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

若要檢視支援轉錄號碼的語言，請參閱 [支援的語言和特定語言功能](#)。

# 入門 Amazon Transcribe

建立轉錄前，您必須符合下列幾項先決條件：

- [註冊 AWS 帳戶](#)
- [安裝 AWS CLI 和 SDKs](#)（如果您使用 AWS 管理主控台 進行轉錄，可以略過此步驟）
- [配置 IAM 憑證](#)
- [設定 Amazon S3 儲存貯體](#)
- [建立 IAM 政策](#)

完成這些先決條件後，您即可進行轉錄。從以下清單中選擇您偏好的轉錄方法以開始使用。

- [AWS CLI](#)
- [AWS 管理主控台](#)
- [AWS 開發套件](#)
- [HTTP](#)
- [WebSockets](#)

## Tip

如果您是初次使用 Amazon Transcribe 或想要探索我們的功能，建議您使用 [AWS 管理主控台](#)。如果您想使用電腦麥克風開始串流，這也是最簡單的選擇。

由於使用 HTTP/2 和 WebSockets 進行串流比其他轉錄方法複雜，因此建議您在開始使用這些方法前，參考 [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](#)。這兩個選項都可讓您存取所有 AWS 服務。免費方案有試用期，在此期間您可以探索 AWS 服務 和估計您的用量。試用期到期後，您可以轉移到付費帳戶。費用是按用量付費計算；請參閱 [Amazon Transcribe 定價](#)，了解詳細資訊。

**i** Tip

設定您的帳戶時，請記下您的 AWS 帳戶 ID，因為您需要它來建立 IAM 實體。

## 安裝 AWS CLI 和 SDKs

若要使用 Amazon Transcribe API，您必須先安裝 AWS CLI。目前 AWS CLI 版本是 2。您可以在[AWS Command Line Interface 《使用者指南》](#)中找到 [Linux](#)、[Mac](#)、[Windows](#) 和 [Docker](#) 的安裝說明。

安裝之後 AWS CLI，您必須針對安全登入資料和進行[設定](#) AWS 區域。

如果您想要 Amazon Transcribe 搭配 SDK 使用，請選取您偏好的語言以取得安裝指示：

- [.NET](#)
- [C++](#)
- [Go](#)
- [Java V2](#)
- [JavaScript](#)
- [PHP V3](#)
- [適用於 Python \(Boto3\) 的 AWS SDK\(批次轉錄\)](#)
- [Python\(串流轉錄\)](#)
- [Ruby V3](#)
- [Rust\(批量轉錄\)](#)
- [Rust\(串流轉錄\)](#)

## 設定 IAM 登入資料

當您建立時 AWS 帳戶，您會從一個登入身分開始，該身分可完整存取您帳戶中的所有 AWS 服務和資源。此身分稱為 AWS 帳戶 根使用者，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。

根據最佳實務，要求使用者 - 包括需要管理員存取權的使用者 - 使用聯合身分提供者來使用臨時憑證來存取 AWS 服務。

聯合身分是指使用透過身分來源提供的登入資料存取 AWS 服務的任何使用者。聯合身分存取 AWS 帳戶時，會擔任角色，並由角色提供臨時憑證。

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。或者，您可以連接並同步到您自己的身分來源中的一組使用者和群組，以用於所有 AWS 帳戶和應用程式。如需詳細資訊，請參閱的 [Identity and Access Management Amazon Transcribe](#)。

若要進一步了解 IAM 最佳實務，請參閱 [中的安全最佳實務 IAM](#)。

## 建立 儲存 Amazon S3 貯體

Amazon S3 是一種安全的物件儲存服務。會將您的檔案（稱為物件）Amazon S3 存放在容器（稱為儲存貯體）中。

若要執行批次轉錄，您必須先將媒體檔案上傳至 儲存 Amazon S3 貯體。如果您未為轉錄輸出指定 儲存 Amazon S3 貯體，會將您的轉錄 Amazon Transcribe 放入暫時 AWS 管理的儲存 Amazon S3 貯體。90 天後會自動刪除受 AWS 管儲存貯體中的轉錄輸出。

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 AWS，您必須建立政策，並將其連接到 IAM 身分（使用者、群組或角色）或 AWS 資源。政策會定義其所連接之實體的權限。例如，如果您已將政策連接至授予其存取權的該角色，則角色只能存取位於儲存 Amazon S3 貯體中的媒體檔案。如果您想要進一步限制該角色，您可以改為限制其對 Amazon S3 儲存貯體中特定檔案的存取。

若要進一步了解如何使用 AWS 政策，請參閱：

- [中的政策和許可 IAM](#)
- [建立 IAM 政策](#)
- [Amazon Transcribe 如何使用 IAM](#)

如需可與 搭配使用的政策範例 Amazon Transcribe，請參閱 [Amazon Transcribe 身分型政策範例](#)。如果您要產生自訂政策，請考慮使用 [AWS 政策產生器](#)。

您可以使用 AWS 管理主控台、AWS CLI 或 AWS SDK 新增政策。如需說明，請參閱 [新增和移除 IAM 身分許可](#)。

政策的格式如下：

JSON

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

Amazon Resource Name (ARNs) 可唯一識別所有 AWS 資源，例如儲存 Amazon S3 貯體。您可以在政策中使用 ARN 以授與使用特定資源的特定動作的權限。例如，如果您想要授予儲存 Amazon S3 貯體及其子資料夾的讀取存取權，您可以將下列程式碼新增至信任政策的 Statement 區段：

```
{
  "Effect": "Allow",
  "Action": [
    "s3:GetObject",
    "s3:ListBucket"
  ],
  "Resource": [
    "arn:aws:s3:::amzn-s3-demo-bucket",
    "arn:aws:s3:::amzn-s3-demo-bucket/*"
  ]
}
```

以下是將 Amazon Transcribe 讀取 (GetObject、ListBucket) 和寫入 (PutObject) 許可授予 儲存 Amazon S3 貯體 `amzn-s3-demo-bucket`、及其子資料夾的範例政策：

JSON

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

## 使用 轉錄 AWS 管理主控台

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 儲存貯體中的媒體檔案，則您要執行批次轉錄。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。

開始批次轉錄之前，您必須先將媒體檔案上傳至 儲存 Amazon S3 貯體。若要使用 串流轉錄 AWS 管理主控台，您必須使用電腦麥克風。

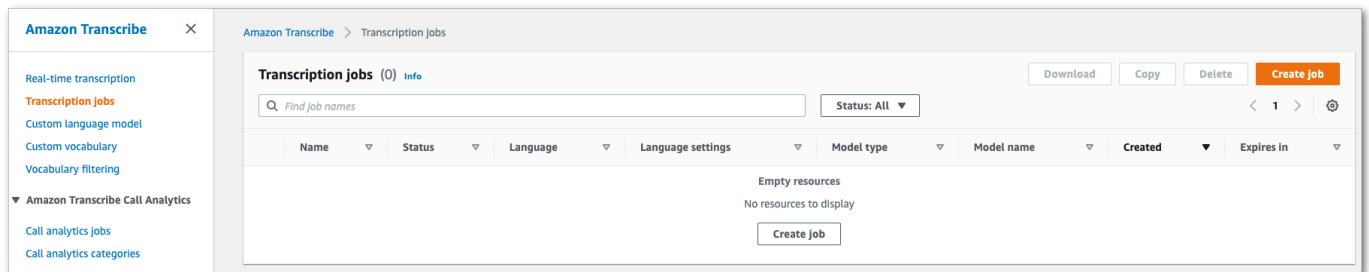
若要檢視支援的媒體格式和其他媒體需求和限制，請參閱 [資料輸入和輸出](#)。

展開下列章節，以取得每個轉錄方法的簡短逐步解說。

## 批次轉錄

首先，請確定您已上傳要轉錄到儲存貯體中的媒體檔案。Amazon S3 如果您不確定如何執行此操作，請參閱《Amazon S3 使用者指南》：[將物件上傳到儲存貯體](#)。

1. 從 [AWS 管理主控台](#)，在左側導覽窗格中，選擇 訓練工作。將引導您前往轉錄作業清單。



選擇建立作業。

2. 完成指定作業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 儲存 Amazon S3 貯體中的物件。對於輸出位置，您可以選擇安全 Amazon S3 的服務受管儲存貯體，也可以指定自己的儲存 Amazon S3 貯體。

如果您選擇服務管理的儲存貯體，則可以在 AWS 管理主控台 中檢視文字記錄預覽，也可以從任務詳細資訊頁面下載文字記錄 (請參閱下方)。

如果您選擇自己的儲存 Amazon S3 貯體，您就無法在 中看到預覽，AWS 管理主控台 而且必須前往儲存 Amazon S3 貯體下載您的文字記錄。

### 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.

選擇下一步。

3. 在 **配置工作** 頁面上選擇任何所需的選項。如果要使用 [自訂詞彙](#) 或 [自訂語言模型](#) 與轉錄搭配使用，則必須在開始轉錄作業前建立這些內容。

## 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

選擇建立作業。

- 您現在位於 **轉錄作業** 頁面。在此處您可以看到轉錄作業的狀態。完成後，選擇您的轉錄。

Transcription jobs (5) [Info](#) Download Copy Delete Create job

Status: All < 1 > ⚙️

Name	Status	Language	Language settings	Model type	Model name	Created	Expires in
<input type="radio"/> my-first-transcription	<span>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 **任務詳細資訊** 頁面。在此處，您可以檢視設定轉錄作業時指定的所有選項。

若要檢視文字記錄，請在 **輸出資料位置** 下的右欄中選擇連結的檔案路徑。這會將您導向至您指定的 Amazon S3 輸出資料夾。選擇您的輸出文件，該文件現在具有 .json 副檔名。

my-first-transcription-job			
<b>Job details</b>			
Name	Model	Audio identification	Input data location
my-first-transcription-job	None	Disabled	<a href="s3://DOC-EXAMPLE-BUCKET/my-media-file.flac">s3://DOC-EXAMPLE-BUCKET/my-media-file.flac</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. 下載文字記錄的方式取決於您選擇的是服務受管儲存 Amazon S3 貯體還是您自己的 Amazon S3 儲存貯體。
  - a. 如果您選擇服務管理的儲存貯體，您可以在轉錄作業的資訊頁面上看到 **轉錄預覽** 窗格，以及 **下載** 按鈕。

The screenshot displays the Amazon Transcribe console interface for a transcription job named "my-first-transcription-job". At the top right, there are "Delete" and "Copy" buttons. The main content is divided into two sections: "Job details" and "Transcription preview".

**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>
Status ✔ Complete	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.

選擇 **下載** 並選擇 **下載文字記錄**。

- b. 如果您選擇自己的儲存 Amazon S3 貯體，則不會在轉錄任務資訊頁面上的轉錄預覽窗格中看到任何文字。反之，您會看到藍色資訊方塊，其中包含所選儲存 Amazon S3 貯體的連結。

**my-first-transcription-job**

---

**Job details**

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

---

**Transcription preview**  ▼

Select download to save a local copy of the transcription.

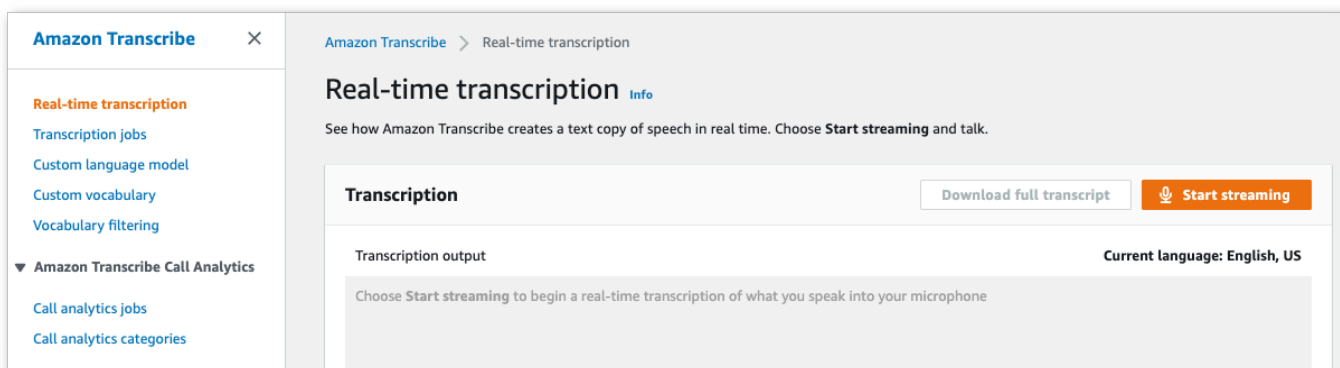
Text
Audio identification
Subtitles

i 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](#). [↗](#)

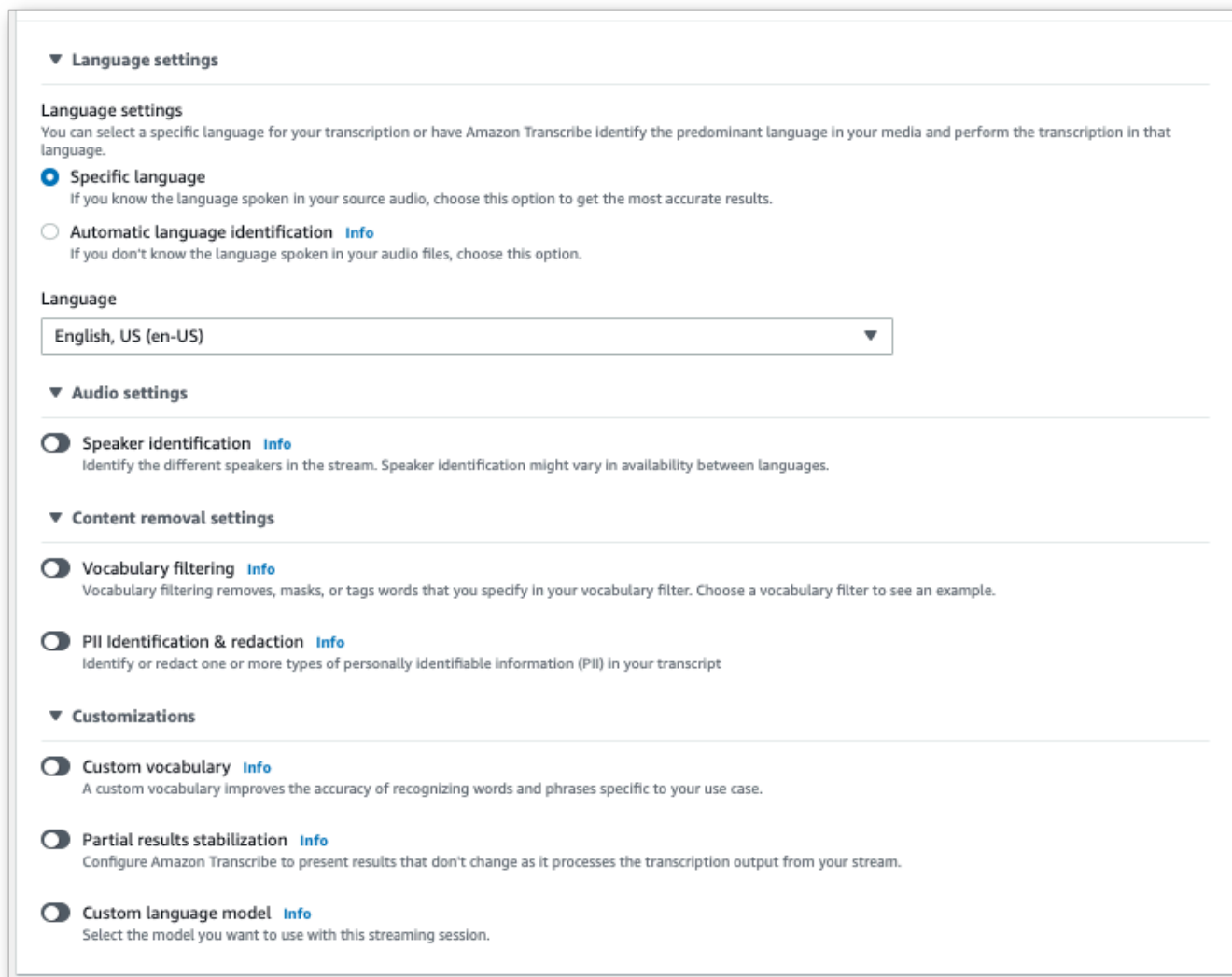
若要存取您的文字記錄，Amazon S3 請使用任務詳細資訊窗格中輸出資料位置下的連結，或文字記錄預覽窗格中藍色資訊方塊中的 S3 儲存貯體連結，前往指定的儲存貯體。

## 串流轉錄

1. 從 [AWS 管理主控台](#)，在左側導覽窗格中，選擇 即時轉錄。這將引導您前往主要的串流媒體頁面，您可以在開始直播前選擇選項。



2. 在轉錄輸出方塊下，您可以選擇各種語言和音訊設定。



3. 選擇適當的設定後，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。您可以即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，選擇停止串流。

## 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.

您現在可以選擇 下載完整文字記錄 以下載文字記錄。

## 使用 轉錄 AWS CLI

使用 AWS CLI 開始轉錄時，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指令，然後執行包含要求主文的 JSON 檔案的 AWS 區域 和位置。本指南的範例展示這兩種方法；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：

- 已將媒體檔案上傳至 Amazon S3 儲存貯體。如果您不確定如何建立儲存 Amazon S3 貯體或上傳檔案，請參閱[建立您的第一個 Amazon S3 儲存貯體](#)並將物件上傳到您的儲存貯體。
- 已安裝 [AWS CLI](#)。

您可以在 AWS CLI 命令參考 Amazon Transcribe 中找到的所有命令。 [AWS CLI](#)

### 開始新的轉錄作業

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

1. 在終端視窗中，執行下列指令：

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

下一行會出現「>」，您現在可以如下一個步驟所述繼續新增必要的參數。

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 `region`、`transcription-job-name`、`media` 和 `language-code` 或 `identify-language`。

如果要指定輸出位置，請在請求中包含 `output-bucket-name`；如果要指定指定輸出儲存貯體的子資料夾，也請包含 `output-key`。

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

如果您選擇不使用 `output-bucket-name` 指定輸出儲存貯體，Amazon Transcribe 會將轉錄輸出放置在服務管理的儲存貯體中。儲存在服務管理儲存貯體中的文字記錄會在 90 天後過期。

Amazon Transcribe 回應方式如下：

```
{
  "TranscriptionJob": {
    "TranscriptionJobName": "my-first-transcription-job",
    "TranscriptionJobStatus": "IN_PROGRESS",
    "LanguageCode": "en-US",
    "Media": {
      "MediaFileUri": "s3://amzn-s3-demo-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 [TranscriptionJobStatus](#) 從 IN\_PROGRESS 變更為 COMPLETED，您的轉錄作業已成功。若要查看更新 [TranscriptionJobStatus](#)，請如下節所示使用 `get-transcription-job` 或 `list-transcription-job` 指令。

## 取得轉錄作業的狀態

如需取得轉錄作業的資訊，請使用 `get-transcription-job` 指令。

此命令的唯一必要參數是 AWS 區域 任務所在的 和任務的名稱。

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

Amazon Transcribe 回應方式如下：

```
{
  "TranscriptionJob": {
    "TranscriptionJobName": "my-first-transcription-job",
    "TranscriptionJobStatus": "COMPLETED",
    "LanguageCode": "en-US",
    "MediaSampleRateHertz": 48000,
    "MediaFormat": "flac",
    "Media": {
      "MediaFileUri": "s3://amzn-s3-demo-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儲存 Amazon S3 貯體，則此儲存貯體會與一起列出 TranscriptFileUri。如果您已選擇服務管理儲存貯體，則會提供暫時 URI；請使用此 URI 下載您的文字記錄。

#### Note

服務受管 Amazon S3 儲存貯體 URIs 有效期只有 15 分鐘。如果您在使用 URI 時收到 AccessDenied 錯誤，請再次執行 get-transcription-job 請求以取得新的臨時 URI。

## 列出您的轉錄作業

若要列出指定 中的所有轉錄任務 AWS 區域，請使用 list-transcription-jobs 命令。

此命令的唯一必要參數是 AWS 區域 轉錄任務所在的。

```
aws transcribe list-transcription-jobs \  
  --region us-west-2
```

Amazon Transcribe 回應方式如下：

```
{  
  "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"  
    }  
  ]  
}
```

## 刪除轉錄作業

若要刪除轉錄作業，請使用 `delete-transcription-job` 指令。

此命令的唯一必要參數是 AWS 區域 任務所在的 和任務的名稱。

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

若要確認刪除請求是否成功，您可以執行 `list-transcription-jobs` 指令。您的工作應該不會再  
在清單中顯示。

## 使用 AWS SDKs 轉錄

您可以將 SDK 用於批次和串流轉錄。如果您要轉錄位於 Amazon S3 儲存貯體中的媒體檔案，則會執行  
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。

如需可與 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，  
AWS SDKs 不支援串流轉錄。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](#)。

如需所有可用 AWS SDKs 和建置器工具的詳細資訊，請參閱 [要建置的工具 AWS](#)。

### Tip

如需使用 AWS SDKs 的其他範例，包括功能特定、案例和跨服務範例，請參閱 [Amazon Transcribe AWS SDKs 的程式碼範例](#) 章節。

您也可以在下述 GitHub 存放庫中找到 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 [建立您的第一個 S3 儲存貯體](#) 並將 [物件上傳到您的儲存貯體](#)。

## 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://amzn-s3-demo-bucket/my-input-files/my-media-
file.flac")
```

```
        .build();

        String outputS3BucketName = "s3://amzn-s3-demo-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://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
  },
  OutputBucketName: "amzn-s3-demo-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://amzn-s3-demo-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，串流適用於 Python (Boto3) 的 AWS SDK 不支援 Amazon Transcribe 標準。若要使用 Python 啟動串流轉錄，請使用此[非同步 Python 開發套件](#)。 [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：[重試用戶端以進行 Amazon Transcribe 串流 \(Java SDK\)](#)。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 軟體開發套件 (SDKs) 適用於許多熱門的程式設計語言。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="#">適用於 C++ 的 AWS SDK</a>	<a href="#">適用於 C++ 的 AWS SDK 程式碼範例</a>
<a href="#">AWS CLI</a>	<a href="#">AWS CLI 程式碼範例</a>
<a href="#">適用於 Go 的 AWS SDK</a>	<a href="#">適用於 Go 的 AWS SDK 程式碼範例</a>
<a href="#">適用於 Java 的 AWS SDK</a>	<a href="#">適用於 Java 的 AWS SDK 程式碼範例</a>
<a href="#">適用於 JavaScript 的 AWS SDK</a>	<a href="#">適用於 JavaScript 的 AWS SDK 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="#">適用於 .NET 的 AWS SDK</a>	<a href="#">適用於 .NET 的 AWS SDK 程式碼範例</a>
<a href="#">適用於 PHP 的 AWS SDK</a>	<a href="#">適用於 PHP 的 AWS SDK 程式碼範例</a>

SDK 文件	代碼範例
<a href="#">AWS Tools for PowerShell</a>	<a href="#">AWS Tools for PowerShell 程式碼範例</a>
<a href="#">適用於 Python (Boto3) 的 AWS SDK</a>	<a href="#">適用於 Python (Boto3) 的 AWS SDK 程式碼範例</a>
<a href="#">適用於 Ruby 的 AWS SDK</a>	<a href="#">適用於 Ruby 的 AWS SDK 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 [Amazon Transcribe AWS SDKs 的程式碼範例](#)。

#### 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 WebSockets 進行轉錄

Amazon Transcribe 支援批次 (HTTP/1.1) 和串流 (HTTP/2) 轉錄的 HTTP。WebSockets 支援串流轉錄。

如果您要轉錄位於 Amazon S3 儲存貯體中的媒體檔案，則您要執行批次轉錄。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。

HTTP 和 WebSockets 都要求您使用 AWS Signature 第 4 版標頭以驗證您的請求。如需詳細資訊，請參閱 [簽署 AWS API 請求](#)。

### 批次轉錄

您可以使用下列標頭提出批次 HTTP 要求：

- 託管
- X-Amz-Target

- content-type
- x-amz-content-sha256
- X-Amz-Date
- 授權

這是 StartTranscriptionJob 範例：

```
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://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
  },
  "OutputBucketName": "amzn-s3-demo-bucket",
  "OutputKey": "my-output-files/"
}
```

其他操作和參數會列在 [API 參考](#) 中；所有 AWS API 操作常用的參數會列在 [通用參數](#) 區段中。其他簽章元素詳述於 [AWS 簽章第 4 版請求的元素](#) 中。

## 串流轉錄

與使用 SDK 相較，使用 HTTP/2 和 WebSockets 的串流轉錄更深入。我們建議您在設定第一個串流前，先查看 [設定串流轉錄](#) 一節。

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

 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 串流，您可以為媒體內容產生即時轉錄。與涉及上傳媒體檔案的批次轉錄不同，串流媒體會 Amazon Transcribe 即時交付至。Amazon Transcribe 然後，也會即時傳回文字記錄。

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)。的常見串流使用案例 Amazon Transcribe 包括運動賽事的即時隱藏式字幕，以及呼叫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語音轉文字功能，以及更快的轉錄時間。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 提供下列串流選項：

- [軟體開發套件](#) (首選)
- [HTTP/2](#)
- [WebSockets](#)
- [AWS 管理主控台](#)

若要轉錄 中的串流音訊 AWS 管理主控台，請對著電腦麥克風說話。

### Tip

如需 SDK 程式碼範例，請參閱 GitHub 上的 [AWS 範例儲存庫](#)。

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區域 Amazon Transcribe 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

由於串流可即時運作，因此會以部分結果產生文字記錄。會根據自然語音區段 Amazon Transcribe 中斷傳入的音訊串流，例如喇叭變更或音訊暫停。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 管理主控台](#)，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。此物件區塊中還有一個 `IsPartial` 欄位。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 一旦您開始串流音訊，就會開始傳回轉錄結果。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。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

部分結果穩定功能會變更 為每個完整區段 Amazon Transcribe 產生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」：
<p>The</p> <p>The</p> <p>The Amazon.</p> <p>The Amazon is</p> <p>The Amazon is the large</p> <p>The Amazon is the largest</p> <p>The Amazon is the largest rainfall.</p> <p>The Amazon is the largest rain forest.</p> <p>The Amazon is the largest rain forest on</p> <p>The Amazon is the largest rain forest on the planet.</p> <p>The Amazon is the largest rain forest on the planet.</p> <p>The Amazon is the largest rain forest on the planet.</p> <p>The Amazon is the largest rain forest on the planet.</p> <p>The Amazon is the largest rain forest on the planet.</p>	<p>0.515</p> <p>1.015</p> <p>1.515</p> <p>2.015</p> <p>2.515</p> <p>3.015</p> <p>3.515</p> <p>4.015</p> <p>4.515</p> <p>5.015</p> <p>5.515</p> <p>6.015</p> <p>6.335</p> <p>6.335</p>	<p>true</p> <p>true</p> <p>true</p> <p>true</p> <p>true</p> <p>true</p> <p>true</p> <p>true</p> <p>true</p> <p>true</p> <p>true</p> <p>true</p> <p>false</p>

當您啟用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' 是指轉錄的字詞或標點符號。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。

### 使用 AWS Machine Learning 部落格深入探索

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。它旨在為想要直接使用 HTTP/2 或 WebSockets 設定串流的使用者提供資訊，而不是使用 AWS SDK。本節中的資訊也可用於建置自己的 SDK。

### Important

我們強烈建議您使用 SDK，而不是直接使用 HTTP/2 和 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 AWS SDKs 轉錄](#)。

## 設定 HTTP/2 串流

使用串流轉錄請求的 [HTTP/2 通訊協定](#) 的關鍵元件 Amazon Transcribe 為：

- 標頭框架。這包含請求的 HTTP/2 標頭，以及授權標頭中的簽章，以 Amazon Transcribe 做為種子簽章來簽署資料影格。
- [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 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. 將下列政策連接至發出請求 IAM 的角色。如需詳細資訊，請參閱[新增 IAM 政策](#)。

## JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "myTranscribeHttp2Policy",
      "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 會傳送下列回應：

```
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
```

```
AAAA0gAAAIKVoRFcTTcjb250ZW50LXR5cGUHABhhcHBsaWNhdGlvbi9vY3RldC1zdHJlYW0L0mV2ZW50LXR5cGUHAAPBdWRpb0V2ZW50DTptZXNzYWdlLXR5cGUHAAV1dmVudAxDb256ZW50LVR5cGUHABphcHBsaWNhdGlvbi94LWFtei1qc29uLTEuMVJJRky88T0AV0FWRWZtdCAQAAAAAQABAIA
+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**：此函數首先建立其輸入的 SHA-256 雜湊，然後使用 Hex 函數進行編碼雜湊。
  - **nonSignatureHeaders**：編碼為字串的 DateTime 標頭。
  - **payload**：包含音訊事件資料的位元組緩衝區。
- c. 從 AWS 私密存取金鑰衍生簽署金鑰，並使用它來簽署 `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。
- **derivedSigningKey**：Signature 第 4 版簽名金鑰。
- **stringToSign**：您針對資料框架計算的字串。

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 會以轉錄事件串流回應，並傳送到您的應用程式。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，會 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	BadRequestException
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 WebSocket 串流

使用串流轉錄請求的 [WebSocket 通訊協定](#) 的關鍵元件 Amazon Transcribe 為：

- 升級請求。這包含請求的查詢參數，以及 Amazon Transcribe 使用做為種子簽章來簽署資料影格的簽章。
- [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. 將下列政策連接至發出請求 IAM 的角色。如需詳細資訊，請參閱 [新增 IAM 政策](#)。

## JSON

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

- 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)。這可確保 AWS 收到請求時，可以計算您為 URL 建立的相同簽章。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。若要這樣做，請從 AWS 私密存取金鑰衍生簽署金鑰。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 Signature 第 4 版的 AWS Signature](#)。

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, datestamp, 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，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
```

使用請求 URL 與您的 WebSocket 程式庫，以便向 Amazon Transcribe 提出請求。

6. 對的請求 Amazon Transcribe 必須包含下列標頭。一般而言，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 升級回應來回應。一般而言，您的 WebSocket 程式庫會管理此回應，並設定通訊端以進行通訊 Amazon Transcribe。

以下是 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 WebSocket 串流請求

建立 WebSocket 連線後，用戶端即可開始傳送一系列音訊影格，每個影格都使用[事件串流編碼](#)。

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
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

如果在處理您的請求時發生例外狀況，會 Amazon Transcribe 回應包含事件串流編碼回應的終端機 WebSocket 框架。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。傳送例外狀況回應後，會 Amazon Transcribe 傳送關閉影格。

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

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)
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**：在與用戶端交握期間 Amazon Transcribe 發生問題。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 升級請求是以不正確的存取金鑰或私密金鑰簽署的。請確定您正確建立存取金鑰，並再次嘗試請求。

Amazon Transcribe 也可以傳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 串流服務的資料影格會以此格式編碼。的回應 Amazon Transcribe 也會使用此編碼。

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

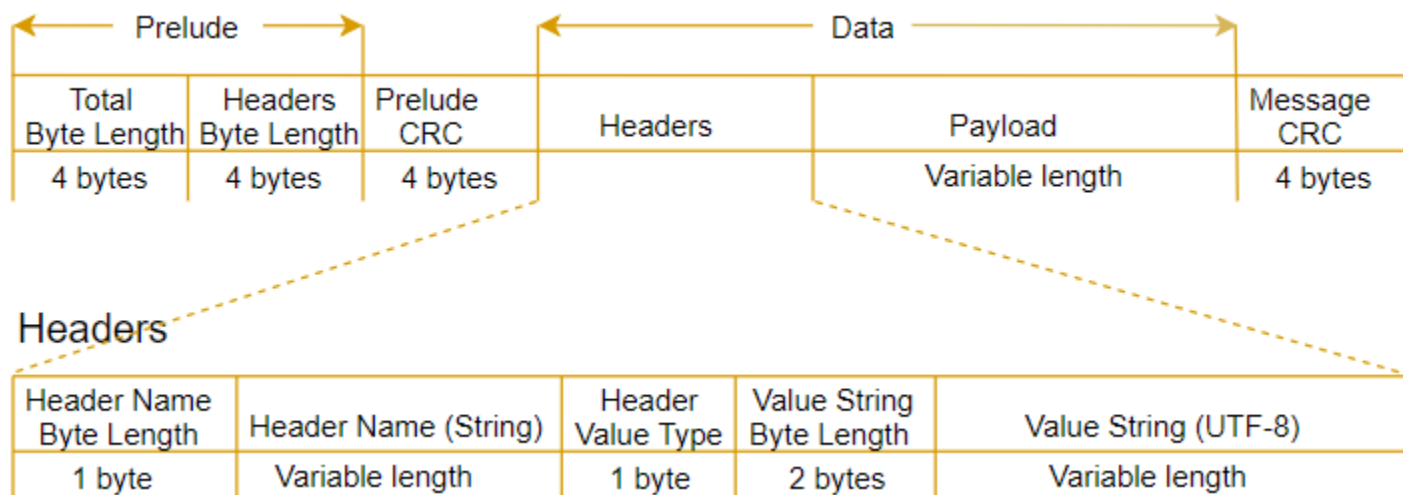
資料區段包含：

1. 標頭
2. 酬載

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。這可確保 Amazon Transcribe 可以立即偵測損毀的位元組長度資訊，而不會造成錯誤，例如緩衝區溢位。
- 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。

標頭框架是串流轉錄的授權框架。Amazon Transcribe 會使用授權標頭的值做為種子，為請求中的資料框架產生授權標頭鍵。

每個標頭都包含下列元件；每個框架有多個標頭。

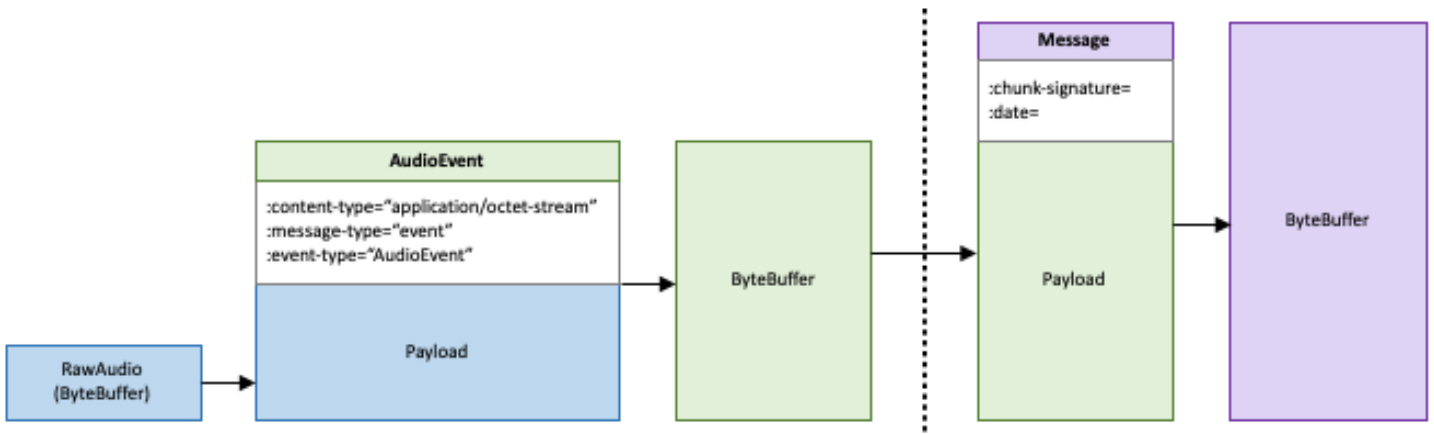
- 標頭名稱位元組長度：標頭名稱的位元組長度。
- 標頭名稱：指出標頭類型的標頭名稱。如需有效值，請參閱下列框架描述。
- 標頭值類型：指出標頭值的數字。下列清單顯示標頭的可能值及其指示內容。
  - 0 – TRUE
  - 1 – FALSE
  - 2 – BYTE
  - 3 – SHORT
  - 4 – INTEGER
  - 5 – LONG
  - 6 – BYTE ARRAY
  - 7 – STRING
  - 8 – TIMESTAMP
  - 9 – UUID
- 數值字串位元組長度：標頭值字串的位元組長度。
- 標頭值：標頭字串的值。此欄位的有效值取決於標頭類型。請參閱 [設定 HTTP/2 串流](#) 或 [設定 WebSocket 串流](#)，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。



## 工作佇列

使用工作佇列時，您可以提交比同時處理更多的轉錄作業。如果沒有工作佇列，達到允許的並行請求的配額時，您必須等到一或多個請求完成後，才能提交新的請求。

對於轉錄任務和通話後分析任務請求，任務佇列是選用的。

如果您啟用任務佇列，會 Amazon Transcribe 建立佇列，其中包含超出限制的所有請求。請求完成後，系統會立即從您的佇列中擷取新的請求並處理。已排入佇列的請求依先進先出 (FIFO) 順序處理。

您最多可新增 10,000 項工作至佇列。如果您超過此限制，您會收到 `LimitExceededConcurrentJobException` 錯誤。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）來處理排入佇列的任務。請注意，這些是提出請求時可以提高的預設值。

### 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。某些預設值可能會在請求時提高。

如果您啟用工作佇列，但未超過並行請求的配額，則會同時處理所有請求。

## 啟用工作佇列

您可以使用 AWS 管理主控台、AWS CLI 或 AWS SDK 啟用工作佇列；請參閱下列範例；請參閱下列範例：

### AWS 管理主控台

1. 登入 [AWS 管理主控台](#)。
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. 在任務設定方塊中，有其他設定面板。如果展開此面板，您可以選擇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。
- 選擇建立作業以執行轉錄作業。

## AWS CLI

此範例使用 [start-transcription-job](#) 指令和 `job-execution-settings` 參數與 `AllowDeferredExecution` 子參數。請注意，您的請求包含 `AllowDeferredExecution` 時，則還必須納入 `DataAccessRoleArn`。

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

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

這是使用 [start-transcription-job](#) 指令的另一個範例，以及啟用佇列的請求主文。

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

檔案 `my-first-queueing-request.json` 包含以下請求主文。

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

## 適用於 Python (Boto3) 的 AWS SDK

此範例使用適用於 Python (Boto3) 的 AWS SDK 來啟用任務佇列，使用 [start\\_transcription\\_job](#) 方法的 `AllowDeferredExecution` 引數。請注意，您的請求包含 `AllowDeferredExecution` 時，則還必須納入 `DataAccessRoleArn`。如需詳細資訊，請參閱 [StartTranscriptionJob](#) 和 [JobExecutionSettings](#)。

如需使用 AWS SDKs 的其他範例，包括功能特定、案例和跨服務範例，請參閱 [Amazon Transcribe AWS SDKs 的程式碼範例](#) 章節。

```
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://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
transcribe.start_transcription_job(
    TranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'amzn-s3-demo-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 管理主控台 或提交 [GetTranscriptionJob](#) 請求來檢視已排入佇列任務的進度。工作已排入佇列時，Status 就是 QUEUED。您的工作開始處理時，狀態會變更為 IN\_PROGRESS，然後在處理完成時變更為 COMPLETED 或 FAILED。

## 標記 資源

標記是一個自訂中繼資料標記，您可將其新增至資源，以輕鬆識別、組織和搜尋資源。標記由兩個單獨的部分組成：標記金鑰和標記值。這被稱為一個 key:value 對。

標記金鑰通常表示較大的分類，而標記值則表示該分類的子集。例如，您可以有標記金鑰 = 顏色和標記值 = 藍色，這將產生 key:value 對 Color:Blue。您可以將標記的值設為空字串，但您無法將標記的值設為 Null。忽略標記值基本上等同於使用空字串。

### Tip

AWS 帳單與成本管理 可以使用標記將您的帳單分隔為動態類別。例如，如果您新增標記以代表公司內的不同部門，例如 Department:Sales 或 Department:Legal、AWS 可以提供每個部門的成本分配。

在 中 Amazon Transcribe，您可以標記下列資源：

- 轉錄作業
- 醫學轉錄作業
- 通話分析通話後轉錄任務
- 自訂詞彙
- 醫療自訂詞彙
- 自訂詞彙篩選
- 呼叫分析類別
- 自訂語言模型

標記索引鍵的長度上限為 128 個字元，標記值的長度上限為 256 個字元；兩者都區分大小寫。每個資源最多 Amazon Transcribe 支援 50 個標記。對於特定資源，每個標記金鑰必須是唯一的，且只有一個值。請注意，您的標記不能以 開頭，aws: 因為 AWS 會保留此字首給系統產生的標記。您無法新增、修改或刪除 aws:\* 標記，而且它們不計入每個資源的標記數限制。

### 特定資源標記的 API 操作

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

若要使用標籤 API，您必須在請求中包含 Amazon Resource Name (ARN)。ARN 具有格式 `arn:partition:service:region:account-id:resource-type/resource-id`。例如，與轉錄作業相關的 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，包括最佳實務，請參閱[標記 AWS 資源](#)。

## 標籤式存取控制

您可以使用標籤來控制中的存取 AWS 帳戶。對於標籤型存取控制，您可以在 IAM 政策的條件元素中提供標籤資訊。然後您可以使用標籤及其關聯的標籤條件金鑰，以控制下列項目的存取權限：

- 資源：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` 條件金鑰，以指定必須連接到該資源的 `key:value` 對。
- 請求：控制您可以在請求傳遞的標籤。
  - 使用 `aws:RequestTag/key-name` 條件索引鍵指定可從 IAM 使用者或角色新增、修改或移除哪些標籤。
- 授權流程：控制授權流程的任何部分，控制標籤型的存取。
  - 使用 `aws:TagKeys/` 條件金鑰，以控制是否可在資源或在請求中由主體使用特定標籤金鑰。在此情況下，此鍵值無關緊要。

如需標籤型存取控制政策的範例，請參閱[根據標籤檢視轉錄作業](#)。

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](#)。

## 將標籤新增至 Amazon Transcribe 資源

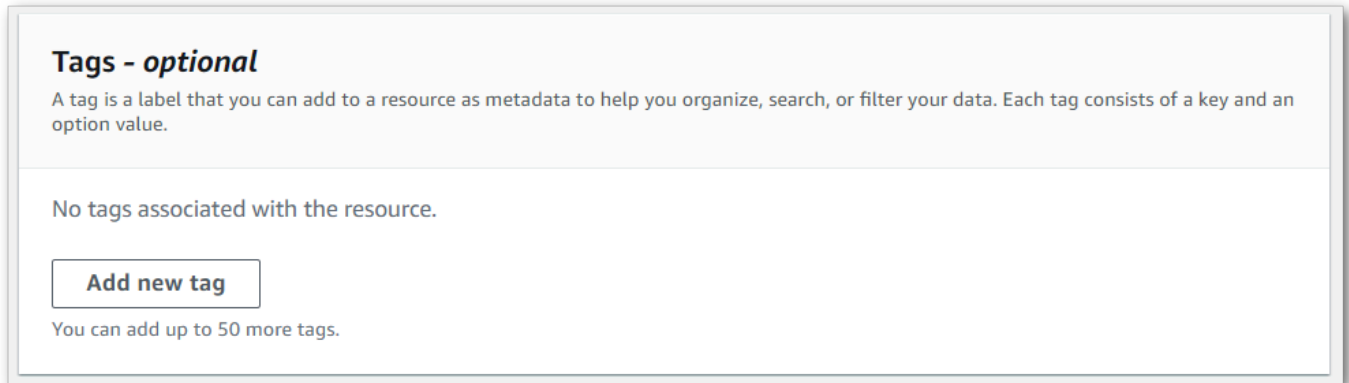
您可以在執行 Amazon Transcribe 任務之前或之後新增標籤。使用現有的 `Create*` 和 `Start*` API，您可以在轉錄要求中新增標籤。

您可以使用 AWS 管理主控台、或 AWS CLI 或 AWS SDK，新增、修改或刪除標籤；請參閱下列項目，取得範例：

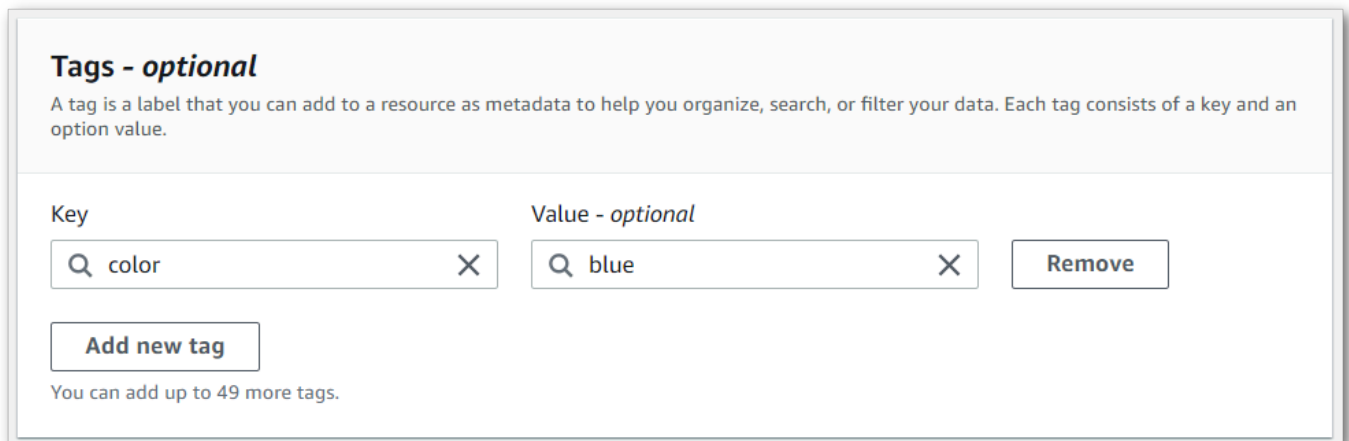
### AWS 管理主控台

1. 登入 [AWS 管理主控台](#)。

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. 捲動至指定作業詳細資訊頁面底部，找到標籤 - 選擇性方塊，然後選擇新增標籤。



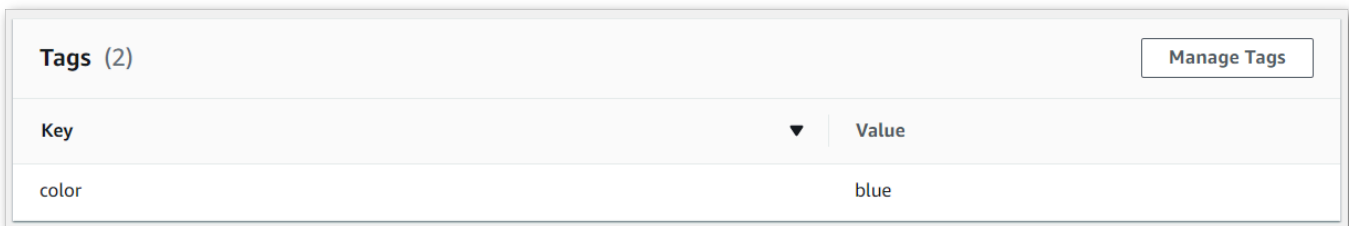
4. 輸入金鑰欄位的資訊，並選擇性輸入值欄位。



5. 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。

選擇建立作業以執行轉錄作業。

6. 您可以導覽至轉錄作業頁面、選擇轉錄作業並捲動至工作資訊頁面的底部，檢視與轉錄作業相關的標籤。如果您想要編輯標籤，可以透過選擇管理標籤以編輯標籤。



## AWS CLI

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

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

這是使用 [start-transcription-job](#) 指令的另一個範例，以及新增標籤到工作的請求內文。

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

檔案 `my-first-tagging-job.json` 包含以下請求內文。

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

## 適用於 Python (Boto3) 的 AWS SDK

下列範例使用 適用於 Python (Boto3) 的 AWS SDK 來新增標籤，方法是使用 [start\\_transcription\\_job](#) 方法的 Tags 引數。如需詳細資訊，請參閱 [StartTranscriptionJob](#) 和 [Tag](#)。

如需使用 AWS SDKs 的其他範例，包括功能特定、案例和跨服務範例，請參閱 [Amazon Transcribe AWS SDKs 的程式碼範例](#) 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://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
transcribe.start_transcription_job(
    TranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'amzn-s3-demo-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)

透過擴音，您可以區分轉錄輸出中的不同發言者。Amazon Transcribe 可以區分最多 30 個唯一發言者，並使用唯一值 (spk\_0 到 spk\_9) 標記每個唯一發言者的文字。

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 管理主控台

1. 登入 [AWS 管理主控台](#)。
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。

若要啟用發言者分割，請在音訊設定中選擇音訊識別。然後選擇發言者分割，並指定發言者的數量。

### 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://amzn-s3-demo-bucket/my-input-files/my-media-file.flac \
--output-bucket-name amzn-s3-demo-bucket \
--output-key my-output-files/ \
--language-code en-US \
--settings ShowSpeakerLabels=true,MaxSpeakerLabels=3
```

以下是使用 [start-transcription-job](#) 的另一個範例，以及啟用該工作的發言者分隔的請求主文。

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

檔案 `my-first-transcription-job.json` 包含以下請求主文。

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

## 適用於 Python (Boto3) 的 AWS SDK

此範例使用適用於 Python (Boto3) 的 AWS SDK 來識別使用 [start\\_transcription\\_job](#) 方法的頻道。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://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
transcribe.start_transcription_job(
    TranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'amzn-s3-demo-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 管理主控台](#)。
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](#)

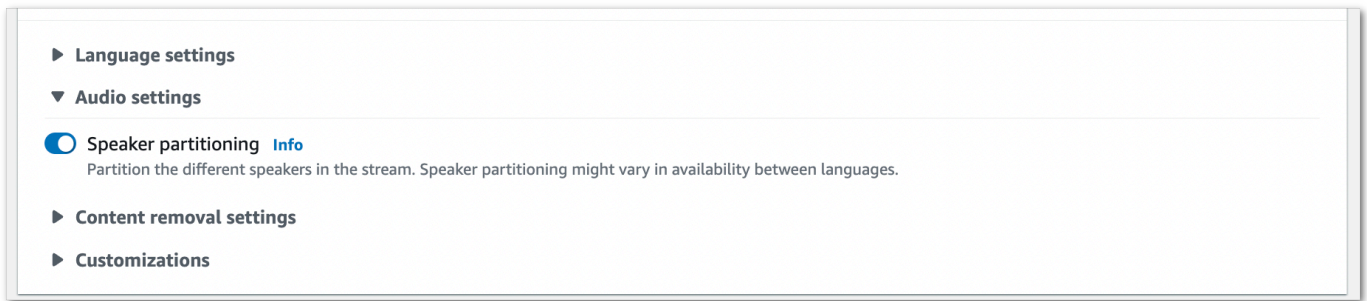
Transcription output Current language: English, US

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 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-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 AWSAPI 操作通用的參數都列在 [通用參數](#) 區段中。

## WebSocket 串流

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 WebSocket 串流](#)。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 AWSAPI 操作通用的參數會列在 [通用參數](#) 區段中。

## 例如日記輸出 (批次)

以下是啟用日記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."
      }
    ],
    "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": [
    {
        "id": 0,
        "start_time": "4.87",
        "speaker_label": "spk_0",
        "end_time": "5.02",
        "alternatives": [
            {
                "confidence": "1.0",
                "content": "I've"
            }
        ],
        "type": "pronunciation"
    },
    {
        "id": 1,
        "start_time": "5.02",
        "speaker_label": "spk_0",
        "end_time": "5.17",
        "alternatives": [
            {
                "confidence": "1.0",
                "content": "been"
            }
        ],
        "type": "pronunciation"
    },
    {
        "id": 2,
        "start_time": "5.17",
        "speaker_label": "spk_0",
```

```
        "end_time": "5.29",
        "alternatives": [
            {
                "confidence": "1.0",
                "content": "on"
            }
        ],
        "type": "pronunciation"
    },
    {
        "id": 3,
        "start_time": "5.29",
        "speaker_label": "spk_0",
        "end_time": "5.64",
        "alternatives": [
            {
                "confidence": "1.0",
                "content": "hold"
            }
        ],
        "type": "pronunciation"
    },
    {
        "id": 4,
        "start_time": "5.64",
        "speaker_label": "spk_0",
        "end_time": "5.84",
        "alternatives": [
            {
                "confidence": "1.0",
                "content": "for"
            }
        ],
        "type": "pronunciation"
    },
    {
        "id": 5,
        "start_time": "6.11",
        "speaker_label": "spk_0",
        "end_time": "6.26",
        "alternatives": [
            {
                "confidence": "1.0",
                "content": "an"
```

```
    }
  ],
  "type": "pronunciation"
},
{
  "id": 6,
  "start_time": "6.26",
  "speaker_label": "spk_0",
  "end_time": "6.88",
  "alternatives": [
    {
      "confidence": "1.0",
      "content": "hour"
    }
  ],
  "type": "pronunciation"
},
{
  "id": 7,
  "speaker_label": "spk_0",
  "alternatives": [
    {
      "confidence": "0.0",
      "content": "."
    }
  ],
  "type": "punctuation"
},
{
  "id": 8,
  "start_time": "8.49",
  "speaker_label": "spk_1",
  "end_time": "8.88",
  "alternatives": [
    {
      "confidence": "1.0",
      "content": "Sorry"
    }
  ],
  "type": "pronunciation"
},
{
  "id": 9,
  "start_time": "8.88",
```

```
    "speaker_label": "spk_1",
    "end_time": "9.05",
    "alternatives": [
      {
        "confidence": "0.902",
        "content": "about"
      }
    ],
    "type": "pronunciation"
  },
  {
    "id": 10,
    "start_time": "9.05",
    "speaker_label": "spk_1",
    "end_time": "9.24",
    "alternatives": [
      {
        "confidence": "1.0",
        "content": "that"
      }
    ],
    "type": "pronunciation"
  },
  {
    "id": 11,
    "speaker_label": "spk_1",
    "alternatives": [
      {
        "confidence": "0.0",
        "content": "."
      }
    ],
    "type": "punctuation"
  }
],
"audio_segments": [
  {
    "id": 0,
    "transcript": "I've been on hold for an hour.",
    "start_time": "4.87",
    "end_time": "6.88",
    "speaker_label": "spk_0",
    "items": [
      0,
    ]
  }
]
```

```
        1,  
        2,  
        3,  
        4,  
        5,  
        6,  
        7  
    ]  
  },  
  {  
    "id": 1,  
    "transcript": "Sorry about that.",  
    "start_time": "8.49",  
    "end_time": "9.24",  
    "speaker_label": "spk_1",  
    "items": [  
      8,  
      9,  
      10,  
      11  
    ]  
  }  
]  
},  
"status": "COMPLETED"  
}
```

## 轉錄多聲道音訊

如果您的音訊有兩個聲道，您可以使用聲道識別來分別轉錄每個聲道的語音。目前Amazon Transcribe不支援具有兩個以上聲道的音訊。

在您的文字記錄中，聲道會指定標籤 `ch_0` 和 `ch_1`。

除了[標準文字記錄區段](#) (transcripts 和 items) 之外，啟用聲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，並依聲道分組，以及其相關的聲道標籤、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，如果某個聲道上的使用者與不同聲道上的使用者同時說話，則每個聲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 [範例聲道識別輸出 \(批次處理\)](#)。

## 批次轉錄中使用聲道識別

若要識別批次轉錄中的聲道，您可以使用 AWS 管理主控台、AWS CLI 或 AWS SDK；請參閱下列範例：

### AWS 管理主控台

1. 登入 [AWS 管理主控台](#)。
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**.

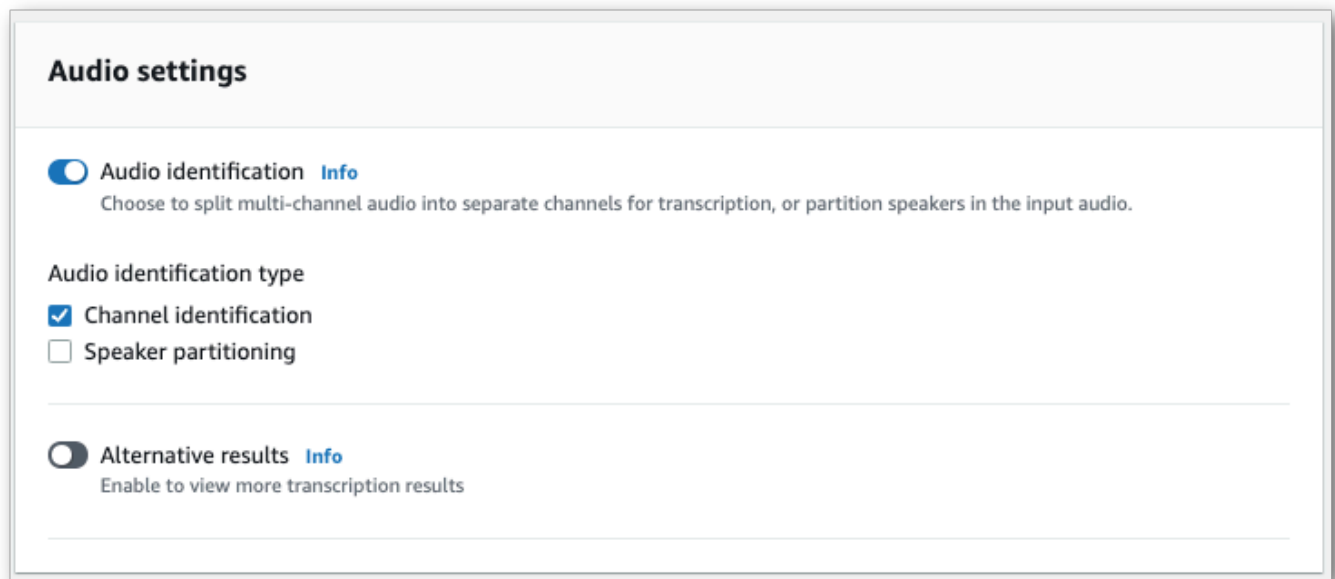
**Language**

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 聲道識別(在「音訊識別類型」標題下)。



**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

**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://amzn-s3-demo-bucket/my-input-files/my-media-file.flac \  
--output-bucket-name amzn-s3-demo-bucket \  
--output-key my-output-files/ \  
--language-code en-US \  
--settings ChannelIdentification=true
```

以下是使用 [start-transcription-job](#) 的另一個範例，以及啟用該工作的聲道識別的請求主文。

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

檔案 *my-first-transcription-job.json* 包含以下請求主文。

```
{
```

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

## 適用於 Python (Boto3) 的 AWS SDK

此範例使用 適用於 Python (Boto3) 的 AWS SDK 來識別使用 [start\\_transcription\\_job](#) 方法的頻道。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://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
transcribe.start_transcription_job(
    TranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'amzn-s3-demo-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)
```

## 在串流轉錄中使用聲道識別

若要識別串流轉錄中的聲道，您可以使用 HTTP/2 或 WebSockets；請參閱下列範例：

### HTTP/2 串流

此範例會建立 HTTP/2 要求，以分隔轉錄輸出中的聲道。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 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-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 AWSAPI 操作常見的參數都列在 [通用參數](#) 區段中。

### WebSocket 串流

此範例會建立預先簽署的 URL，以分隔轉錄輸出中的聲道。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 WebSocket 串流](#)。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=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 AWSAPI 操作常見的參數都列在 [通用參數](#) 區段中。

## 範例聲道識別輸出 (批次處理)

這是啟用聲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": [
  {
    "id": 0,
    "channel_label": "ch_0",
    "start_time": "4.86",
    "end_time": "5.01",
    "alternatives": [
      {
        "confidence": "1.0",
        "content": "I've"
      }
    ],
    "type": "pronunciation"
  },
  {
    "id": 1,
    "channel_label": "ch_0",
    "start_time": "5.01",
    "end_time": "5.16",
    "alternatives": [
      {
        "confidence": "1.0",
        "content": "been"
      }
    ],
    "type": "pronunciation"
  },
  {
    "id": 2,
    "channel_label": "ch_0",
    "start_time": "5.16",
    "end_time": "5.28",
    "alternatives": [
      {
        "confidence": "1.0",
        "content": "on"
      }
    ],
    "type": "pronunciation"
  },
],
```

```
{
  "id": 3,
  "channel_label": "ch_0",
  "start_time": "5.28",
  "end_time": "5.62",
  "alternatives": [
    {
      "confidence": "1.0",
      "content": "hold"
    }
  ],
  "type": "pronunciation"
},
{
  "id": 4,
  "channel_label": "ch_0",
  "start_time": "5.62",
  "end_time": "5.83",
  "alternatives": [
    {
      "confidence": "1.0",
      "content": "for"
    }
  ],
  "type": "pronunciation"
},
{
  "id": 5,
  "channel_label": "ch_0",
  "start_time": "6.1",
  "end_time": "6.25",
  "alternatives": [
    {
      "confidence": "1.0",
      "content": "an"
    }
  ],
  "type": "pronunciation"
},
{
  "id": 6,
  "channel_label": "ch_0",
  "start_time": "6.25",
  "end_time": "6.87",
```

```
        "alternatives": [
          {
            "confidence": "1.0",
            "content": "hour"
          }
        ],
        "type": "pronunciation"
      },
      {
        "id": 7,
        "channel_label": "ch_0",
        "alternatives": [
          {
            "confidence": "0.0",
            "content": "."
          }
        ],
        "type": "punctuation"
      },
      {
        "id": 8,
        "channel_label": "ch_1",
        "start_time": "8.5",
        "end_time": "8.89",
        "alternatives": [
          {
            "confidence": "1.0",
            "content": "Sorry"
          }
        ],
        "type": "pronunciation"
      },
      {
        "id": 9,
        "channel_label": "ch_1",
        "start_time": "8.89",
        "end_time": "9.06",
        "alternatives": [
          {
            "confidence": "0.9176",
            "content": "about"
          }
        ],
        "type": "pronunciation"
      }
    ]
  }
```

```
    },
    {
      "id": 10,
      "channel_label": "ch_1",
      "start_time": "9.06",
      "end_time": "9.25",
      "alternatives": [
        {
          "confidence": "1.0",
          "content": "that"
        }
      ],
      "type": "pronunciation"
    },
    {
      "id": 11,
      "channel_label": "ch_1",
      "alternatives": [
        {
          "confidence": "0.0",
          "content": "."
        }
      ],
      "type": "punctuation"
    }
  ],
  "audio_segments": [
    {
      "id": 0,
      "transcript": "I've been on hold for an hour.",
      "start_time": "4.86",
      "end_time": "6.87",
      "channel_label": "ch_0",
      "items": [
        0,
        1,
        2,
        3,
        4,
        5,
        6,
        7
      ]
    }
  ],
}
```

```
    {
      "id": 1,
      "transcript": "Sorry about that.",
      "start_time": "8.5",
      "end_time": "9.25",
      "channel_label": "ch_1",
      "items": [
        8,
        9,
        10,
        11
      ]
    }
  ],
  "status": "COMPLETED"
}
```

## 識別媒體中的優勢語言

Amazon Transcribe 能夠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，串流前幾秒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來翻譯文字記錄 Amazon Translate，例如。

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 AWS 管理主控台 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

目前支援多語言識別：en-AB、en-AU、en-GB、en-IE、en-IN、en-NZ、en-US、en-WL、en-ZA、es-ES、es-US、fr-CA、fr-FR、zh-CN、zh-TW、pt-BR、pt-PT、de-CH、de-DE、af-ZA、ar-AE、da-DK、he-IL、hi-IN、id-ID、fa-IR、it-IT、ja-JP、ko-KR、ms-MY、nl-NL、ru-RUta-IN、TH-IN、TH-IN、TH-IN

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) 只會 Amazon Transcribe 使用您在將音訊與正確語言配對時指定的語言，這可以加快語言識別速度並改善與指派正確語言方言相關聯的準確性。

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

如果您在請求中包含語言代碼，而且您提供的語言代碼不符合音訊中識別的語言，則會從您指定的語言代碼 Amazon Transcribe 中選取最接近的語言代碼。然後，它會產生該語言的文字記錄。例如，如果您的媒體是美式英文 (en-US) fr-FR，而且您提供 Amazon Transcribe 語言代碼 zh-CN、和 de-DE，則 Amazon Transcribe 可能會將媒體與德文 (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。

## 將語言識別與其他 Amazon Transcribe 功能結合

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，如果使用語言識別搭配內容修訂，您僅限於美式英文 (en-US) 或美式西班牙文 (es-US)，因為這僅適用於修訂的語言。請參閱 [支援的語言和特定語言功能](#)，了解詳細資訊。

### ⚠ Important

如果您在啟用內容修訂的情況下使用自動語言識別，且音訊包含美式英文 (en-US) 或美式西班牙文 (es-US) 以外的語言，則您的文字記錄中只會修訂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 en-US 自訂詞彙，但 Amazon Transcribe 判斷媒體中說的語言為 en-AU，則您的自訂詞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 管理主控台、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 管理主控台

1. 登入 [AWS 管理主控台](#)。
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 [start-transcription-job](#) 指令和 `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://amzn-s3-demo-bucket/my-input-files/my-media-file.flac \
--output-bucket-name amzn-s3-demo-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://amzn-s3-demo-bucket/my-input-files/my-media-file.flac \
--output-bucket-name amzn-s3-demo-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
```

這是使用 [start-transcription-job](#) 指令的另一個範例，以及識別語言的請求內文。

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

檔案 `my-first-language-id-job.json` 包含以下請求內文。

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://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
  },
  "OutputBucketName": "amzn-s3-demo-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://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
  },
  "OutputBucketName": "amzn-s3-demo-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"
    }
  }
}

```

適用於 Python (Boto3) 的 AWS SDK

此範例使用 `start_transcription_job` 方法的 `IdentifyLanguage` 引數適用於 Python (Boto3) 的 AWS SDK，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 SDKs 的其他範例，包括功能特定、案例和跨服務範例，請參閱 [Amazon Transcribe AWS SDKs 的程式碼範例](#) 章節。

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://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
transcribe.start_transcription_job(
    TranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'amzn-s3-demo-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://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
transcribe.start_transcription_job(
    TranscriptionJobName = job_name,
    Media = {
```

```
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'amzn-s3-demo-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說的主要語言。Amazon Transcribe 需要至少一秒鐘的語音來識別語言。

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來翻譯文字記錄 Amazon Transcribe，例如。

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音訊中識別的語言，會從您指定的語言代碼 Amazon Transcribe 中選取最接近的語言。然後，它會產生該語言的文字記錄。例如，如果您的媒體是美式英文 (en-US) fr-FR，並且您提供 Amazon Transcribe 語言代碼 zh-CN、和 de-DE，Amazon Transcribe 可能會將媒體與德文 (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頻道，Amazon Transcribe 可以識別每個頻道中說的主要語言。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。對於多語言識別，僅支援 PCM。

## 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

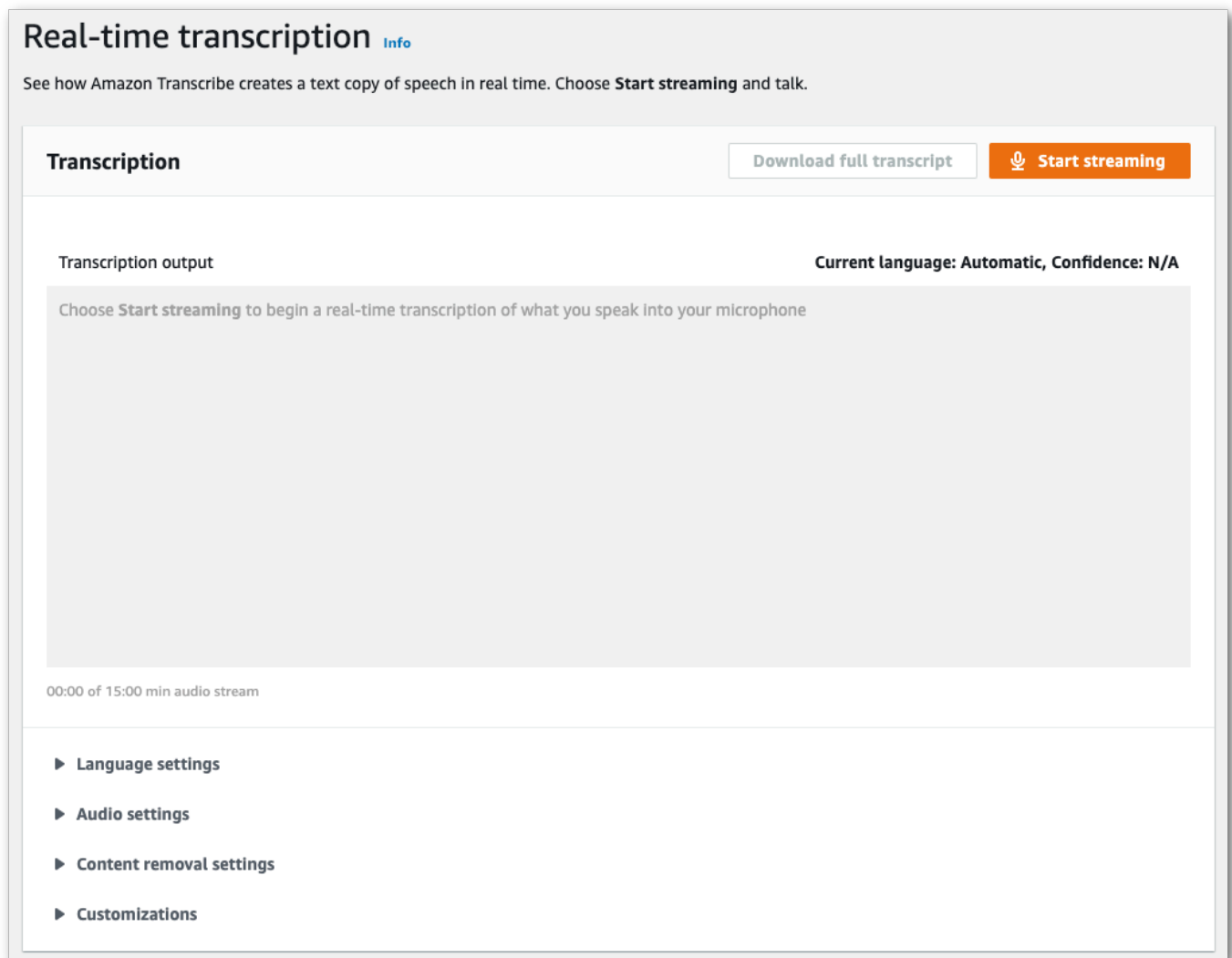
多語言識別目前不支援修訂和自訂語言模型。

## 透過串流媒體使用語言識別

您可以使用 AWS 管理主控台、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 管理主控台

1. 登入 [AWS 管理主控台](#)。
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。



3. 選取自動語言識別或自動多語言識別。

### ▼ 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 en-GB 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-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-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](#) 中找到參數定義；所有 AWS API 操作常見的參數都列在 [通用參數](#) 區段中。

## 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。如需搭配使用 WebSocket 串流的詳細資訊 Amazon Transcribe，請參閱 [設定 WebSocket 串流](#)。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。如需搭配使用 WebSocket 串流的詳細資訊 Amazon Transcribe，請參閱 [設定 WebSocket 串流](#)。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](#) 中找到參數定義；所有 AWS API 操作常見的參數都列在 [通用參數](#) 區段中。

## 替代轉錄

Amazon Transcribe 轉錄音訊時，它會建立相同文字記錄的不同版本，並為每個版本指派可信度分數。在一般轉錄中，您僅會獲得可信度分數最高的版本。

如果您開啟替代文字記錄，會 Amazon Transcribe 傳回可信度較低的其他版本文字記錄。您最多可以選擇傳回 10 個替代轉錄。如果您指定的替代項目數量大於 Amazon Transcribe 識別項目的數量，則只會傳回實際的替代項目數量。

所有替代選項都位於相同的轉錄輸出檔案中，並在區段層級顯示。區段會依語音中的自然停頓而定，例如，講者改變或音訊停頓。

替代轉錄僅適用於批次轉錄。

轉錄輸出的結構如下所示。程式碼範例中的省略號 (.....) 是為簡潔起見而移除內容的位置。

### 1. 指定區段的完整最終轉錄。

```
"results": {
  "language_code": "en-US",
  "transcripts": [
    {
      "transcript": "The amazon is the largest rainforest on the planet."
    }
  ],
  .....
```

### 2. 前一 transcript 部分中每個單字的可信度分數。

```
"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。每個區段的替代選項是依可信度分數遞減排序。

```

"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. 轉錄輸出尾聲處的狀態。

```
"status": "COMPLETED"  
}
```

## 請求替代轉錄

您可以使用 AWS 管理主控台、AWS CLI 或 AWS SDK，請求替代轉錄，請參閱下列範例：

### AWS 管理主控台

1. 登入 [AWS 管理主控台](#)。
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.

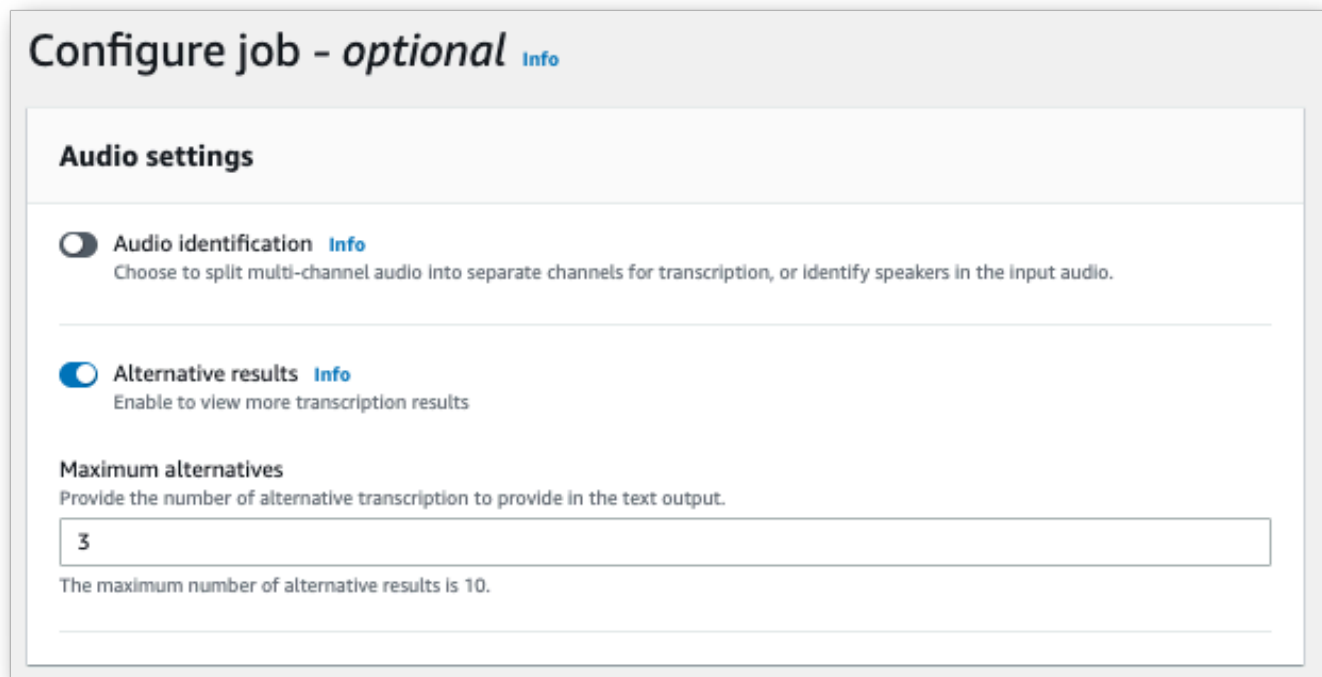
**Automatic language identification [Info](#)**  
If you don't know the language spoken in your audio files, choose this option. You have access to fewer options for additional processing than if you choose **Specific language**.

**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。

選擇替代結果，然後指定您想要在文字記錄中輸入的替代轉錄結果數量上限。



4. 選擇建立作業以執行轉錄作業。

## AWS CLI

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

請注意，如果您的請求包含 `ShowAlternatives=true`，則還必須納入 `MaxAlternatives`。

```
aws transcribe start-transcription-job \
  --region us-west-2 \
  --transcription-job-name my-first-transcription-job \
  --media MediaFileUri=s3://amzn-s3-demo-bucket/my-input-files/my-media-file.flac \
  --output-bucket-name amzn-s3-demo-bucket \
  --output-key my-output-files/ \
  --language-code en-US \
  --settings ShowAlternatives=true,MaxAlternatives=4
```

以下是使用 [start-transcription-job](#) 指令的另一個範例，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
```

檔案 `my-first-alt-transcription-job.json` 包含以下請求主體。

```
{
  "TranscriptionJobName": "my-first-transcription-job",
  "Media": {
    "MediaFileUri": "s3://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
  },
  "OutputBucketName": "amzn-s3-demo-bucket",
  "OutputKey": "my-output-files/",
  "LanguageCode": "en-US",
  "Settings": {
    "ShowAlternatives": true,
    "MaxAlternatives": 4
  }
}
```

## 適用於 Python (Boto3) 的 AWS SDK

下列範例使用適用於 Python (Boto3) 的 AWS SDK 來請求替代轉錄，方法是使用 [start\\_transcription\\_job](#) 方法的 `ShowAlternatives` 引數。如需詳細資訊，請參閱 [StartTranscriptionJob](#) 及 [ShowAlternatives](#)。

如需使用 AWSSDKs 的其他範例，包括功能特定、案例和跨服務範例，請參閱 [Amazon Transcribe AWS SDKs 的程式碼範例](#) 章節。

請注意，如果您的請求包含 `'ShowAlternatives': True`，則還必須納入 `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://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
transcribe.start_transcription_job(
    TranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'amzn-s3-demo-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)
```

## 透過自訂詞彙和自訂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術語，例如品牌名稱、縮寫、技術單字和術語，Amazon Transcribe 則可能無法在您的轉錄輸出中正確擷取這些術語。

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](#) 目的是調整和提高所有情況下特定單字的識別和格式。這涉及到提供 Amazon Transcribe 單字和發音和顯示格式。

如果 Amazon Transcribe 未正確轉譯文字記錄中的特定詞彙，您可以建立自訂詞彙檔案，告知您希望這些詞彙 Amazon Transcribe 如何顯示。這種特定單字的方法最適合用於更正品牌名稱和縮詞之類的術語。

- [自訂語言模型](#) 目的是擷取與術語相關的上下文。這包括 Amazon Transcribe 提供大量的網域特定文字資料。

如果 Amazon Transcribe 未正確轉譯技術術語或在文字記錄中使用不正確的同音電話，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，自訂語言模型可以了解使用「浮冰」(浮冰) 與「流」(線性流) 的時機。

這種情境感知方法最適合轉錄大量領域特定語音。自訂語言模型可以產生在自訂詞彙上顯著提高準確性。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詞彙。

### Tip

若要達到最高的轉錄準確性，請將自訂詞彙與您的自訂語言模型搭配使用。

如需如何使用 建立自訂詞彙的影片示範 AWS 管理主控台，請參閱[使用自訂詞彙](#)。

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 \(CLM\) 提升轉錄準確性](#)。

### 使用 AWS Machine Learning 部落格深入探索

自訂詞彙：

- [使用 即時轉錄 F1 競賽 Amazon Transcribe](#)

自訂語言模型：

- [建置自訂語言模型，大幅提升speech-to-text效能 Amazon Transcribe](#)
- [使用 Amazon Transcribe 自訂語言模型，提高課堂講解的轉錄準確性](#)

## 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 個自訂詞彙檔案 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 管理主控台，您的詞彙檔案可以是文字 (\*.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
- 若要使用自訂詞彙，它必須在與您的轉錄 AWS 區域 相同的 中建立。

### Tip

您可以使用 測試自訂詞彙 AWS 管理主控台。一旦自訂詞彙可供使用，請登入 AWS 管理主控台，選取即時轉錄、捲動至自訂、開啟自訂詞彙，然後從下拉式清單中選取自訂詞彙。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 管理主控台 AWS CLI、和 AWS SDKs 都以相同方式使用自訂詞彙表；每個方法的清單使用方式各不相同，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](#) 和 [使用清單建立自訂詞彙](#)。

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 不再支援自訂詞彙。請將資料欄保留空白。將忽略此欄中的任何值。我們將在未來移除此欄的支援。	IPA 不再支援自訂詞彙。請將資料欄保留空白。將忽略此欄中的任何值。我們將在未來移除此欄的支援。	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>如果此欄中的資料列為空，Amazon Transcribe 會使用資料 Phrase 欄的內容來判斷輸出。</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：

- 您的資料表必須包含所有四個資料欄標頭 (Phrase, SoundsLike, IPA, and DisplayAs)。資料Phrase欄必須在每個資料列上包含項目。透過 IPA和 SoundsLike 提供發音輸入的能力不再受支援，您可以讓資料欄保持空白。這些欄中的任何值都會遭到忽略。
- 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 自訂詞彙SoundsLike不再支援 和 。請將資料欄保留空白。系統會忽略這些欄中的任何值。我們將在未來移除此欄的支援。
- 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。
- 您必須先將自訂詞彙檔案上傳至 Amazon S3 儲存貯體，並使用 `aws s3 cp` 處理，[CreateVocabulary](#)才能將其包含在轉錄請求中。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	[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

## 建立自訂詞彙表

若要處理自訂詞彙資料表以搭配使用 Amazon Transcribe，請參閱下列範例：

### AWS 管理主控台

1. 登入 [AWS 管理主控台](#)。
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 Amazon S3 位置匯入 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。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	SoundsLike (optional)	IPA (optional)	DisplayAs (optional)
No rows added yet			

[Add 10 rows](#)

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 ▼

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 ▼

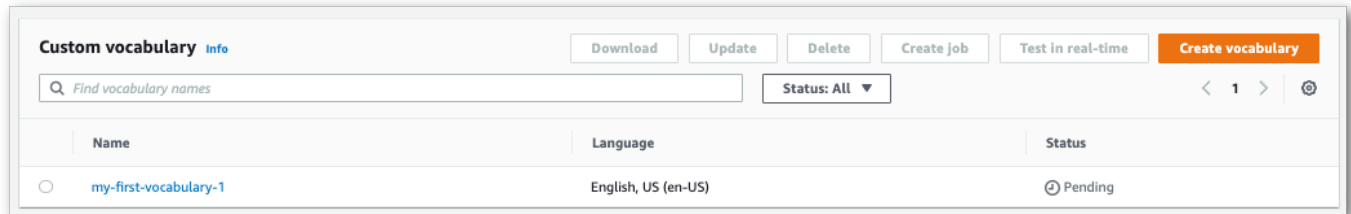
Filter Phrase, SoundsLike, IPA or DisplayAs Show all ▼ < 1 >

<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>	-	-	Amazon EC2
	<span>⚠</span> Phrase contains unsupported characters (" "). Phrase contains a formatting error.			
<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

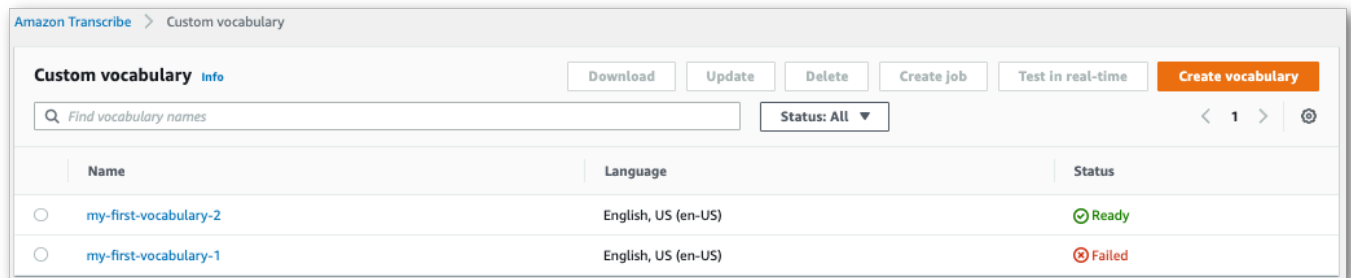
Add row

選擇核取記號 (✓) 以儲存變更，或選擇「X」以捨棄變更。

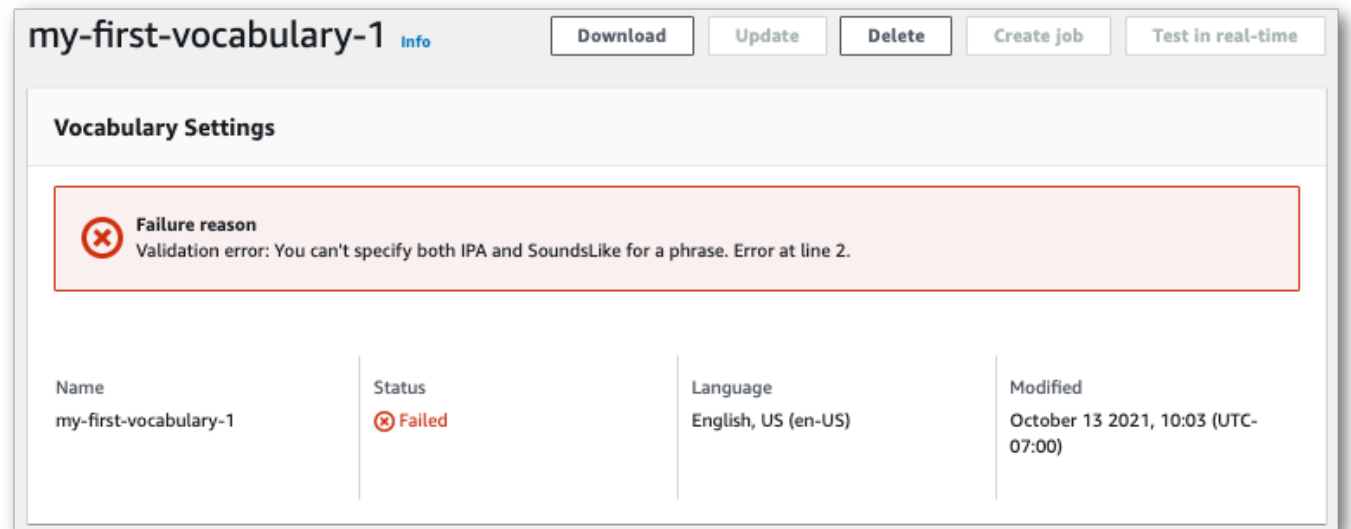
5. 或者，將標籤新增至您的自訂詞彙。完成所有欄位並滿意您的詞彙後，選擇頁面底部的建立詞彙。這會引導您返回自訂詞彙頁面，您可以在此檢視自訂詞彙的狀態。狀態從「待處理」變更為「就緒」時，您的自訂詞彙可以與轉錄搭配使用。



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](#)。

若要在轉錄任務中使用現有的自訂詞彙，請在呼叫 [StartTranscriptionJob](#) 操作時 VocabularyName 在 [Settings](#) 欄位中設定，或從下拉式清單 AWS 管理主控台中選擇自訂詞彙。

```
aws transcribe create-vocabulary \  
--vocabulary-name my-first-vocabulary \  
--vocabulary-file-uri s3://amzn-s3-demo-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
```

檔案 *my-first-vocab-table.json* 包含以下請求內文。

```
{  
  "VocabularyName": "my-first-vocabulary",  
  "VocabularyFileUri": "s3://amzn-s3-demo-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
```

適用於 Python (Boto3) 的 AWS SDK

此範例使用從資料表使用 [create\\_vocabulary](#) 方法適用於 Python (Boto3) 的 AWS SDK 建立自訂詞彙。如需詳細資訊，請參閱 [CreateVocabulary](#)。

若要在轉錄任務中使用現有的自訂詞彙，請在呼叫 [StartTranscriptionJob](#) 操作時 VocabularyName 在 [Settings](#) 欄位中設定，或從下拉式清單 AWS 管理主控台中選擇自訂詞彙。

如需使用 AWS SDKs 的其他範例，包括功能特定、案例和跨服務範例，請參閱 [Amazon Transcribe AWS SDKs 的程式碼範例](#) 章節。

```
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://amzn-s3-demo-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

如果您為自訂詞彙檔案建立新的儲存 Amazon S3 貯體，請確定提出 [CreateVocabulary](#) 請求 IAM 的角色具有存取此儲存貯體的許可。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指定 IAM 角色，方法是包含 `DataAccessRoleArn` 參數。如需中 IAM 角色和政策的詳細資訊 Amazon Transcribe，請參閱 [Amazon Transcribe 身分型政策範例](#)。

## 使用清單建立自訂詞彙

### Important

清單格式的自訂詞彙已被取代，因此，如果您要建立新的自訂詞彙，我們強烈建議您使用 [表格格式](#)。

您可以使用 AWS 管理主控台、AWS CLI、或 AWS SDKs 從清單中建立自訂詞彙。

- **AWS 管理主控台：**您必須建立並上傳包含自訂詞彙的文字檔案。您可以使用以分行符號或逗號分隔的項目。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，您的自訂詞彙將無法被 Amazon Transcribe 接受。

- AWS CLI 和 AWS SDK：您必須使用 [Phrases](#) 標記，在 API 呼叫中以逗號分隔的項目納入自訂詞彙。

如果項目包含多個單字，您必須使用連字符連接每個單字。例如，您將以 **Los-Angeles** 納入「Los Angeles」和以 **Andorra-la-Vella** 納入「Andorra la Vella」。

以下是兩種有效清單格式的範例。請參閱 [建立自訂詞彙清單](#)，取得方法特定範例。

- 以逗號分隔的項目：

```
Los-Angeles,CLI,Eva-Maria,ABCs,Andorra-la-Vella
```

- 分行符號的項目：

```
Los-Angeles
CLI
Eva-Maria
ABCs
Andorra-la-Vella
```

### Important

您僅可使用您的語言支援的字元。請參閱您語言的 [字元集](#)，了解詳細資訊。

[CreateMedicalVocabulary](#) 操作不支援自訂詞彙清單。如果要建立自訂醫療詞彙，您必須使用表格格式；請參閱 [使用資料表建立自訂詞彙](#)，了解說明指示。

## 建立自訂詞彙清單

若要處理要搭配使用的自訂詞彙清單 Amazon Transcribe，請參閱下列範例：

### AWS CLI

此範例使用 [create-vocabulary](#) 指令並搭配清單格式的自訂詞彙檔案。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-list.json
```

檔案 *my-first-vocab-list.json* 包含以下請求主文。

```
{  
  "VocabularyName": "my-first-vocabulary",  
  "LanguageCode": "en-US",  
  "Phrases": [  
    "CLI", "Eva-Maria", "ABCs"  
  ]  
}
```

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
```

適用於 Python (Boto3) 的 AWS SDK

此範例使用從清單使用 [create\\_vocabulary](#) 方法適用於 Python (Boto3) 的 AWS SDK 建立自訂詞彙。如需詳細資訊，請參閱 [CreateVocabulary](#)。

如需使用 AWS SDKs 的其他範例，包括功能特定、案例和跨服務範例，請參閱 [Amazon Transcribe AWS SDKs 的程式碼範例](#) 章節。

```
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

如果您為自訂詞彙檔案建立新的儲存 Amazon S3 貯體，請確定提出 [CreateVocabulary](#) 請求 IAM 的角色具有存取此儲存貯體的許可。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指定 IAM 角色，方法是包含 `DataAccessRoleArn` 參數。如需中 IAM 角色和政策的詳細資訊 Amazon Transcribe，請參閱 [Amazon Transcribe 身分型政策範例](#)。

## 使用自訂的詞彙

建立自訂詞彙後，您可以包含在轉錄請求中；請參閱以下章節以取得範例。

您要包含在請求中的自訂詞彙的語言，必須與您為媒體指定的語言代碼相符。如果語言不相符，您的自訂詞彙就不會套用至您的轉錄，且不會出現警告或錯誤。

### 在批次轉錄中使用自訂詞彙

若要搭配批次轉錄使用自訂詞彙，請參閱下列範例：

#### AWS 管理主控台

1. 登入 [AWS 管理主控台](#)。
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**

為工作命名並指定輸入媒體。選擇性包含任何其他欄位，然後選擇下一步。

3. 在配置工作頁面底部的自訂面板中，開啟自訂詞彙。

## 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. 從下拉式選單中選擇您的自訂詞彙。

選擇建立作業以執行轉錄作業。

## AWS CLI

此範例使用 [start-transcription-job](#) 指令和 Settings 參數與 VocabularyName 子參數。如需詳細資訊，請參閱 [StartTranscriptionJob](#) 和 [Settings](#)。

```
aws transcribe start-transcription-job \  
--region us-west-2 \  
--transcription-job-name my-first-transcription-job \  
--media MediaFileUri=s3://amzn-s3-demo-bucket/my-input-files/my-media-file.flac \  
--output-bucket-name amzn-s3-demo-bucket \  
--output-key my-output-files/ \  
--language-code en-US \  
--settings VocabularyName=my-first-vocabulary
```

這是使用 [start-transcription-job](#) 指令的另一個範例，以及包含工作的自訂詞彙的請求主文。

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

檔案 *my-first-vocabulary-job.json* 包含以下請求主文。

```
{  
  "TranscriptionJobName": "my-first-transcription-job",  
  "Media": {  
    "MediaFileUri": "s3://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"  
  },  
  "OutputBucketName": "amzn-s3-demo-bucket",  
  "OutputKey": "my-output-files/",  
  "LanguageCode": "en-US",  
  "Settings": {  
    "VocabularyName": "my-first-vocabulary"  
  }  
}
```

## 適用於 Python (Boto3) 的 AWS SDK

此範例使用適用於 Python (Boto3) 的 AWS SDK 來包含自訂詞彙，並使用 [start\\_transcription\\_job](#) 方法的引 Settings 數。如需詳細資訊，請參閱 [StartTranscriptionJob](#) 和 [Settings](#)。

如需使用 AWS SDKs 的其他範例，包括功能特定、案例和跨服務範例，請參閱 [Amazon Transcribe AWS SDKs 的程式碼範例](#) 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://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
transcribe.start_transcription_job(
    TranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'amzn-s3-demo-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


若要將自訂詞彙與串流轉錄搭配使用，請參閱下列範例：

### AWS 管理主控台

1. 登入 [AWS 管理主控台](#)。
2. 在導覽窗格中，選擇 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 自訂詞彙，然後從下拉式選單中選擇自訂詞彙。

▼ **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 ▼

加入您要套用到串流的其他設定。

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

此範例會建立包含您自訂詞彙的 HTTP/2 請求。如需搭配 HTTP/2 串流使用的詳細資訊 Amazon Transcribe，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-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 操作常見的參數都列在 [通用參數](#) 區段中。

## WebSocket 串流

此範例會建立個預先簽署的 URL，將您的自訂詞彙套用至 WebSocket 串流。已加入分行符號以提高可讀性。如需搭配 WebSocket 串流使用的詳細資訊 Amazon Transcribe，請參閱 [設定 WebSocket 串流](#)。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
```

```
&vocabulary-name=my-first-vocabulary
```

您可以在 [API 參考](#) 中找到參數定義；所有 AWS API 操作常用的參數都列在 [通用參數](#) 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) 以提高單字的辨識度，自訂語言模型會學習與特定單字相關的內容。這包括單字的使用方式和時間，以及單字與其他詞語的關係。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「浮冰」是比「冰流」更頻繁出現的單字對。

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，如果您在請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)。

### 自訂語言模型特定的 API 操作

[CreateLanguageModel](#), [DeleteLanguageModel](#), [DescribeLanguageModel](#),  
[ListLanguageModels](#)

## 資料來源

您可以使用任何類型的文字資料以訓練模型。但是，您的文字內容越接近音訊內容，您的模型就越準確。因此，選擇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文字記錄。被認為是領域內資料。領域內文字資料與您要轉錄的音訊有完全相同的術語、用法和上下文。

如果您沒有準確的文字記錄，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術語。這被視為與領域相關的資料。

建立健全的自訂語言模型可能需要大量的文字資料，其中必須包含音訊說出的術語。您最多可以提供 Amazon Transcribe 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) 領域內文字記錄時，您可以提供 Amazon Transcribe 最多 200 MB 的文字資料來調整模型，這稱為調校資料。

## 訓練與調整資料

訓練資料的目的是教導 Amazon Transcribe 辨識新詞彙，並了解使用這些詞彙的內容。為了建立強大的模型，Amazon Transcribe 可能需要大量的相關文字資料。強烈建議提供盡可能最多的訓練資料，上限是 2 GB 的資料。

調整資料的目的在於協助精簡和最佳化從訓練資料中學習的情境關係。建立自訂語言模型不需要調整資料。

您可以決定如何最適當選擇訓練，以及選擇性調整資料的方式。每種情況都是獨一無二，且須視您擁有的資料類型和數量。如果缺少領域內訓練資料，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獨一無二的資料。重疊的資料可能會導致您的自訂語言模型偏差和偏離，進而影響其準確性。

一般指導原則是，我們建議盡可能使用準確的領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 個單字的準確領域內文字記錄，請將作為訓練資料使用。在這種情況下，不需要包括調整資料。這是訓練自訂語言模型的理想案例。
- 如果您的領域內部文字記錄不到 10,000 個單字，但未達到理想的結果，請考慮使用與領域相關的書面文字 (例如技術報告) 以增強您的訓練資料。在這種情況下，請保留一小部分 (10-25%) 的領域內文字記錄資料作為調整資料使用。
- 如果您沒有領域內的文字記錄，請將所有與領域相關的文字上傳為訓練資料。在這種情況下，轉錄文字樣式的文字比書面文字更適當。這是訓練自訂語言模型最不具效率的案例。

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

建立自訂語言模型前，您必須：

- 準備您的資料。資料必須以純文字格式儲存，且不能包含任何特殊字元。
- 將資料上傳至 Amazon S3 儲存貯體。建議為訓練和調整資料建立個別的資料夾。
- 請確定 Amazon Transcribe 可存取您的 Amazon S3 儲存貯體。您必須指定具有存取許可 IAM 的角色，才能使用您的資料。

## 準備您的資料

您可以將所有資料編譯在一個檔案中，也可以另存為多個檔案。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。

無論您進行培訓或調整資料所使用的文字檔案數量。上傳一個包含 100,000 個單字的檔案會產生與上傳 10 個檔案 (含 10,000 個單字) 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案都符合以下條件：

- 它們都使用與您要建立的模型相同的語言。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 編碼的純文字格式。
- 它們不包含任何特殊字元或格式，例如 HTML 標記。
- 訓練資料的合併大小上限為 2 GB，調整資料的大小上限為 200 MB。

如果不符合這些條件，您的模型會失敗。

## 上傳資料

上傳資料前，請先為訓練資料建立新資料夾。如果使用調整資料，請建立獨立的資料夾

您的儲存貯體的 URI 可能如下所示：

- `s3://amzn-s3-demo-bucket/my-model-training-data/`
- `s3://amzn-s3-demo-bucket/my-model-tuning-data/`

將您的訓練和調整資料上傳至適當的儲存貯體。

您可以稍後將更多資料新增至這些儲存貯體。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資料更新。

## 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儲存 Amazon S3 貯體許可 IAM 的角色。如果您尚未擁有可存取您放置訓練資料的儲存 Amazon S3 貯體的角色，則必須建立一個。建立角色後，您可以連接政策以授與該角色權限。請勿將政策連接至使用者

如需範例政策，請參閱 [Amazon Transcribe 身分型政策範例](#)。

若要了解如何建立新的 IAM 身分，請參閱 [IAM 身分（使用者、使用者群組和角色）](#)。

若要了解政策的詳細資訊，請參閱：

- [IAM 的政策和許可](#)
- [建立 IAM 政策](#)
- [AWS 資源的存取權管理](#)

## 建立您的自訂語言模型

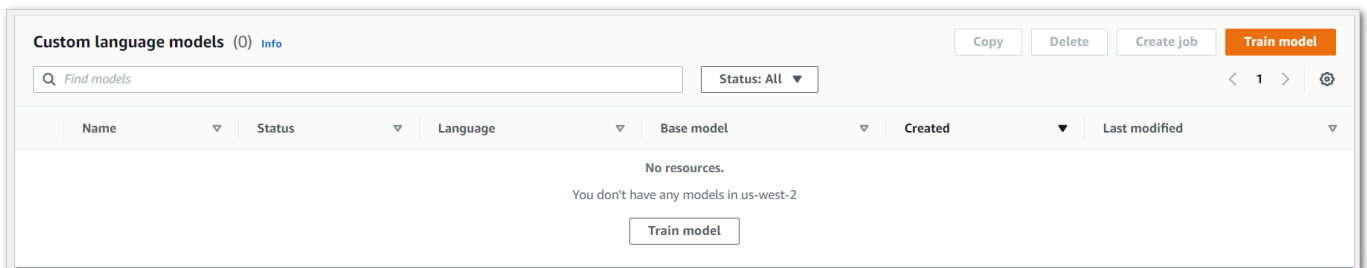
建立自訂語言模型時，您必須選擇基礎模型。有兩種基本模型選項：

- **NarrowBand**：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 Hz 記錄的電話對話。
- **WideBand**：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 管理主控台、AWS CLI、AWS SDKs；請參閱下列範例：

### AWS 管理主控台

1. 登入 [AWS 管理主控台](#)。
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，選擇 訓練模型。



此會引導您前往 訓練模型 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，並選擇性新增調整資料。您必須包含具有存取資料許可 IAM 的角色。

## 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，選擇頁面底部的 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](#) 和 [LanguageModel](#)。

```
aws transcribe create-language-model \  
--base-model-name NarrowBand \  
--model-name my-first-language-model \  
--input-data-config S3Uri=s3://amzn-s3-demo-bucket/my-clm-training-  
data/,TuningDataS3Uri=s3://amzn-s3-demo-bucket/my-clm-tuning-  
data/,DataAccessRoleArn=arn:aws:iam::111122223333:role/ExampleRole \  
--language-code en-US
```

這是使用 [create-language-model](#) 指令的另一個範例，以及建立自訂語言模型的請求主文。

```
aws transcribe create-language-model \  
--cli-input-json file://filepath/my-first-language-model.json
```

檔案 *my-first-language-model.json* 包含以下請求主文。

```
{  
  "BaseModelName": "NarrowBand",  
  "ModelName": "my-first-language-model",  
  "InputDataConfig": {  
    "S3Uri": "s3://amzn-s3-demo-bucket/my-clm-training-data/",  
    "TuningDataS3Uri": "s3://amzn-s3-demo-bucket/my-clm-tuning-data/",  
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/ExampleRole"  
  },  
  "LanguageCode": "en-US"  
}
```

## 適用於 Python (Boto3) 的 AWS SDK

此範例使用 適用於 Python (Boto3) 的 AWS SDK 來使用 [create\\_language\\_model](#) 方法建立 CLM。如需詳細資訊，請參閱 [CreateLanguageModel](#) 和 [LanguageModel](#)。

如需使用 AWS SDKs 的其他範例，包括功能特定、案例和跨服務範例，請參閱 [Amazon Transcribe AWS SDKs的程式碼範例](#) 章節。

```
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://amzn-s3-demo-bucket/my-clm-training-data/',
        'TuningDataS3Uri': 's3://amzn-s3-demo-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 會持續更新適用於自訂語言模型的基本模型。為了從這些更新中受益，我們建議每 6 到 12 個月訓練新的自訂語言模型一次。

若要查看您的自訂語言模型是否使用最新的基礎模型，請使用 AWS CLI 或 AWS SDK 執行 [DescribeLanguageModel](#) 請求，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。如果語言不相符，則您的自訂語言模型不會套用於轉錄，且不會出現警告或錯誤。

### 在批次轉錄中使用自訂語言模型

若要搭配批次轉錄使用自訂語言模型，請參閱下列範例：

## AWS 管理主控台

1. 登入 [AWS 管理主控台](#)。
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. 在任務設定面板的模型類型下，選擇自訂語言模型方塊。

### 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**

您還必須從下拉式選單中選擇輸入語言。

### 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，從下拉式選單選擇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 Amazon S3 的位置。

- 選擇下一步以取得其他組態選項。

選擇建立作業以執行轉錄作業。

## AWS CLI

此範例使用 [start-transcription-job](#) 指令和 [ModelSettings](#) 參數與 [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://amzn-s3-demo-bucket/my-input-files/my-media-file.flac \  
--output-bucket-name amzn-s3-demo-bucket \  
--output-key my-output-files/ \  
--language-code en-US \  
--model-settings LanguageModelName=my-first-language-model
```

這是使用 [start-transcription-job](#) 指令的另一個範例，以及包含該工作的自訂語言模型的請求主文。

```
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` 包含以下請求主文。

```
{  
  "TranscriptionJobName": "my-first-transcription-job",  
  "Media": {  
    "MediaFileUri": "s3://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"  
  },  
  "OutputBucketName": "amzn-s3-demo-bucket",  
  "OutputKey": "my-output-files/",  
  "LanguageCode": "en-US",  
  "ModelSettings": {  
    "LanguageModelName": "my-first-language-model"  
  }  
}
```

適用於 Python (Boto3) 的 AWS SDK

此範例使用適用於 Python (Boto3) 的 AWS SDK 來包含使用 [start\\_transcription\\_job](#) 方法的 `ModelSettings` 引數的自訂語言模型。如需詳細資訊，請參閱 [StartTranscriptionJob](#) 及 [ModelSettings](#)。

如需使用 AWS SDKs 的其他範例，包括功能特定、案例和跨服務範例，請參閱 [Amazon Transcribe AWS SDKs 的程式碼範例](#) 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://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
transcribe.start_transcription_job(
    TranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'amzn-s3-demo-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


若要搭配串流轉錄使用自訂語言模型，請參閱下列範例：

### AWS 管理主控台

1. 登入 [AWS 管理主控台](#)。
2. 在導覽窗格中，選擇 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，然後從下拉式選單中選擇模型。

▼ **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 ▼

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。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

此範例會建立包含您自訂語言模型的 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 [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 操作通用的參數都列在 [通用參數](#) 區段中。

## WebSocket 串流

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 WebSocket 串流](#)。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
```

```
&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 操作通用的參數都列在 [通用參數](#) 區段中。

## 使用自訂詞彙篩選刪除、遮蔽或標記單字

自訂詞彙篩選是文字檔案，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，自訂詞彙篩選是完全自訂功能，因此您可以選擇任何您想要的單字。例如，如果您有即將推出的新產品，您可以在會議記錄中遮蔽產品名稱。在這種情況下，您可以讓利益相關者隨時了解最新狀態，同時維持產品名稱機密，直到發布。

詞彙篩選有三種顯示方法：mask、remove 和 tag。請參閱下列範例，了解每個運作方式。

- 遮蔽：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：刪除指定的單字，在原來位置不留下任何內容。

```
"transcript": "You can specify a list of or words, and removes them from transcripts automatically."
```

- 標籤：為每個指定的單字增加標籤 ("vocabularyFilterMatch": true)，但不會改變單字本身。標記允許快速的文字記錄替換和編輯。

```
"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，您可以指定自訂詞彙篩選條件和您要套用的篩選方法。Amazon Transcribe 然後，當您的轉錄中出現確切的字詞相符項目時，根據您指定的篩選方法進行修改。

自訂詞彙篩選可以套用於批次和串流轉錄請求。若要了解如何建立自訂詞彙篩選，請參閱 [建立詞彙篩選條件](#)。若要了解如何套用自訂詞彙篩選，請參閱 [使用自訂詞彙篩選](#)。

**Note**

Amazon Transcribe 會自動遮罩種族敏感詞彙，但您可以透過聯絡 [AWS 技術支援](#) 來選擇退出此預設篩選條件。

如需詞彙篩選的影片說明，請參閱 [使用詞彙篩選](#)。

**詞彙篩選特定的 API 操作**

[CreateVocabularyFilter](#), [DeleteVocabularyFilter](#), [GetVocabularyFilter](#), [ListVocabularyFilters](#), [UpdateVocabularyFilter](#)

## 建立詞彙篩選條件

有兩個選項可建立自訂詞彙篩選：

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 管理主控台 AWS CLI、或 AWS SDKs 使用此方法。
  - 如果使用 AWS 管理主控台，您可以為自訂詞彙檔案提供本機路徑或 Amazon S3 URI。
  - 如果使用 AWS CLI AWS SDKs，您必須將自訂詞彙檔案上傳至 Amazon S3 儲存貯體，並在請求中包含 Amazon S3 URI。
2. 直接在 API 請求中包含逗號分隔單字的清單。
  - 您可以使用 參數搭配 AWS CLI 或 AWS SDKs 使用此方法 [Words](#)。

如需每種方法的範例，請參閱 [建立自訂詞彙篩選](#)

建立自訂詞彙篩選時要注意的事項：

- 關鍵字不區分大小寫。例如，「curse (詛咒)」和「CURSE (詛咒)」會視為同一個單字。
- 只會篩選完全相符的字詞。例如，如果您的篩選包含「swear(宣誓)」，但您的媒體包含「swears(發誓)」或「swearing(發誓)」一詞，則不會篩選這些單字。只有「swear(發誓)」的執行個體會被篩選。因此，您必須包含要篩選的單字的所有變體。
- 篩選不適用於包含在其他單字中的單字。例如，如果詞彙篩選包含「marine(海洋)」，但不包含「submarine(潛艇)」，則「submarine(潛艇)」就會出現在您的轉錄結果中。

- 每個項目只能包含一個單字 (不可包含空格)。
- 如果您將自訂詞彙篩選儲存為文字檔案，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，每個最多可以有 50 Kb 的大小。
- 您僅可使用您的語言支援的字元。請參閱您語言的[字元集](#)，了解詳細資訊。

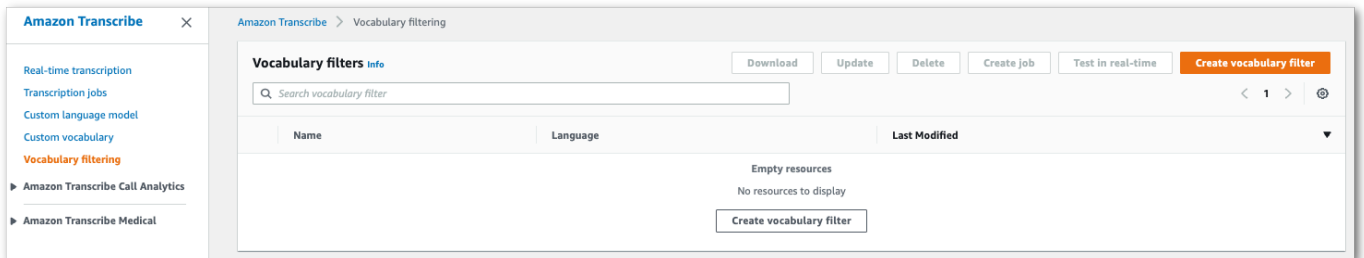
## 建立自訂詞彙篩選

若要處理自訂詞彙篩選條件以搭配使用 Amazon Transcribe，請參閱下列範例：

### AWS 管理主控台

繼續前，請將您的自訂詞彙篩選儲存為文字 (\*.txt) 檔案。您可以選擇性地將檔案上傳至儲存 Amazon S3 貯體。

1. 登入 [AWS 管理主控台](#)。
2. 在導覽窗格中，選擇詞彙篩選。這會開啟詞彙篩選頁面，您可以在其中檢視現有的自訂詞彙篩選或建立新的詞彙篩選。
3. 選擇建立詞彙篩選。



這會引導您前往建立詞彙篩選頁面。輸入新自訂詞彙篩選的名稱。

選擇詞彙輸入來源下的檔案上傳(檔案上傳)或 S3 位置選項。然後指定自訂詞彙檔案的位置。

## 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. 或者，將標籤新增至您的自訂詞彙篩選。完成所有欄位後，請選擇頁面底部的建立詞彙篩選。如果處理檔案時沒有錯誤，這會引導您回到詞彙篩選頁面。

您的自訂詞彙篩選現在可以開始使用。

## AWS CLI

此範例使用 [create-vocabulary-filter](#) 指令，將單字清單處理為可用的自訂詞彙篩選條件。如需詳細資訊，請參閱[CreateVocabularyFilter](#)。

選項 1：您可以使用 `words` 參數將單詞清單包含到請求中。

```
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 儲存貯體，然後使用 `vocabulary-filter-file-uri` 參數，在請求中加入檔案的 URI。

```
aws transcribe create-vocabulary-filter \  
--vocabulary-filter-name my-first-vocabulary-filter \  
--language-code en-US \  
--vocabulary-filter-file-uri s3://amzn-s3-demo-bucket/my-vocabulary-filters/my-vocabulary-filter.txt
```

這是使用 [create-vocabulary-filter](#) 指令的另一個範例，以及建立自訂詞彙篩選的請求內文。

```
aws transcribe create-vocabulary-filter \  
--cli-input-json file://filepath/my-first-vocab-filter.json
```

檔案 `my-first-vocab-filter.json` 包含以下請求內文。

選項 1：您可以使用 `Words` 參數將單詞清單包含到請求中。

```
{  
  "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 儲存貯體，然後使用 `VocabularyFilterFileUri` 參數，在請求中加入檔案的 URI。

```
{  
  "VocabularyFilterName": "my-first-vocabulary-filter",  
  "LanguageCode": "en-US",  
  "VocabularyFilterFileUri": "s3://amzn-s3-demo-bucket/my-vocabulary-filters/my-vocabulary-filter.txt"  
}
```

**Note**

如果您的請求中包含 `VocabularyFilterFileUri`，則無法使用 `Words`；您必須選擇其中一個。

適用於 Python (Boto3) 的 AWS SDK

此範例使用適用於 Python (Boto3) 的 AWS SDK，使用 [create\\_vocabulary\\_filter](#) 方法來建立自訂詞彙篩選條件。如需詳細資訊，請參閱 [CreateVocabularyFilter](#)。

如需使用 AWS SDKs 的其他範例，包括功能特定、案例和跨服務範例，請參閱 [Amazon Transcribe AWS SDKs 的程式碼範例](#) 章節。

選項 1：您可以使用 `Words` 參數將單詞清單包含到請求中。

```
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 儲存貯體，然後使用 `VocabularyFilterFileUri` 參數，在請求中加入檔案的 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://amzn-s3-demo-bucket/my-vocabulary-filters/my-
vocabulary-filter.txt'
```

)

**Note**

如果您的請求中包含 `VocabularyFilterFileUri`，則無法使用 `Words`；您必須選擇其中一個。

**Note**

如果您為自訂詞彙篩選檔案建立新的儲存 Amazon S3 貯體，請確定提出 `CreateVocabularyFilter` 請求 IAM 的角色具有存取此儲存貯體的許可。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指定 IAM 角色，方法是包含 `DataAccessRoleArn` 參數。如需中 IAM 角色和政策的詳細資訊 Amazon Transcribe，請參閱 [Amazon Transcribe 身分型政策範例](#)。

## 使用自訂詞彙篩選

建立自訂詞彙篩選後，您可以包含在轉錄請求中；請參閱以下章節以取得範例。

您要包含在請求中的自訂詞彙篩選的語言，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納入一個自訂詞彙篩選。如果自訂詞彙篩選的語言與您的音訊中識別的語言不符，您的篩選就不會套用至您的轉錄，且不會出現警告或錯誤。

### 在批次轉錄中使用自訂詞彙篩選

若要搭配批次轉錄使用自訂詞彙篩選，請參閱下列範例：

#### AWS 管理主控台

1. 登入 [AWS 管理主控台](#)。
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**

為工作命名並指定輸入媒體。選擇性包含任何其他欄位，然後選擇下一步。

3. 在設定工作頁面的內容移除面板中，開啟詞彙篩選。

## 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 ▼

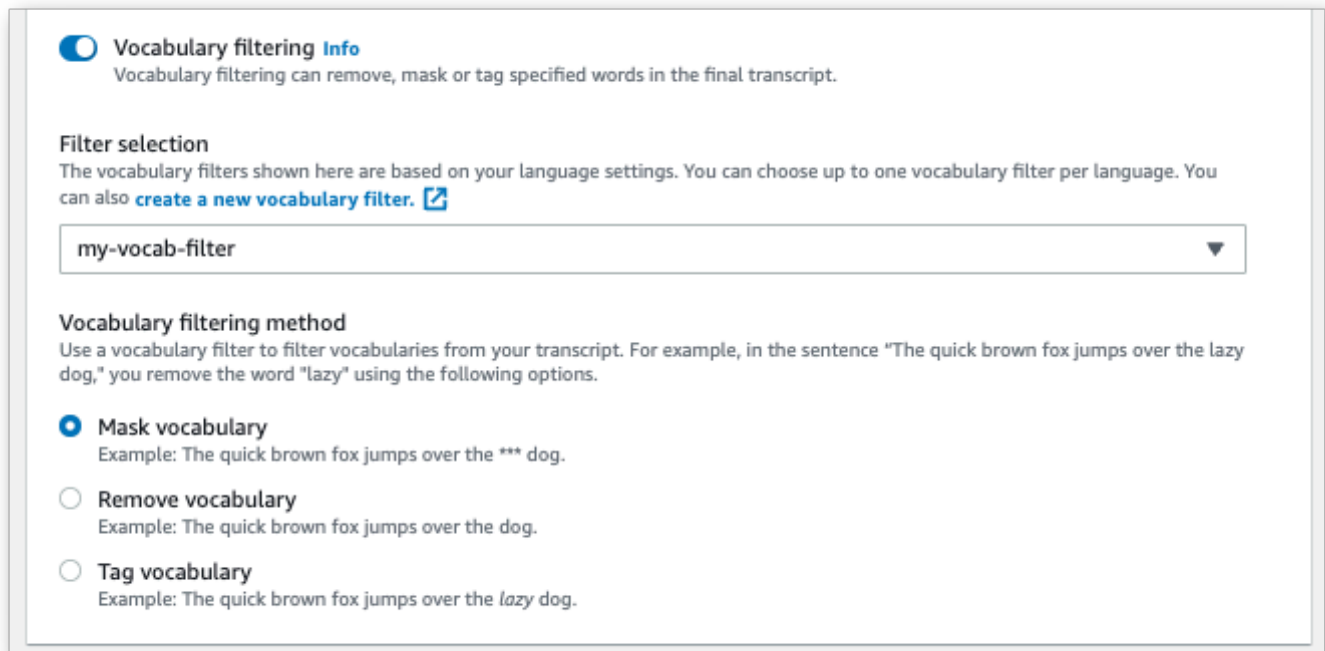
---

### 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. 從下拉式功能表中選擇自訂詞彙篩選，然後指定篩選方法。



5. 選擇建立作業以執行轉錄作業。

## AWS CLI

此範例使用 [start-transcription-job](#) 指令和 `Settings` 參數與 `VocabularyFilterName` 和 `VocabularyFilterMethod` 子參數。如需詳細資訊，請參閱 [StartTranscriptionJob](#) 和 [Settings](#)。

```
aws transcribe start-transcription-job \
--region us-west-2 \
--transcription-job-name my-first-transcription-job \
--media MediaFileUri=s3://amzn-s3-demo-bucket/my-input-files/my-media-file.flac \
--output-bucket-name amzn-s3-demo-bucket \
--output-key my-output-files/ \
--language-code en-US \
--settings VocabularyFilterName=my-first-vocabulary-filter,VocabularyFilterMethod=mask
```

這是使用 [start-transcription-job](#) 指令的另一個範例，以及包含工作的自訂詞彙篩選的請求內文。

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

檔案 `my-first-vocabulary-filter-job.json` 包含以下請求內文。

```
{
  "TranscriptionJobName": "my-first-transcription-job",
  "Media": {
    "MediaFileUri": "s3://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
  },
  "OutputBucketName": "amzn-s3-demo-bucket",
  "OutputKey": "my-output-files/",
  "LanguageCode": "en-US",
  "Settings": {
    "VocabularyFilterName": "my-first-vocabulary-filter",
    "VocabularyFilterMethod": "mask"
  }
}
```

適用於 Python (Boto3) 的 AWS SDK

此範例使用適用於 Python (Boto3) 的 AWS SDK，使用 [start\\_transcription\\_job](#) 方法的 `Settings` 數來包含自訂詞彙篩選條件。如需詳細資訊，請參閱 [StartTranscriptionJob](#) 和 [Settings](#)。

如需使用 AWS SDKs 的其他範例，包括功能特定、案例和跨服務範例，請參閱 [Amazon Transcribe AWS SDKs 的程式碼範例](#) 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://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
transcribe.start_transcription_job(
    TranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'amzn-s3-demo-bucket',
    OutputKey = 'my-output-files/',
    LanguageCode = 'en-US',
    Settings = {
        'VocabularyFilterName': 'my-first-vocabulary-filter',
        'VocabularyFilterMethod': 'mask'
    }
)
```

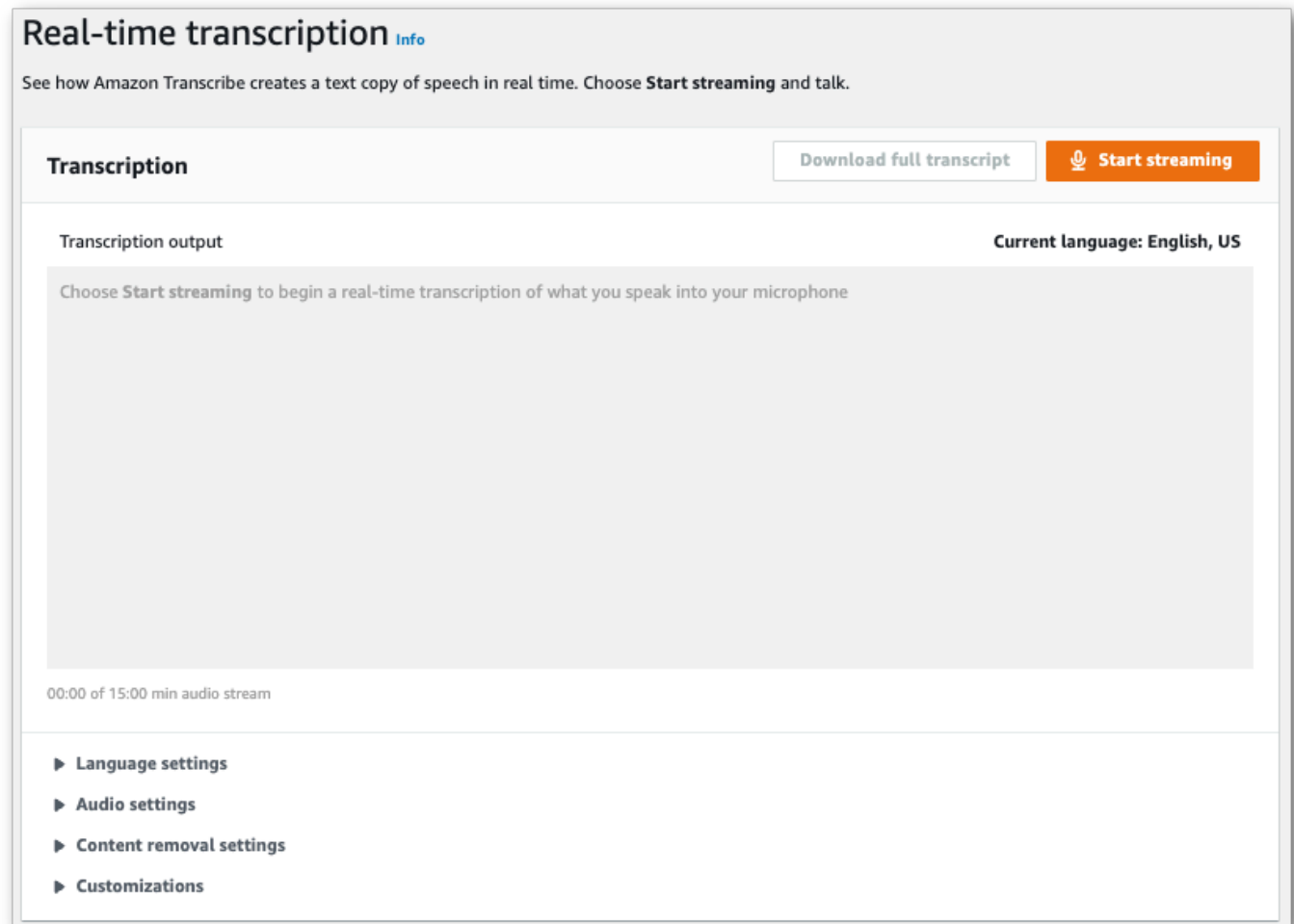
```
while True:
    status = transcribe.get_transcription_job(TranscriptionJobName = job_name)
    if status['TranscriptionJob']['TranscriptionJobStatus'] in ['COMPLETED', 'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
print(status)
```

## 在串流轉錄中使用自訂詞彙篩選

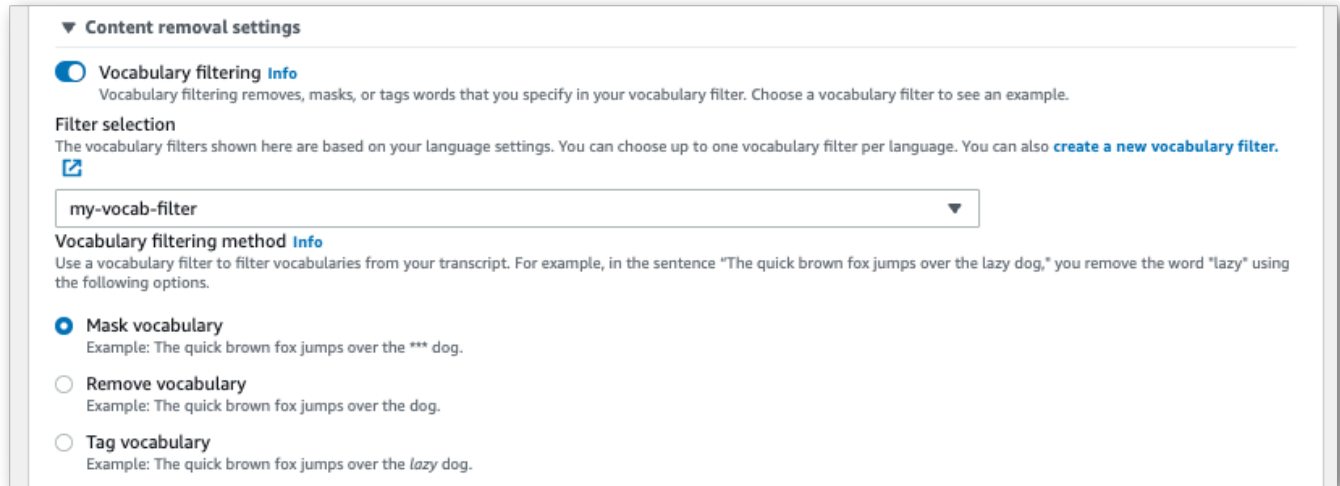
若要將自訂詞彙篩選與串流轉錄搭配使用，請參閱下列範例：

### AWS 管理主控台

1. 登入 [AWS 管理主控台](#)。
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。



### 3. 開啟詞彙篩選。從下拉式功能表中選擇自訂詞彙篩選，並指定篩選方法。



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。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

此範例會建立包含您自訂詞彙篩選和篩選方法的 HTTP/2 請求。如需搭配 HTTP/2 串流使用的詳細資訊 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-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 操作常見的參數都列在 [通用參數](#) 區段中。

## WebSocket 串流

此範例會建立個預先簽署的 URL，將您的自訂詞彙篩選套用至 WebSocket 串流。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 WebSocket 串流](#)。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  
&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 操作常用的參數都列在 [通用參數](#) 區段中。

## 偵測有毒語音

有毒語音偵測目的是幫助管理點對點對話的社群媒體平台，例如線上遊戲和社群聊天平台。使用有毒的言語可能會嚴重不利於個人、同儕團體和社群。標記有害語言可以幫助組織維持對話和平，並維護安全和包容的線上環境，讓使用者可以自由創作、共享和參與。

Amazon Transcribe 毒性偵測會利用音訊和文字的提示來識別和分類七個類別的語音型有毒內容，包括性騷擾、仇恨說話、威脅、濫用、褻瀆、侮辱和圖形。除了文字之外，Amazon Transcribe 有毒內容偵測還使用語音提示，例如音調和音高，模擬磨練語音中的有毒內容意圖。這是標準內容審核系統的改良版，該系統旨在僅關注特定術語，而無需考量意圖。

Amazon Transcribe 標記和分類有毒語音，將必須手動處理的資料量降至最低。讓內容管理員可以快速有效管理平台上的語言環境。

有毒的語言類別包括：

- 褻瀆：包含不禮貌、粗俗或冒犯性的單字、片語或縮寫的語言。
- 仇恨言論：因身份 (例如種族、民族、性別、宗教、性取向、能力和國籍) 而批評、侮辱、譴責或做出非人性化的言論。
- 性暗示：使用直接或間接引用身體部位、身體特徵或性別，表現出性興趣、活動或激起性欲的語言。
- 侮辱：包括貶低、羞辱、嘲笑、侮辱或貶低的言語。種類型的語言也被標記為霸凌。
- 暴力或威脅：包括試圖對個人或團體造成痛苦、傷害或敵意的威脅性言論。
- 圖形：使用視覺描述和令人不愉快的生動圖像式的語言。這種類型的語言通常故意拖長，以延長收件人的不適感。
- 騷擾或辱罵：目的式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。

有毒內容偵測會分析語音區段 (自然暫停間的語音)，並指定可信度分數給這些區段。可信度分數是介於 0 和 1 之間的值。可信度分數較高表示內容在相關類別中是有毒語音的可能性越高。您可以使用這些可信度分數，針對您的使用案例，設定適當的有毒內容偵測閾值。

### Note

有毒內容偵測僅適用於美國英文 (en-US) 的批次轉錄。

檢視 JSON 格式的[範例輸出](#)。

## 使用有毒語音偵測

### 在批次轉錄中使用有毒語音偵測

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 管理主控台

1. 登入 [AWS 管理主控台](#)。
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。

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 heading with the instruction 'Choose the type of model to use for the transcription job.' Below this are 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 heading 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.' Below this are 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 the text 'Choose the language of the input audio.' The selected option is 'English, US (en-US)'.
- Additional settings**: A section header with a right-pointing arrow.

3. 在指定作業詳細資訊頁面上，您也可以視需要啟用 PII 修訂。請注意，毒性偵測不支援其他列出的選項。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，選擇毒性偵測。

### 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

4. 選擇建立作業以執行轉錄作業。
5. 轉錄作業完成後，您可以從轉錄作業詳細資訊頁面的下載下拉式功能表下載您的文字記錄。

## AWS CLI

此範例使用 [start-transcription-job](#) 指令和 `ToxicityDetection` 參數。如需詳細資訊，請參閱 [StartTranscriptionJob](#) 和 [ToxicityDetection](#)。

```
aws transcribe start-transcription-job \  
--region us-west-2 \  
--transcription-job-name my-first-transcription-job \  
--media MediaFileUri=s3://amzn-s3-demo-bucket/my-input-files/my-media-file.flac \  
--output-bucket-name amzn-s3-demo-bucket \  
--output-key my-output-files/ \  
--language-code en-US \  
--toxicity-detection ToxicityCategories=ALL
```

以下是使用 [start-transcription-job](#) 指令的另一個範例，以及包含替代毒性偵測的請求內文。

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

檔案 `my-first-toxicity-job.json` 包含以下請求內文。

```
{  
  "TranscriptionJobName": "my-first-transcription-job",  
  "Media": {  
    "MediaFileUri": "s3://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"  
  },  
  "OutputBucketName": "amzn-s3-demo-bucket",  
  "OutputKey": "my-output-files/",  
  "LanguageCode": "en-US",  
  "ToxicityDetection": [  
    {  
      "ToxicityCategories": [ "ALL" ]  
    }  
  ]  
}
```

## 適用於 Python (Boto3) 的 AWS SDK

此範例使用適用於 Python (Boto3) 的 AWS SDK 為 [start\\_transcription\\_job](#) ToxicityDetection 方法啟用。如需詳細資訊，請參閱 [StartTranscriptionJob](#) 和 [ToxicityDetection](#)。

如需使用 AWS SDKs 的其他範例，包括功能特定、案例和跨服務範例，請參閱 [Amazon Transcribe AWS SDKs 的程式碼範例](#) 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://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
transcribe.start_transcription_job(
    TranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'amzn-s3-demo-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

在您的轉錄輸出中標記有毒語音並進行分類。有毒語音的每個執行個體都會分類並指定可信度分數 (介於 0 和 1 之間的值)。可信度值較高表示內容在指定類別中是有毒語音的可能性越高。

## 範例輸出(JSON)

以下是 JSON 格式的範例輸出，顯示具有相關可信度分數的分類有毒語音。

```
{
  "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"  
}
```

# 修訂或識別個人身分識別資訊

修訂可用來從您的文字記錄中遮蔽或移除敏感內容的個人身分識別資訊 (PII)。PII 的類型 Amazon Transcribe 會因批次和串流轉錄而異。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 而不修訂；如需輸出範例，請參閱 [PII 識別輸出範例](#)。

啟用修訂後，您可以選擇只產生修訂的文字記錄，或同時產生修訂後的文字記錄和未修訂的文字記錄。如果您選擇只產生修訂文字記錄，請注意，您的媒體是唯一儲存完整對話之處。如果您刪除原始媒體，則沒有未修訂 PII 的記錄。因此，除了修訂後的文字記錄之外，產生未修訂的文字記錄可能須謹慎以待。

若要了解使用批次轉錄的 PII 修訂的詳細資訊，請參閱：[在批次工作中修訂 PII](#)。

若要了解 PII 修訂或使用串流轉錄進行識別的詳細資訊，請參閱：[在即時串流中修訂或識別 PII](#)。

## Important

修訂功能旨在識別和移除敏感資料。不過，由於機器學習的預測性質，Amazon Transcribe 可能無法識別和移除文字記錄中敏感資料的所有執行個體。我們強烈建議您檢閱任何已修訂的輸出，以確保其符合您的需求。

修訂功能不符合醫療隱私法 (例如 1996 美國醫療保險流通與責任法案 (HIPAA)) 規定，去除身分識別。

如需 Amazon Transcribe 修訂功能的影片逐步解說，請參閱 [使用內容修訂來識別和修訂 PII](#)。

## 在批次工作中修訂 PII

在批次轉錄任務期間從文字記錄修訂個人身分識別資訊 (PII) 時，會在文字記錄的主要文字內文 [PII] 中將 PII 的每個已識別執行個體 Amazon Transcribe 取代為 `[REDACTED]`。您也可以在轉錄輸出的逐字部分中檢視修訂的 PII 類型。如需了解輸出範例，請參閱 [編輯輸出的範例 \(批次處理\)](#)。

批次轉錄的修訂適用於美式英文 (en-US) 和美式西班牙文 (es-US)。修訂與 [語言識別](#) 不相容。

已修訂和未修訂的文字記錄都會存放在相同的輸出 Amazon S3 儲存貯體中。會將它們 Amazon Transcribe 存放在您指定的儲存貯體中，或存放在服務管理的預設儲存 Amazon S3 貯體中。

## Amazon Transcribe 可以辨識批次轉錄的 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、MasterCard 卡和 Discover 信用卡和簽帳卡上存在 3 位數卡驗證碼 (CVV)。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 可以辨識過期日期，例如 01/21、01/2021 和 2021 年 1 月。
CREDIT_DEBIT_NUMBER	信用卡或簽帳卡號碼。這些號碼的長度從 13 到 16 位數不等，但當只有最後 4 位數時，Amazon Transcribe 也會辨識信用卡或簽帳金融卡號碼。
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 Mr.、Mrs.、Miss 或 Dr. Amazon Transcribe does，不會將此實體類型套用至屬於組織或地址的名稱。例如，會將 John Doe Organizat

PII 類型	描述
	ion Amazon Transcribe 識別為組織，並將 Jane Doe Street 識別為地址。
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) 是核發給美國公民、永久居民和臨時工作居民的 9 位數號碼。當只有最後 4 位數時，Amazon Transcribe 也會識別社會安全號碼。

您可以使用 AWS 管理主控台、AWS CLI 或 AWS SDK 啟動批次轉錄任務。

## AWS 管理主控台

1. 登入 [AWS 管理主控台](#)。
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].

- 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 [start-transcription-job](#) 指令和 `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://amzn-s3-demo-bucket/my-input-files/my-media-file.flac \
--output-bucket-name amzn-s3-demo-bucket \
--output-key my-output-files/ \
--language-code en-US \
```

```
--content-redaction
RedactionType=PII,RedactionOutput=redacted,PiiEntityTypes=NAME,ADDRESS,BANK_ACCOUNT_NUMBER
```

這是使用 [start-transcription-job](#) 方法的另一個範例，且請求主文會針對該工作修訂 PII。

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

檔案 `my-first-redaction-job.json` 包含以下請求主文。

```
{
  "TranscriptionJobName": "my-first-transcription-job",
  "Media": {
    "MediaFileUri": "s3://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
  },
  "OutputBucketName": "amzn-s3-demo-bucket",
  "OutputKey": "my-output-files/",
  "LanguageCode": "en-US",
  "ContentRedaction": {
    "RedactionOutput": "redacted",
    "RedactionType": "PII",
    "PiiEntityTypes": [
      "NAME",
      "ADDRESS",
      "BANK_ACCOUNT_NUMBER"
    ]
  }
}
```

## 適用於 Python (Boto3) 的 AWS SDK

此範例使用 適用於 Python (Boto3) 的 AWS SDK，使用 [start\\_transcription\\_job](#) 方法的 `ContentRedaction` 引數來修訂內容。如需詳細資訊，請參閱 [StartTranscriptionJob](#) 及 [ContentRedaction](#)。

如需使用 AWS SDKs 的其他範例，包括功能特定、案例和跨服務範例，請參閱 [Amazon Transcribe AWS SDKs 的程式碼範例](#) 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://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
transcribe.start_transcription_job(
    TranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'amzn-s3-demo-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

僅支援批次任務的 PII 修訂 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（中部）、歐洲（法蘭克福）、歐洲（愛爾蘭）、歐洲（倫敦）、歐洲（巴黎）、中東（巴林）、南美洲（聖保羅）、美國東部（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，會在 Entities 物件下 Amazon Transcribe 標記轉錄結果中的 PII。如需輸出範例，請參閱 [編輯流輸出範例](#) 和 [PII 識別輸出範例](#)。

您可以使用這些英文方言來修訂和識別具有串流轉錄的 PII：蘇格蘭 (en-AB)、澳洲 (en-AU)、加拿大 (en-CA)、英國 (en-GB)、愛爾蘭 (en-IE)、印度 (en-IN)、紐西蘭 (en-NZ)、美國 (en-US)、威爾斯 (en-WL) 和南非 (en-ZA)；西班牙文方言：美國 (es-US)、西班牙 (es-ES)；法文方言：法文 (fr-FR)、加拿大 (fr-CA)；葡萄牙方言：葡萄牙 (pt-PT)、巴西 (pt-BR)；義大利方言：義大利 (it-IT)；德文方言：德國 (de-DE)、瑞士 (de-CH)。

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 可以辨識串流轉錄的 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、MasterCard 卡和 Discover 信用卡和簽帳卡上存在 3 位數卡驗證碼 (CVV)。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 可以辨識過期日期，例如 01/21、01/2021 和 2021 年 1 月。

PII 類型	描述
CREDIT_DEBIT_NUMBER	信用卡或簽帳卡號碼。這些號碼的長度從 13 到 16 位數不等，但當只有最後 4 位數時，Amazon Transcribe 也會辨識信用卡或簽帳金融卡號碼。
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 Mr.、Mrs.、Miss 或 Dr. Amazon Transcribe does，不會將此實體類型套用至屬於組織或地址的名稱。例如，會將 John Doe Organization Amazon Transcribe 識別為組織，並將 Jane Doe Street 識別為地址。
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) 是核發給美國公民、永久居民和臨時工作居民的 9 位數號碼。當只有最後 4 位數時，Amazon Transcribe 也會識別社會安全號碼。
AGE	個人的年齡，包括數量和時間單位。例如，在「我 40 歲」一詞中，將「40 歲」Amazon Transcribe 視為年齡。
DATE_TIME	日期可以包含年、月、日、星期幾或一天中的時間。例如，Amazon Transcribe 會將「2020 年 1 月 19 日」或「上午 11 點」視為日期。Amazon Transcribe 會辨識部分日期、日期範圍和日期間隔。它也會辨識幾十年，例如「1990 年代」。

PII 類型	描述
LICENSE_PLATE	車輛的車牌是由註冊車輛的州或國家/地區核發。客車的格式通常為 5 到 8 位數，由大寫字母和數字組成。格式會根據發行州或國家/地區的位置而有所不同。
PASSPORT_NUMBER	指派給個人護照的唯一識別符。格式通常包含字母和數字的組合，並因國家/地區而異。
PASSWORD	用作密碼的英數字串，例如 <code>"*very20special#pass*"</code> 。
USERNAME	識別帳戶的使用者名稱，例如登入名稱、螢幕名稱、別名或控制代碼。
VEHICLE_IDENTIFICATION_NUMBER	車輛識別號碼 (VIN) 可唯一識別車輛。VIN 內容和格式在 ISO 3779 規格中定義。每個國家/地區都有 VIN 的特定代碼和格式。

您可以使用 AWS 管理主控台、WebSocket 或 HTTP/2 啟動串流轉錄。

## AWS 管理主控台

1. 登入 [AWS 管理主控台](#)。
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 串流

此範例會在 WebSocket 串流中建立使用 PII 修訂 (或 PII 識別) 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 WebSocket 串流](#)。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 操作常見的參數都列在 [通用參數](#) 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-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 操作常見的參數都列在 [通用參數](#) 區段中。

**Note**

串流的 PII 修訂僅支援以下項目 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 修訂和識別輸出範例

下列範例顯示批次和串流工作的修訂輸出，以及串流工作的 PII 識別。

使用自動內容修訂的轉錄作業會產生兩種類型的 confidence 值。自動語音辨識 (ASR) 可信度表示具有 pronunciation 的 type 或 punctuation 為特定表達用語。在下列的轉錄輸出中，單字 Good 具有 1.0 的 confidence。此可信度值表示 Amazon Transcribe 100% 確信此文字記錄中的措辭是「良好」。`[PII]` 標籤的 confidence 值是標記為修訂的語音確實是 PII 的可信度。在下列文字記錄輸出中，confidence 的 0.9999 表示 Amazon Transcribe 有 99.99% 的確信其在文字記錄中修訂的實體是 PII。

### 編輯輸出的範例 (批次處理)

```
{
  "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": [
      {
```

```

        "id": 0,
        "start_time": "2.86",
        "end_time": "3.35",
        "alternatives": [
            {
                "confidence": "1.0",
                "content": "Good"
            }
        ],
        "type": "pronunciation"
    },
    Items removed for brevity
    {
        "id": 8,
        "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"
}

```

以下是未編輯的文字記錄以供比較：

```

{
  "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
        Security number 000000000. My credit card number is 5555555555555555
        and my C V V code is 000. I hope that Amazon Transcribe is doing a good
        at redacting that personal information away. Let's check."
      }
    ],
    "items": [
      {
        "id": 0,
        "start_time": "2.86",
        "end_time": "3.35",
        "alternatives": [
          {
            "confidence": "1.0",
            "content": "Good"
          }
        ],
        "type": "pronunciation"
      },
      Items removed for brevity
      {
        "id": 8,
        "start_time": "5.56",
        "end_time": "6.25",
        "alternatives": [
          {
            "confidence": "0.9999",
            "content": "Mike",
          }
        ],
        "type": "pronunciation"
      },
      Items removed for brevity
    ],
  },
  "status": "COMPLETED"
}

```

## 編輯流輸出範例

```
{
  "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,
```



```
    },
    {
      "Content": "is",
      "EndTime": 0.7899375,
      "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
}
}
}
}
```

# 建立影片字幕

Amazon Transcribe 支援 WebVTT (\*.vtt) 和 SubRip (\*.srt) 輸出，以用作影片字幕。您可以在設定批次影片轉錄作業時，選擇一或兩種檔案類型。使用字幕功能時，系統會產生您選擇的字幕檔案和一般的文字記錄檔案 (包含其他資訊)。字幕和轉錄檔案輸出到相同的目的地。

會在朗讀文字的同時顯示字幕，並維持可見，直到自然暫停或發言者完成通話為止。請注意，如果您在轉錄請求中啟用字幕，且您的音訊不包含語音，則不會建立字幕檔案。

## Important

Amazon Transcribe 使用的預設開始索引 0 作為字幕輸出，這與更廣泛使用的值不同<sup>1</sup>。如果您需要的開始索引 1，您可以使用 [OutputStartIndex](#) 參數在 AWS 管理主控台 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，因此在建立字幕前，請務必確認您需要的開始索引。如果您不確定要使用的值，我們建議您選擇 1。請參閱 [Subtitles](#)，了解詳細資訊。

支援字幕的功能：

- 內容修訂 — 任何修訂後的內容會在字幕和一般文字記錄輸出檔案中顯示為「PII」。音訊不會改變。
- 詞彙篩選器 — 字幕檔案是從轉錄檔案產生，因此您在標準轉錄輸出中篩選的任何單字也會在字幕篩選。篩選後的內容會在您的文字記錄和字幕檔案中顯示為空格或 \*\*\*。音訊不會改變。
- 發言者分段標記 — 如果指定的字幕區段有多個發言者，則使用破折號以區分每個發言者。這同時適用於 WebVTT 和 SubRip 格式；例如：
  - --發言者 1 說出的文字
  - --發言者 2 說出的文字

字幕檔案會存放在與轉錄輸出相同的 Amazon S3 位置。

請參閱 [Amazon Transcribe 影片短片：在不編寫任何代碼的情況下建立影片字幕](#)，取得建立字幕的影片說明。

# 產生字幕檔案

您可以使用 AWS 管理主控台、AWS CLI 或 AWS SDK 建立字幕檔案；請參閱下列範例：

## AWS 管理主控台

1. 登入 [AWS 管理主控台](#)。
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。字幕選項位於輸出資料面板。
3. 選擇字幕檔案所需的格式，然後為開始索引選擇值。請注意，Amazon Transcribe 預設值為 0，但使用範圍 1 更為廣泛。如果您不確定要使用的值，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。
5. 選擇建立作業以執行轉錄作業。

## AWS CLI

此範例使用 [start-transcription-job](#) 指令和 `Subtitles` 參數。如需詳細資訊，請參閱 [StartTranscriptionJob](#) 和 [Subtitles](#)。

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

```

```
--transcription-job-name my-first-transcription-job \  
--media MediaFileUri=s3://amzn-s3-demo-bucket/my-input-files/my-media-file.flac \  
--output-bucket-name amzn-s3-demo-bucket \  
--output-key my-output-files/ \  
--language-code en-US \  
--subtitles Formats=vtt,srt,OutputStartIndex=1
```

這是使用 [start-transcription-job](#) 指令的另一個範例，以及新增字幕到工作的請求內文。

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

檔案 *my-first-subtitle-job.json* 包含以下請求內文。

```
{  
  "TranscriptionJobName": "my-first-transcription-job",  
  "Media": {  
    "MediaFileUri": "s3://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"  
  },  
  "OutputBucketName": "amzn-s3-demo-bucket",  
  "OutputKey": "my-output-files/",  
  "LanguageCode": "en-US",  
  "Subtitles": {  
    "Formats": [  
      "vtt", "srt"  
    ],  
    "OutputStartIndex": 1  
  }  
}
```

## 適用於 Python (Boto3) 的 AWS SDK

此範例使用 適用於 Python (Boto3) 的 AWS SDK，使用 [start\\_transcription\\_job](#) 方法的 Subtitles 引數來新增字幕。如需詳細資訊，請參閱 [StartTranscriptionJob](#) 和 [Subtitles](#)。

如需使用 AWS SDKs 的其他範例，包括功能特定、案例和跨服務範例，請參閱 [Amazon Transcribe AWS SDKs 的程式碼範例](#) 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://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
transcribe.start_transcription_job(
    TranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'amzn-s3-demo-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 通話分析來深入了解客戶與客服人員的互動。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 儲存貯體中的檔案，則您要執行通話後轉錄。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](#)。您的通話後分析文字記錄會存放在您在請求中指定的 Amazon S3 儲存貯體中。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](#)  
即

時：[StartCallAnalyticsStreamTranscription](#)，[StartCallAnalyticsStreamTranscriptionWebSoc](#)

## 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 Call Analytics 可用的功能與 Amazon Transcribe 和 Amazon Transcribe Medical 的功能進行比較，請參閱[功能表](#)。

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 [工作佇列](#)，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。對於頻率大於 8kHz 的音訊檔案，音訊持續時間必須小於或等於 2 小時。請注意，某些壓縮的非 WAV 音訊檔案格式的檔案大小限制可能較小。
- 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](#)。

- 有些通話分析配額與 Amazon Transcribe 和 Amazon Transcribe Medical 不同；如需詳細資訊，請參閱 [AWS 一般參考](#)。

**i** 使用 AWS Machine Learning 部落格深入探索

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](#)

若要檢視通話分析輸出和功能範例，請參閱我們的 [GitHub 示範](#)。我們還提供 [JSON to Word document](#) 應用程式，將您的文字記錄轉換為方便閱讀的格式。

## 區域可用性和配額

下列支援通話分析 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潔摘要，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% 的對話輪換 (兩個說話者之間的back-and-forth) 符合指定的情緒，會將情緒 Amazon Transcribe 視為相符。

- **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 (en-US) 和美國西班牙文 () 支援通話後 PII 修訂 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 [修訂或識別個人身分識別資訊](#)。

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。

情緒分析立即可用，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

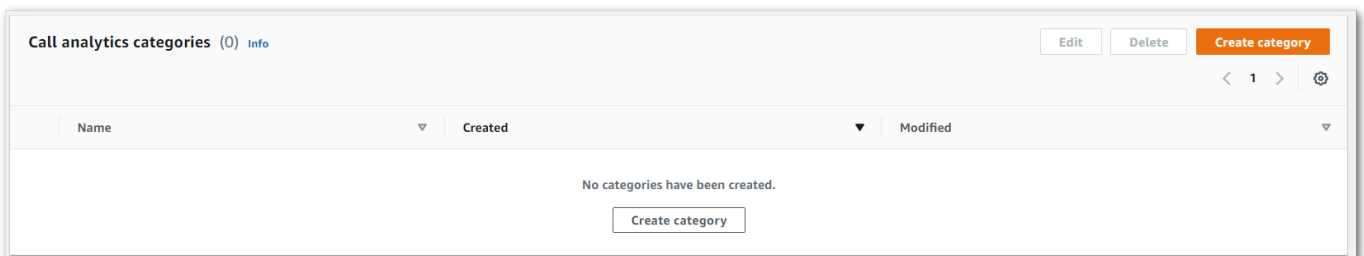
## 通話後與即時類別

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 管理主控台、AWS CLI 或 AWS SDK；請參閱下列範例：

### AWS 管理主控台

1. 在導覽窗格中，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.

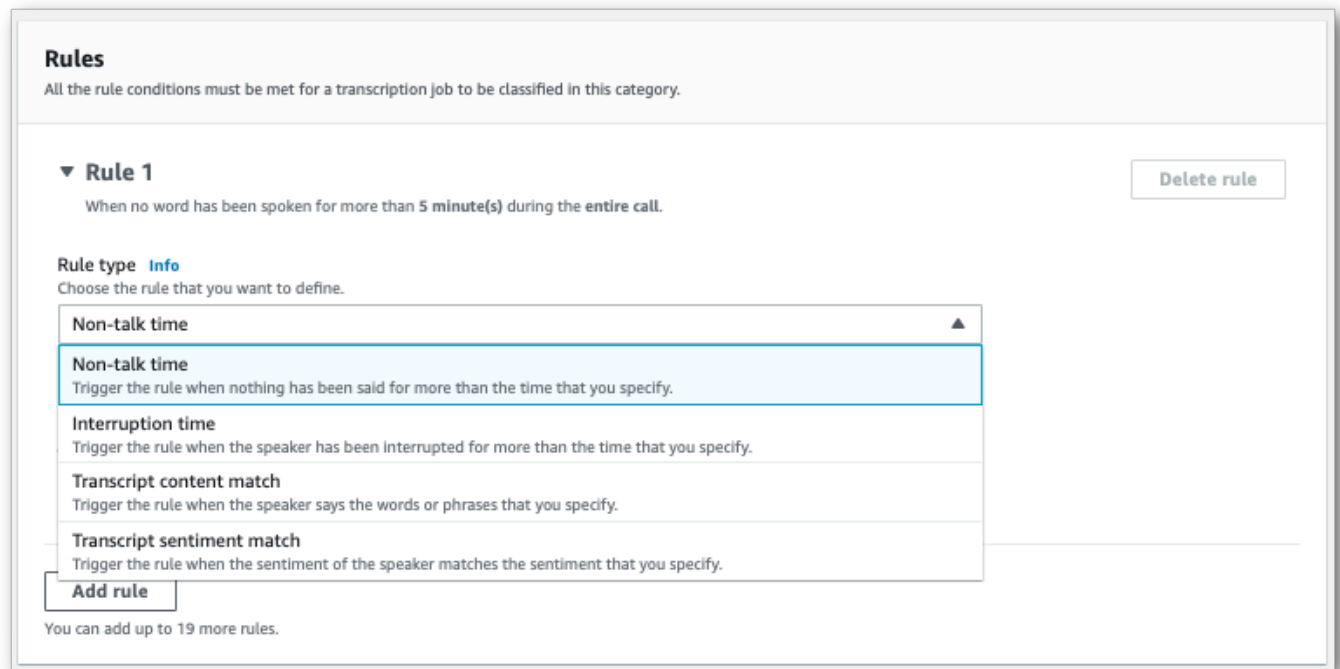
▼ Rule 1 Delete rule

**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** ▲  
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  ▼

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"
    }
  }
]
}

```

適用於 Python (Boto3) 的 AWS SDK

此範例會使用 [create\\_call\\_analytics\\_category](#) 方法的 `CategoryName` 和 `Rules` 引數 適用於 Python (Boto3) 的 AWS SDK 來建立類別。如需詳細資訊，請參閱 [CreateCallAnalyticsCategory](#)、[CategoryProperties](#) 及 [Rule](#)。

如需使用 AWS SDKs 的其他範例，包括功能特定、案例和跨服務範例，請參閱 [Amazon Transcribe AWS SDKs 的程式碼範例](#) 章節。

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

開始通話後分析轉錄之前，您必須建立 Amazon Transcribe 要在音訊中符合的所有**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。

若要開始通話後分析轉錄，您可以使用 AWS 管理主控台、AWS CLI 或 AWS SDK；請參閱下列範例：

### AWS 管理主控台

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. 在導覽窗格的 Amazon Transcribe 通話分析下，選擇通話分析任務。
2. 選擇建立任務。

## 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 IAM 的角色。

### 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://amzn-s3-demo-bucket/my-input-files/my-media-file.flac \  
--output-location s3://amzn-s3-demo-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://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"  
  },  
  "OutputLocation": "s3://amzn-s3-demo-bucket/my-output-files/",  
  "ChannelDefinitions": [  
    {  
      "ChannelId": 0,  
      "ParticipantRole": "AGENT"  
    },  
    {  
      "ChannelId": 1,  
      "ParticipantRole": "CUSTOMER"  
    }  
  ]  
}
```

```
}
```

## 適用於 Python (Boto3) 的 AWS SDK

此範例使用適用於 Python (Boto3) 的 AWS SDK 來啟動使用 [start\\_call\\_analytics\\_job](#) 方法的通話分析任務。如需詳細資訊，請參閱[StartCallAnalyticsJob](#)及[ChannelDefinition](#)。

如需使用 AWS SDKs 的其他範例，包括功能特定、案例和跨服務範例，請參閱 [Amazon Transcribe AWS SDKs 的程式碼範例](#) 章節。

```
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://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
output_location = "s3://amzn-s3-demo-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

通話後分析文字記錄會依區段逐段顯示。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，分析任務將在沒有產生摘要的情況下完成：

- 對話內容不足：對話必須至少包含客服人員和客戶的轉彎。當對話內容不足時，服務會傳回錯誤碼 `INSUFFICIENT_CONVERSATION_CONTENT`。
- 安全護欄：對話必須符合適當的安全護欄，以確保產生適當的摘要。不符合這些護欄時，服務會傳回錯誤碼 `FAILED_SAFETY_GUIDELINES`。

您可以在輸出中的 `Skipped` 區段 `AnalyticsJobDetails` 中找到錯誤碼。您也可以在此 [GetCallAnalyticsJob](#) API 回應 `CallAnalyticsJobDetails` 中找到中的錯誤原因。

### 範例錯誤輸出

```
{
  "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。

- 定性逐段情緒值：

```
"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

採用 Amazon Bedrock 技術：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 管理主控台

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)

## 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://amzn-s3-demo-bucket/my-input-files/my-media-file.flac \  
--output-location s3://amzn-s3-demo-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://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"  
  },  
  "OutputLocation": "s3://amzn-s3-demo-bucket/my-output-files/",  
  "ChannelDefinitions": [  
    {  
      "ChannelId": 0,  
      "ParticipantRole": "AGENT"  
    },  
    {  
      "ChannelId": 1,  
      "ParticipantRole": "CUSTOMER"  
    }  
  ]  
}
```

```
    }
  ],
  "Settings": {
    "Summarization":{
      "GenerateAbstractiveSummary": true
    }
  }
}
```

### 適用於 Python (Boto3) 的 AWS SDK

此範例使用 適用於 Python (Boto3) 的 AWS SDK 啟動通話分析，並使用 [start\\_call\\_analytics\\_job](#) 方法啟用摘要。如需詳細資訊，請參閱[StartCallAnalyticsJob](#)。

如需使用 AWS SDKs 的其他範例，包括功能特定、案例和跨服務範例，請參閱 [Amazon Transcribe AWS SDKs 的程式碼範例](#) 章節。

```
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://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
output_location = "s3://amzn-s3-demo-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](#)
- [語言識別](#)會偵測每個音訊頻道中的主要語言

除了即時通話分析之外，Amazon Transcribe 還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，如果您為片語「我想要與管理員交談」設定篩選條件，則為該確切片語設定 Amazon Transcribe 篩選條件。

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](#)。

### 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

### 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。立即可用，因此不支援自訂，例如模型培訓或自訂類別。

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。

**Note**

這些英文方言支援即時 PII 識別：澳洲 (en-AU)、英國 (en-GB)、美國 (en-US) 和西班牙文方言 (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。

若要檢視使用此功能識別的 PII 清單，或使用 [進一步了解 PII 識別 Amazon Transcribe](#)，請參閱 [修訂或識別個人身分識別資訊](#)。

這是 [輸出範例](#)。

## PII(敏感資料) 修訂

敏感資料修訂會以 PII(例如範例 [NAME]) 取代文字記錄中的個人身分識別資訊 (PII)。此參數有助於保護客戶資訊。

**Note**

這些英文方言支援即時 PII 修訂：澳洲 (en-AU)、英國 (en-GB)、美國 (en-US) 和西班牙文方言 (es-US)。

使用即時通話分析，進行每個完整音訊區段的 PII 修訂。

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 [修訂或識別個人身分識別資訊](#)。

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。每個語音區段提供此指標值，並以定性值 (positive、neutral、mixed 或 negative) 表示。

使用此參數，您可以定性評估每個通話參與者的整體情緒，以及每個語音區段期間每位參與者的情緒。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。

每個完整的音訊區段會使用即時通話分析執行情緒分析。

情緒分析立即可用，因此不支援自訂，例如模型訓練或自訂類別。

這是[輸出範例](#)。

## 語言識別

語言識別會在即時通話分析期間，自動辨識並判斷音訊串流每個頻道內的主要語言。一旦識別，通話分析會根據偵測到的語言處理並傳回最合適的轉錄，即時透過串流傳回此資訊。

此功能可讓您自動識別和識別音訊串流每個頻道中所使用的主要語言。偵測到語言後，通話分析會即時處理和交付已識別語言的適當轉錄。

目前支援串流轉錄的所有 [Call Analytics 串流支援語言](#) 都支援自動語言識別，無需額外費用，並且可在 Call Analytics 串流支援的 [AWS 區域中](#) 使用。

### Important

通話分析僅支援單一語言識別，可識別音訊頻道中說的主要語言。不支援多語言識別，這表示每個頻道只能轉錄為一種語言。

若要使用語言識別，您必須提供至少兩個語言代碼和最多五個語言代碼，而且您只能從[支援的 Call Analytics 串流](#)語言中，為每個串流選擇一個語言方言。這表示您無法選取 en-US 和 en-AU 做為相同轉錄的語言選項。使用此功能時，LanguageCode 參數必須在請求中保持 null，因為 LanguageCode 和 IdentifyLanguage 是互斥選項。

### Warning

如果您指定的語言代碼不符合實際口語語言，系統會從您的選項中選取最相似的語言，這可能會導致文字記錄不準確。

使用語言識別功能，您可以：

- 自動即時偵測主要語言
- 在不同的管道中處理不同的語言
- 接收語言偵測的可信度分數
- 套用特定語言的自訂詞彙

若要使用語言識別，您必須設定下列參數：

### 必要參數：

- `identifyLanguage` - 設定為 `true` 以啟用語言識別。
- `languageOptions` - 當 `identifyLanguage` 設為 `true` 時可能使用的語言代碼清單。您必須提供至少兩種語言選擇，因為不支援單一語言選擇。

### 選用參數：

- `preferredLanguage` - 來自所提供 `languageOptions` 的預期主要語言。新增偏好的語言可協助 Call Analytics 更快識別語言。
- `vocabularyNames` - 自訂詞彙名稱以提高準確性。請注意，詞彙名稱區分大小寫，如果自訂詞彙的語言與識別的媒體語言不相符，則不會套用到轉錄。
- `vocabularyFilterNames` - 詞彙篩選名稱，以自訂文字記錄輸出。

這是[輸出範例](#)。

## 建立即時轉錄的類別

即時通話分析支援建立自訂類別，您可以使用該類別量身打造文字記錄分析，以最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。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。

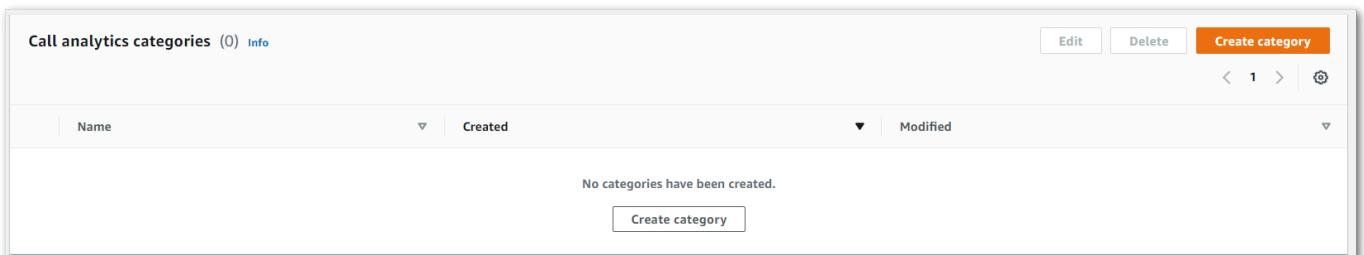
### 通話後與即時類別

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 管理主控台、AWS CLI 或 AWS SDK；請參閱下列範例：

## AWS 管理主控台

1. 在導覽窗格中，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」。

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](#)**

Real time call analytics ▼

**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.

Choose a template ▲

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.

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"
      }
    }
  ]
}
```

### 適用於 Python (Boto3) 的 AWS SDK

此範例使用適用於 Python (Boto3) 的 AWS SDK，使用 [create\\_call\\_analytics\\_category](#) 方法的 `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 SDKs 的其他範例，包括功能特定、案例和跨服務範例，請參閱 [Amazon Transcribe AWS SDKs 的程式碼範例](#) 章節。

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與[來自音訊檔案的通話後分析](#)類似的文字記錄，並將其存放在中指定的 Amazon S3 儲存貯體中 OutputLocation。此外，通話後分析會記錄您的音訊串流，並將其儲存為相同儲存 Amazon S3 貯體中的音訊檔案 (WAV 格式)。如果您啟用修訂，修訂的文字記錄和修訂後的音訊檔案也會存放在指定的儲存 Amazon S3 貯體中。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。對於持續時間超過 90 分鐘的長時間執行音訊串流，轉錄也可能無法提供通話後分析。

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：您想要存放通話後文字記錄的 Amazon S3 儲存貯體。
- 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開始即時通話分析轉錄，您可以使用 AWS 管理主控台(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 管理主控台 僅提供具有預先載入音訊範例的即時通話分析示範。如果您想要使用自己的音訊，您必須使用 API(HTTP/2、WebSockets 或 SDK)。

## 通話後分析輸出範例

通話後文字記錄會依區段逐段顯示。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

開始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。

若要開始即時通話分析轉錄，您可以使用 AWS 管理主控台、HTTP/2 或 WebSockets；請參閱下列範例：

### Important

目前，AWS 管理主控台 僅提供具有預先載入音訊範例的即時通話分析示範。如果您想要使用自己的音訊，您必須使用 API(HTTP/2、WebSockets 或 SDK)。

## AWS 管理主控台

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 管理主控台可使用示範。若要啟動自訂即時分析轉錄，您必須使用 [API](#)。

## 1. 在導覽窗格的 Amazon Transcribe 通話分析下，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。



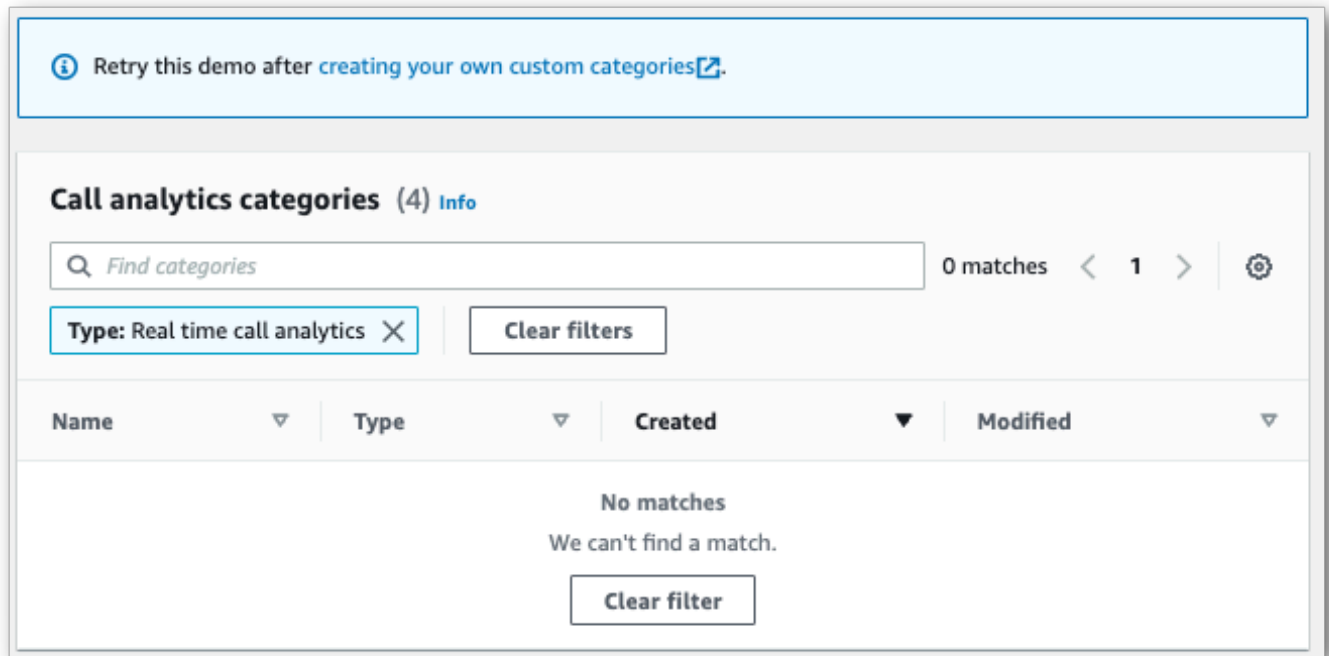
### Step 1: Specify input audio

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://amzn-s3-demo-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 操作通用的參數都列在 [通用參數](#) 區段中。

## WebSocket 串流

此範例會建立預先簽署 URL，並在 WebSocket 串流中使用通話分析。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 WebSocket 串流](#)。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://amzn-s3-demo-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 操作通用的參數會列在 [通用參數](#) 區段中。

### Tip

以上 HTTP/2 和 WebSocket 的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。

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

即時通話分析文字記錄會依區段逐段顯示。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
    }
]
},

```

## 語言識別

以下是轉錄輸出中語言識別的外觀。

```

{
  "LanguageCode": "en-US",
  "LanguageIdentification": [
    {
      "Code": "en-US",
      "Score": 0.823
    },
    {
      "Code": "de-DE",
      "Score": 0.177
    }
  ]
}

```

## 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
    }
}],
"LanguageCode": "en-US",
"LanguageIdentification": [
    {
        "Code": "en-US",
        "Score": 0.823
    },
    {
        "Code": "de-DE",
        "Score": 0.177
    }
]
},
"CategoryEvent": {
    "MatchedCategories": [
```

```
        "network-complaint"  
    ],  
    "MatchedDetails": {  
        "network issues" : {  
            "TimestampRanges": [  
                {  
                    "BeginOffsetMillis": 9299375,  
                    "EndOffsetMillis": 7899375  
                }  
            ]  
        }  
    }  
}
```

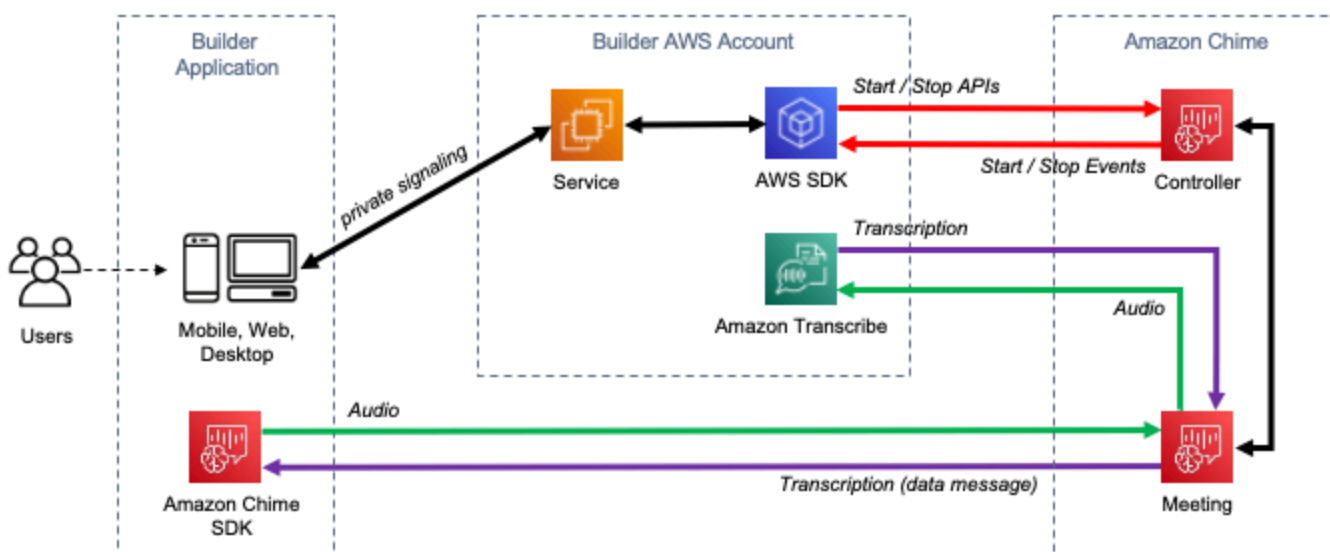
## 即時轉錄您的 Amazon Chime 呼叫

Amazon Transcribe 已與 Amazon Chime SDK 整合，可促進 Amazon Chime 即時轉錄通話。

當您使用 Amazon Chime SDK API 請求轉錄時，Amazon Chime 會開始串流音訊到 Amazon Transcribe 並在通話期間繼續這樣做。

SDK Amazon Chime 使用其「主動發言者」演算法來選取前兩個作用中發言者，然後透過單一串流將其音訊 Amazon Transcribe 以兩個不同頻道傳送至 Amazon Chime Meeting。會議參與者會透過 Amazon Chime SDK 資料訊息接收使用者歸屬的轉錄。您可以在 [《Amazon Chime SDK 開發人員指南》](#) 中檢視傳遞範例。

Amazon Chime 轉錄的資料流程如下圖所示：



如需如何設定即時 Amazon Chime 轉錄的其他資訊和詳細說明，請參閱 SDK Amazon Chime 開發人員指南中的 [使用 Amazon Chime SDK 即時轉錄](#)。如需 API 操作，請參閱 [Amazon Chime SDK API 參考資料](#)。

### 📌 使用 AWS Machine Learning 部落格深入探索

若要了解透過即時轉錄提升準確性的更多資訊，請參閱：

- [Amazon Chime SDK 會議現在支援使用 Amazon Transcribe 和 進行即時轉錄 Amazon Transcribe Medical](#)
- [Amazon Chime 適用於遠距醫療解決方案的 SDK](#)

# Amazon Transcribe AWS SDKs的程式碼範例

下列程式碼範例示範如何搭配 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中查看內容中的動作。

案例是向您展示如何呼叫服務中的多個函數或與其他 AWS 服務組合來完成特定任務的程式碼範例。

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 AWS SDKs 的 Amazon Transcribe 基本範例](#)
  - [使用 AWS SDKs 的 Amazon Transcribe 動作](#)
    - [CreateVocabulary 搭配 AWS SDK 或 CLI 使用](#)
    - [DeleteMedicalTranscriptionJob 搭配 AWS SDK 或 CLI 使用](#)
    - [DeleteTranscriptionJob 搭配 AWS SDK 或 CLI 使用](#)
    - [DeleteVocabulary 搭配 AWS SDK 或 CLI 使用](#)
    - [GetTranscriptionJob 搭配 AWS SDK 或 CLI 使用](#)
    - [GetVocabulary 搭配 AWS SDK 或 CLI 使用](#)
    - [ListMedicalTranscriptionJobs 搭配 AWS SDK 或 CLI 使用](#)
    - [ListTranscriptionJobs 搭配 AWS SDK 或 CLI 使用](#)
    - [ListVocabularies 搭配 AWS SDK 或 CLI 使用](#)
    - [StartMedicalTranscriptionJob 搭配 AWS SDK 或 CLI 使用](#)
    - [StartTranscriptionJob 搭配 AWS SDK 或 CLI 使用](#)
    - [UpdateVocabulary 搭配 AWS SDK 或 CLI 使用](#)
  - [使用 AWS SDKs Amazon Transcribe 案例](#)
    - [建置 Amazon Transcribe 串流應用程式](#)
    - [使用 AWS SDK 將文字轉換為語音並返回文字](#)
    - [使用 SDK 建立和精簡 Amazon Transcribe 自訂詞彙 AWS](#)
    - [使用 AWS SDK 透過 Amazon Transcribe 轉錄音訊並取得任務資料](#)

## 使用 AWS SDKs 的 Amazon Transcribe 基本範例

下列程式碼範例示範如何搭配 AWS SDK 使用 Amazon Transcribe 的基本功能。

### 範例

- [使用 AWS SDKs 的 Amazon Transcribe 動作](#)
  - [CreateVocabulary 搭配 AWS SDK 或 CLI 使用](#)
  - [DeleteMedicalTranscriptionJob 搭配 AWS SDK 或 CLI 使用](#)
  - [DeleteTranscriptionJob 搭配 AWS SDK 或 CLI 使用](#)
  - [DeleteVocabulary 搭配 AWS SDK 或 CLI 使用](#)
  - [GetTranscriptionJob 搭配 AWS SDK 或 CLI 使用](#)
  - [GetVocabulary 搭配 AWS SDK 或 CLI 使用](#)
  - [ListMedicalTranscriptionJobs 搭配 AWS SDK 或 CLI 使用](#)
  - [ListTranscriptionJobs 搭配 AWS SDK 或 CLI 使用](#)
  - [ListVocabularies 搭配 AWS SDK 或 CLI 使用](#)
  - [StartMedicalTranscriptionJob 搭配 AWS SDK 或 CLI 使用](#)
  - [StartTranscriptionJob 搭配 AWS SDK 或 CLI 使用](#)
  - [UpdateVocabulary 搭配 AWS SDK 或 CLI 使用](#)

## 使用 AWS SDKs 的 Amazon Transcribe 動作

下列程式碼範例示範如何使用 AWS SDKs 執行個別 Amazon Transcribe 動作。每個範例均包含 GitHub 的連結，您可以在連結中找到設定和執行程式碼的相關說明。

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。您可以在 [使用 AWS SDKs Amazon Transcribe 案例](#) 中查看內容中的動作。

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

- [CreateVocabulary 搭配 AWS SDK 或 CLI 使用](#)
- [DeleteMedicalTranscriptionJob 搭配 AWS SDK 或 CLI 使用](#)
- [DeleteTranscriptionJob 搭配 AWS SDK 或 CLI 使用](#)
- [DeleteVocabulary 搭配 AWS SDK 或 CLI 使用](#)

- [GetTranscriptionJob 搭配 AWS SDK 或 CLI 使用](#)
- [GetVocabulary 搭配 AWS SDK 或 CLI 使用](#)
- [ListMedicalTranscriptionJobs 搭配 AWS SDK 或 CLI 使用](#)
- [ListTranscriptionJobs 搭配 AWS SDK 或 CLI 使用](#)
- [ListVocabularies 搭配 AWS SDK 或 CLI 使用](#)
- [StartMedicalTranscriptionJob 搭配 AWS SDK 或 CLI 使用](#)
- [StartTranscriptionJob 搭配 AWS SDK 或 CLI 使用](#)
- [UpdateVocabulary 搭配 AWS SDK 或 CLI 使用](#)

## CreateVocabulary 搭配 AWS SDK 或 CLI 使用

下列程式碼範例示範如何使用 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

適用於 .NET 的 SDK

### Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
/// <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《適用於 .NET 的 AWS SDK API 參考》中的 [CreateVocabulary](#)。

## 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。針對 `vocabulary-file-uri`，指定該文字檔案的 Amazon Simple Storage Service (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://amzn-s3-demo-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》中的 [CreateVocabulary](#)。

## Python

### 適用於 Python 的 SDK (Boto3)

#### Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
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《適用於 Python (Boto3) 的 AWS SDK API 參考資料》中的 [CreateVocabulary](#)。

## SAP ABAP

### 適用於 SAP ABAP 的開發套件

#### Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```

TRY.
  IF it_phrases IS NOT INITIAL.
    oo_result = lo_tnb->createvocabulary(
      iv_vocabularyname = iv_vocabulary_name
      iv_languagecode = iv_language_code
      it_phrases = it_phrases ).
  ELSEIF iv_vocab_file_uri IS NOT INITIAL.
    oo_result = lo_tnb->createvocabulary(
      iv_vocabularyname = iv_vocabulary_name
      iv_languagecode = iv_language_code
      iv_vocabularyfileuri = iv_vocab_file_uri ).
  ENDIF.
  MESSAGE 'Custom vocabulary created.' TYPE 'I'.

```

```
CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
  MESSAGE lo_bad_request_ex TYPE 'I'.
  RAISE EXCEPTION lo_bad_request_ex.
CATCH /aws1/cx_tnblimitexceededex INTO DATA(lo_limit_ex).
  MESSAGE lo_limit_ex TYPE 'I'.
  RAISE EXCEPTION lo_limit_ex.
CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).
  MESSAGE lo_internal_ex TYPE 'I'.
  RAISE EXCEPTION lo_internal_ex.
CATCH /aws1/cx_tnbconflictexception INTO DATA(lo_conflict_ex).
  MESSAGE lo_conflict_ex TYPE 'I'.
  RAISE EXCEPTION lo_conflict_ex.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 [CreateVocabulary](#)。

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。

## DeleteMedicalTranscriptionJob 搭配 AWS SDK 或 CLI 使用

下列程式碼範例示範如何使用 DeleteMedicalTranscriptionJob。

.NET

適用於 .NET 的 SDK

### Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
/// <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《適用於 .NET 的 AWS SDK API 參考》中的 [DeleteMedicalTranscriptionJob](#)。

## 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》中的 [DeleteMedicalTranscriptionJob](#)。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [DeleteMedicalTranscriptionJob](#)。

## JavaScript

### 適用於 JavaScript (v3) 的 SDK

#### Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

#### 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《[適用於 JavaScript 的 AWS SDK 開發人員指南](#)》。
- 如需 API 詳細資訊，請參閱《[適用於 JavaScript 的 AWS SDK API 參考](#)》中的 [DeleteMedicalTranscriptionJob](#)。

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。

## DeleteTranscriptionJob 搭配 AWS SDK 或 CLI 使用

下列程式碼範例示範如何使用 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

適用於 .NET 的 SDK

### Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
/// <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《適用於 .NET 的 AWS SDK API 參考》中的 [DeleteTranscriptionJob](#)。

## 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》中的 [DeleteTranscriptionJob](#)。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [DeleteTranscriptionJob](#)。

## JavaScript

適用於 JavaScript (v3) 的 SDK

### Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

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 [《適用於 JavaScript 的 AWS SDK 開發人員指南》](#)。
- 如需 API 的詳細資訊，請參閱 [《適用於 JavaScript 的 AWS SDK API 參考資料》](#) 中的 [DeleteTranscriptionJob](#)。

## Python

### 適用於 Python (Boto3) 的 SDK

#### Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
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《適用於 Python (Boto3) 的 AWS SDK API 參考資料》中的 [DeleteTranscriptionJob](#)。

## SAP ABAP

### 適用於 SAP ABAP 的開發套件

#### Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
TRY.  
  lo_tnb->deletetranscriptionjob( iv_job_name ).  
  MESSAGE 'Transcription job deleted.' TYPE 'I'.  
CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).  
  MESSAGE lo_bad_request_ex TYPE 'I'.  
  RAISE EXCEPTION lo_bad_request_ex.  
CATCH /aws1/cx_tnblimitexceededex INTO DATA(lo_limit_ex).  
  MESSAGE lo_limit_ex TYPE 'I'.  
  RAISE EXCEPTION lo_limit_ex.  
CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).  
  MESSAGE lo_internal_ex TYPE 'I'.  
  RAISE EXCEPTION lo_internal_ex.  
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 [DeleteTranscriptionJob](#)。

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。

## DeleteVocabulary 搭配 AWS SDK 或 CLI 使用

下列程式碼範例示範如何使用 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

適用於 .NET 的 SDK

### Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
/// <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《適用於 .NET 的 AWS SDK API 參考》中的 [DeleteVocabulary](#)。

## 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》中的 [DeleteVocabulary](#)。

## Python

### 適用於 Python 的 SDK (Boto3)

#### Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
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《適用於 Python (Boto3) 的 AWS SDK API 參考資料》中的 [DeleteVocabulary](#)。

## SAP ABAP

### 適用於 SAP ABAP 的開發套件

#### Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
TRY.  
  lo_tnb->deletevocabulary( iv_vocabulary_name ).  
  MESSAGE 'Vocabulary deleted.' TYPE 'I'.  
CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).  
  MESSAGE lo_bad_request_ex TYPE 'I'.  
CATCH /aws1/cx_tnblimitexceedex INTO DATA(lo_limit_ex).  
  MESSAGE lo_limit_ex TYPE 'I'.  
  RAISE EXCEPTION lo_limit_ex.  
CATCH /aws1/cx_tnbnotfoundexception INTO DATA(lo_not_found_ex).  
  MESSAGE lo_not_found_ex TYPE 'I'.  
CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).  
  MESSAGE lo_internal_ex TYPE 'I'.  
  RAISE EXCEPTION lo_internal_ex.  
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 [DeleteVocabulary](#)。

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。

## GetTranscriptionJob 搭配 AWS SDK 或 CLI 使用

下列程式碼範例示範如何使用 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

### 適用於 .NET 的 SDK

#### Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
/// <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《適用於 .NET 的 AWS SDK API 參考》中的 [GetTranscriptionJob](#)。

## 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` 參數來查看您在此作業中轉錄的音訊檔案。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://amzn-s3-demo-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》中的 [GetTranscriptionJob](#)。

## Python

### 適用於 Python 的 SDK (Boto3)

#### Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
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《適用於 Python (Boto3) 的 AWS SDK API 參考資料》中的 [GetTranscriptionJob](#)。

## SAP ABAP

### 適用於 SAP ABAP 的開發套件

#### Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
TRY.  
    oo_result = lo_tnb->gettranscriptionjob( iv_job_name ).  
    DATA(lo_job) = oo_result->get_transcriptionjob( ).  
    MESSAGE 'Retrieved transcription job details.' TYPE 'I'.  
    CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).  
    MESSAGE lo_bad_request_ex TYPE 'I'.  
    RAISE EXCEPTION lo_bad_request_ex.  
    CATCH /aws1/cx_tnbnotfoundexception INTO DATA(lo_not_found_ex).  
    MESSAGE lo_not_found_ex TYPE 'I'.  
    RAISE EXCEPTION lo_not_found_ex.  
    CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).  
    MESSAGE lo_internal_ex TYPE 'I'.  
    RAISE EXCEPTION lo_internal_ex.  
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 [GetTranscriptionJob](#)。

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。

## GetVocabulary 搭配 AWS SDK 或 CLI 使用

下列程式碼範例示範如何使用 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

### 適用於 .NET 的 SDK

#### Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
/// <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《適用於 .NET 的 AWS SDK API 參考》中的 [GetVocabulary](#)。

## 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》中的[GetVocabulary](#)。

## Python

### 適用於 Python 的 SDK (Boto3)

#### Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在[AWS 程式碼範例儲存庫](#)中設定和執行。

```
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《適用於 Python (Boto3) 的 AWS SDK API 參考資料》中的 [GetVocabulary](#)。

## SAP ABAP

### 適用於 SAP ABAP 的開發套件

#### Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
TRY.  
    oo_result = lo_tnb->getvocabulary( iv_vocabulary_name ).  
    MESSAGE 'Retrieved vocabulary details.' TYPE 'I'.  
CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).  
    MESSAGE lo_bad_request_ex TYPE 'I'.  
    RAISE EXCEPTION lo_bad_request_ex.  
CATCH /aws1/cx_tnbnotfoundexception INTO DATA(lo_not_found_ex).  
    MESSAGE lo_not_found_ex TYPE 'I'.  
    RAISE EXCEPTION lo_not_found_ex.  
CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).  
    MESSAGE lo_internal_ex TYPE 'I'.  
    RAISE EXCEPTION lo_internal_ex.  
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 [GetVocabulary](#)。

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。

## ListMedicalTranscriptionJobs 搭配 AWS SDK 或 CLI 使用

下列程式碼範例示範如何使用 ListMedicalTranscriptionJobs。

.NET

適用於 .NET 的 SDK

### Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
/// <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《適用於 .NET 的 AWS SDK API 參考》中的 [ListMedicalTranscriptionJobs](#)。

## CLI

## AWS CLI

## 列出醫學轉錄作業

下列 `list-medical-transcription-jobs` 範例列出與 AWS 您的帳戶和區域相關聯的醫療轉錄任務。若要取得有關特定轉錄作業的詳細資訊，請複製轉錄輸出中 `MedicalTranscriptionJobName` 參數的值，然後指定 `get-medical-transcription-job` 命令的 `MedicalTranscriptionJobName` 選項值。若要進一步查看轉錄作業，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
+LaDtZPC4u6ttoHLRlEefzqstHXSgapXg3tEBtm9piIaPB6M0M5BB6t86+qtmocTR/
qrteHZBBudhTfbCwhsxaqujHiiUvFdm3BQbKKWIW06yV9b+4f38oD2lVIan
+vfUs3gBYA15VTDmXXzQPBOHPjtwmFI+IWX15nSUjWuN3TUy1HgPWzDaYT8qBtu0Z+3UG4V6b
+K2CC0XszXg5rBq9hYgNzy4XoFh/6s5DoSznzq49Q9xHgHdT2yBADFmvFK7myZBsJ75+2vQZ0SVpWUPy3WT/32zFAC
+mFYfUjtTZ8n/jq7aQEjQ42A
+X/7K6Jg0cdVPtEg8P1Dr5kgYYG3q30mYXX37U3FZuJmnTI63VtIXsNn0U5eGoY0btpk00Nq9UkzGsjxqj84ZD5n
+S0EGy9ZUYBJRRcGeYUM3Q4DbSjFuoSAqcFdLIWZdp8qIREMQIBWy7BLwSdyqsQo2vRrd53hm5aWM7SVf6pPq6X/
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》中的 [ListMedicalTranscriptionJobs](#)。

## JavaScript

適用於 JavaScript (v3) 的 SDK

### Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

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 [《適用於 JavaScript 的 AWS SDK 開發人員指南》](#)。
- 如需 API 詳細資訊，請參閱 [《適用於 JavaScript 的 AWS SDK API 參考》](#) 中的 [ListMedicalTranscriptionJobs](#)。

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。

## ListTranscriptionJobs 搭配 AWS SDK 或 CLI 使用

下列程式碼範例示範如何使用 ListTranscriptionJobs。

### .NET

適用於 .NET 的 SDK

#### Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
/// <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《適用於 .NET 的 AWS SDK API 參考》中的 [ListTranscriptionJobs](#)。

## CLI

## AWS CLI

## 列出您的轉錄作業

下列 `list-transcription-jobs` 範例列出與 AWS 您的帳戶和區域相關聯的轉錄任務。

```
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》中的 [ListTranscriptionJobs](#)。

## Java

### SDK for Java 2.x

#### Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```

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 的詳細資訊，請參閱《AWS SDK for Java 2.x API 參考資料》中的 [ListTranscriptionJobs](#)。

## JavaScript

### 適用於 JavaScript (v3) 的 SDK

#### Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

#### 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《[適用於 JavaScript 的 AWS SDK 開發人員指南](#)》。
- 如需 API 的詳細資訊，請參閱《[適用於 JavaScript 的 AWS SDK API 參考資料](#)》中的 [ListTranscriptionJobs](#)。

## Python

### 適用於 Python (Boto3) 的 SDK

#### Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
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《適用於 Python (Boto3) 的 AWS SDK API 參考資料》中的 [ListTranscriptionJobs](#)。

## SAP ABAP

### 適用於 SAP ABAP 的開發套件

#### Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
TRY.
  IF iv_job_filter IS NOT INITIAL.
    oo_result = lo_tnb->listtranscriptionjobs( iv_jobnamecontains =
iv_job_filter ).
  ELSE.
    oo_result = lo_tnb->listtranscriptionjobs( ).
  ENDIF.
  MESSAGE 'Retrieved transcription jobs list.' TYPE 'I'.
CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
  MESSAGE lo_bad_request_ex TYPE 'I'.
  RAISE EXCEPTION lo_bad_request_ex.
CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).
  MESSAGE lo_internal_ex TYPE 'I'.
  RAISE EXCEPTION lo_internal_ex.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 [ListTranscriptionJobs](#)。

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。

## ListVocabularies 搭配 AWS SDK 或 CLI 使用

下列程式碼範例示範如何使用 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

適用於 .NET 的 SDK

#### Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
/// <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《適用於 .NET 的 AWS SDK API 參考》中的 [ListVocabularies](#)。

## CLI

### AWS CLI

#### 列出您的自訂詞彙

下列 `list-vocabularies` 範例列出與 AWS 您的帳戶和區域相關聯的自訂詞彙。

```
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》中的 [ListVocabularies](#)。

## Python

### 適用於 Python 的 SDK (Boto3)

#### Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
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《適用於 Python (Boto3) 的 AWS SDK API 參考資料》中的 [ListVocabularies](#)。

## SAP ABAP

### 適用於 SAP ABAP 的開發套件

#### Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```

TRY.
  IF iv_vocab_filter IS NOT INITIAL.
    oo_result = lo_tnb->listvocabularies( iv_namecontains =
iv_vocab_filter ).
  ELSE.
    oo_result = lo_tnb->listvocabularies( ).
  ENDIF.
  MESSAGE 'Retrieved vocabularies list.' TYPE 'I'.

```

```
CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
  MESSAGE lo_bad_request_ex TYPE 'I'.
  RAISE EXCEPTION lo_bad_request_ex.
CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).
  MESSAGE lo_internal_ex TYPE 'I'.
  RAISE EXCEPTION lo_internal_ex.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 [ListVocabularies](#)。

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。

## StartMedicalTranscriptionJob 搭配 AWS SDK 或 CLI 使用

下列程式碼範例示範如何使用 StartMedicalTranscriptionJob。

.NET

適用於 .NET 的 SDK

### Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
/// <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《適用於 .NET 的 AWS SDK API 參考》中的 [StartMedicalTranscriptionJob](#)。

## 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": "amzn-s3-demo-bucket",  
  "Media": {  
    "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"  
  }  
}
```

輸出：

```
{  
  "MedicalTranscriptionJob": {  
    "MedicalTranscriptionJobName": "simple-dictation-medical-transcription-  
job",  
    "TranscriptionJobStatus": "IN_PROGRESS",  
    "LanguageCode": "language-code",  
    "Media": {  
      "MediaFileUri": "s3://amzn-s3-demo-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": "amzn-s3-demo-bucket",  
  "Media": {  
    "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"  
  }  
}
```

輸出：

```
{  
  "MedicalTranscriptionJob": {  
    "MedicalTranscriptionJobName": "simple-conversation-medical-  
transcription-job",  
    "TranscriptionJobStatus": "IN_PROGRESS",  
    "LanguageCode": "language-code",  
    "Media": {  
      "MediaFileUri": "s3://amzn-s3-demo-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": "amzn-s3-demo-bucket",  
  "Media": {  
    "MediaFileUri": "s3://amzn-s3-demo-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://amzn-s3-demo-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": "amzn-s3-demo-bucket",  
  "Media": {  
    "MediaFileUri": "s3://amzn-s3-demo-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://amzn-s3-demo-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": "amzn-s3-demo-bucket",
  "Media": {
    "MediaFileUri": "s3://amzn-s3-demo-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://amzn-s3-demo-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": "amzn-s3-demo-bucket",
"Media": {
  "MediaFileUri": "s3://amzn-s3-demo-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://amzn-s3-demo-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": "amzn-s3-demo-bucket",
  "Media": {
    "MediaFileUri": "s3://amzn-s3-demo-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://amzn-s3-demo-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》中的 [StartMedicalTranscriptionJob](#)。

## JavaScript

### 適用於 JavaScript (v3) 的 SDK

#### Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

#### 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 [《適用於 JavaScript 的 AWS SDK 開發人員指南》](#)。
- 如需 API 詳細資訊，請參閱 [《適用於 JavaScript 的 AWS SDK API 參考》](#) 中的 [StartMedicalTranscriptionJob](#)。

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。

## StartTranscriptionJob 搭配 AWS SDK 或 CLI 使用

下列程式碼範例示範如何使用 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

### 適用於 .NET 的 SDK

#### Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
/// <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《適用於 .NET 的 AWS SDK API 參考》中的 [StartTranscriptionJob](#)。

## 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://amzn-s3-demo-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://amzn-s3-demo-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://amzn-s3-demo-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://amzn-s3-demo-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://amzn-s3-demo-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://amzn-s3-demo-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://amzn-s3-demo-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://amzn-s3-demo-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://amzn-s3-demo-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://amzn-s3-demo-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://amzn-s3-demo-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://amzn-s3-demo-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://amzn-s3-demo-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://amzn-s3-demo-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》中的 [StartTranscriptionJob](#)。

## JavaScript

### 適用於 JavaScript (v3) 的 SDK

#### Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

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《[適用於 JavaScript 的 AWS SDK 開發人員指南](#)》。
- 如需 API 的詳細資訊，請參閱《[適用於 JavaScript 的 AWS SDK API 參考資料](#)》中的 [StartTranscriptionJob](#)。

## Python

### 適用於 Python (Boto3) 的 SDK

#### Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
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《適用於 Python (Boto3) 的 AWS SDK API 參考資料》中的 [StartTranscriptionJob](#)。

## SAP ABAP

## 適用於 SAP ABAP 的開發套件

 Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
TRY.
  DATA(lo_media) = NEW /aws1/cl_tnbmedia( iv_mediafileuri = iv_media_uri ).
  DATA(lo_settings) = NEW /aws1/cl_tnbsettings( ).
  IF iv_vocabulary_name IS NOT INITIAL.
    lo_settings = NEW /aws1/cl_tnbsettings( iv_vocabularyname =
iv_vocabulary_name ).
  ENDIF.

  oo_result = lo_tnb->starttranscriptionjob(
    iv_transcriptionjobname = iv_job_name
    io_media = lo_media
    iv_mediaformat = iv_media_format
    iv_languagecode = iv_language_code
    io_settings = lo_settings ).

  MESSAGE 'Transcription job started.' TYPE 'I'.
  CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
  MESSAGE lo_bad_request_ex TYPE 'I'.
  RAISE EXCEPTION lo_bad_request_ex.
  CATCH /aws1/cx_tnblimitexceededex INTO DATA(lo_limit_ex).
  MESSAGE lo_limit_ex TYPE 'I'.
  RAISE EXCEPTION lo_limit_ex.
  CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).
  MESSAGE lo_internal_ex TYPE 'I'.
  RAISE EXCEPTION lo_internal_ex.
  CATCH /aws1/cx_tnbconflictexception INTO DATA(lo_conflict_ex).
  MESSAGE lo_conflict_ex TYPE 'I'.
  RAISE EXCEPTION lo_conflict_ex.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 [StartTranscriptionJob](#)。

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。

## UpdateVocabulary 搭配 AWS SDK 或 CLI 使用

下列程式碼範例示範如何使用 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

適用於 .NET 的 SDK

### Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
/// <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《適用於 .NET 的 AWS SDK API 參考》中的 [UpdateVocabulary](#)。

## 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://amzn-s3-demo-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》中的 [UpdateVocabulary](#)。

## Python

### 適用於 Python 的 SDK (Boto3)

#### Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
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《適用於 Python (Boto3) 的 AWS SDK API 參考資料》中的 [UpdateVocabulary](#)。

## SAP ABAP

### 適用於 SAP ABAP 的開發套件

#### Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
TRY.
  IF it_phrases IS NOT INITIAL.
    oo_result = lo_tnb->updatevocabulary(
      iv_vocabularyname = iv_vocabulary_name
      iv_languagecode = iv_language_code
      it_phrases = it_phrases ).
  ELSEIF iv_vocab_file_uri IS NOT INITIAL.
    oo_result = lo_tnb->updatevocabulary(
      iv_vocabularyname = iv_vocabulary_name
      iv_languagecode = iv_language_code
      iv_vocabularyfileuri = iv_vocab_file_uri ).
  ENDIF.
  MESSAGE 'Vocabulary updated.' TYPE 'I'.
CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
  MESSAGE lo_bad_request_ex TYPE 'I'.
CATCH /aws1/cx_tnblimitexceededex INTO DATA(lo_limit_ex).
  MESSAGE lo_limit_ex TYPE 'I'.
  RAISE EXCEPTION lo_limit_ex.
CATCH /aws1/cx_tnbnotfoundexception INTO DATA(lo_not_found_ex).
  MESSAGE lo_not_found_ex TYPE 'I'.
CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).
  MESSAGE lo_internal_ex TYPE 'I'.
  RAISE EXCEPTION lo_internal_ex.
CATCH /aws1/cx_tnbconflictexception INTO DATA(lo_conflict_ex).
  MESSAGE lo_conflict_ex TYPE 'I'.
  RAISE EXCEPTION lo_conflict_ex.
```

```
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 [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。

## 使用 AWS SDKs Amazon Transcribe 案例

下列程式碼範例示範如何在 Amazon Transcribe AWS SDKs 中實作常見案例。這些案例示範如何呼叫 Amazon Transcribe 中的多個函數，或與其他 AWS 服務結合，藉以完成特定任務。每個案例均包含完整原始碼的連結，您可在連結中找到如何設定和執行程式碼的相關指示。

案例的目標是獲得中等水平的經驗，協助您了解內容中的服務動作。

### 範例

- [建置 Amazon Transcribe 串流應用程式](#)
- [使用 AWS SDK 將文字轉換為語音並返回文字](#)
- [使用 SDK 建立和精簡 Amazon Transcribe 自訂詞彙 AWS](#)
- [使用 AWS SDK 透過 Amazon Transcribe 轉錄音訊並取得任務資料](#)

## 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 SDK

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。

如需完整的原始碼和如何設定及執行的指示，請參閱 [GitHub](#) 上的完整範例。

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 將文字轉換為語音並返回文字

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。

如需完整的原始碼和如何設定及執行的指示，請參閱 [GitHub](#) 上的完整範例。

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。

## 使用 SDK 建立和精簡 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 的 SDK (Boto3)

#### Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

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。

## 使用 AWS SDK 透過 Amazon Transcribe 轉錄音訊並取得任務資料

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

#### SDK for Java 2.x

##### Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

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);
        })
        .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 static Subscription currentSubscription;
    private final InputStream inputStream;

    private AudioStreamPublisher(InputStream inputStream) {
```

```
        this.inputStream = inputStream;
    }

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

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

public static class SubscriptionImpl implements Subscription {
    private static final int CHUNK_SIZE_IN_BYTES = 1024;
    private final Subscriber<? super AudioStream> subscriber;
    private final InputStream inputStream;
    private final ExecutorService executor = Executors.newFixedThreadPool(1);
    private final 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 的 SDK (Boto3)

#### Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
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:  
        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")
    file_uri = "s3://test-transcribe/answer2.wav"
    transcribe_file("Example-job", file_uri, transcribe_client)

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

- 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。
  - [GetTranscriptionJob](#)
  - [StartTranscriptionJob](#)

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。

## 中的安全性 Amazon Transcribe

的雲端安全 AWS 是最高優先順序。身為 AWS 客戶，您可以受益於資料中心和網路架構，其建置旨在滿足最安全敏感組織的需求。

安全是 AWS 和 之間的共同責任。[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：

- 雲端的安全性：AWS 負責保護在 中執行 AWS 服務的基礎設施 AWS 雲端。AWS 也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的有效性，做為[AWS 合規計畫](#)的一部分。若要了解適用的合規計劃 Amazon Transcribe，請參閱[AWS 合規計劃範圍內的服務](#)。
- 雲端內部的安全：您的責任須視所使用的 AWS 服務而定。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。下列主題說明如何設定 Amazon Transcribe 以符合您的安全和合規目標。您也會了解如何使用其他服務 AWS 來監控和保護您的 Amazon Transcribe 資源。

### 主題

- [的 Identity and Access Management Amazon Transcribe](#)
- [中的資料保護 Amazon Transcribe](#)
- [監控 Amazon Transcribe](#)
- [的合規驗證 Amazon Transcribe](#)
- [中的彈性 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 Amazon Transcribe](#)

## 的 Identity and Access Management Amazon Transcribe

AWS Identity and Access Management (IAM) 是一種 AWS 服務，可協助管理員安全地控制對 AWS 資源的存取。IAM 管理員可控制誰可以驗證（登入）和授權（具有許可）來使用 Amazon Transcribe 資源。IAM 是您可以免費使用 AWS 服務的。

### 主題

- [目標對象](#)
- [使用身分驗證](#)
- [使用政策管理存取權](#)
- [Amazon Transcribe 如何使用 IAM](#)
- [預防跨服務混淆代理人](#)
- [Amazon Transcribe 身分型政策範例](#)
- [對 Amazon Transcribe 身分和存取進行故障診斷](#)

## 目標對象

使用方式 AWS Identity and Access Management (IAM) 會根據您的角色而有所不同：

- 服務使用者 — 若無法存取某些功能，請向管理員申請所需許可 (請參閱 [對 Amazon Transcribe 身分和存取進行故障診斷](#))
- 服務管理員 — 負責設定使用者存取權並提交相關許可請求 (請參閱 [Amazon Transcribe 如何使用 IAM](#))
- IAM 管理員 — 撰寫政策以管理存取控制 (請參閱 [Amazon Transcribe 身分型政策範例](#))

## 使用身分驗證

身分驗證是您 AWS 使用身分憑證登入的方式。您必須以 AWS 帳戶根使用者、IAM 使用者或擔任 IAM 角色進行身分驗證。

您可以使用身分來源的登入資料，例如 AWS IAM Identity Center (IAM Identity Center)、單一登入身分驗證或 Google/Facebook 登入資料，以聯合身分的形式登入。如需有關登入的詳細資訊，請參閱《AWS 登入 使用者指南》中的[如何登入您的 AWS 帳戶](#)。

對於程式設計存取，AWS 提供 SDK 和 CLI 以密碼編譯方式簽署請求。如需詳細資訊，請參閱《IAM 使用者指南》中的[API 請求的 AWS 第 4 版簽署程序](#)。

## AWS 帳戶 根使用者

當您建立時 AWS 帳戶，您會從一個名為 AWS 帳戶 theroot 使用者的登入身分開始，該身分具有對所有 AWS 服務和資源的完整存取權。強烈建議不要使用根使用者來執行日常任務。有關需要根使用者憑證的任務，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

最佳實務是要求人類使用者使用聯合身分提供者，以 AWS 服務 使用臨時憑證存取。

聯合身分是您企業目錄、Web 身分提供者的使用者，或使用身分來源的 AWS 服務 憑證存取 Directory Service。聯合身分會擔任角色，而該角色會提供臨時憑證。

若需集中化管理存取權限，建議使用 AWS IAM Identity Center。如需詳細資訊，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 [https://docs.aws.amazon.com/IAM/latest/UserGuide/id\\_users.html](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_users.html) 是一種身分具備單人或應用程式的特定許可權。建議以臨時憑證取代具備長期憑證的 IAM 使用者。如需詳細資訊，請參閱《IAM 使用者指南》中的 [要求人類使用者使用聯合身分提供者來 AWS 使用臨時憑證存取](#)。

[IAM 群組](#) 會指定 IAM 使用者集合，使管理大量使用者的許可權更加輕鬆。如需詳細資訊，請參閱《IAM 使用者指南》中的 [IAM 使用者的使用案例](#)。

## IAM 角色

IAM 角色 [https://docs.aws.amazon.com/IAM/latest/UserGuide/id\\_roles.html](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html) 的身分具有特定許可權，其可以提供臨時憑證。您可以透過 [從使用者切換到 IAM 角色（主控台）](#) 或呼叫 AWS CLI 或 AWS API 操作來擔任角色。如需詳細資訊，請參閱《IAM 使用者指南》中的 [擔任角色的方法](#)。

IAM 角色適用於聯合身分使用者存取、臨時 IAM 使用者許可、跨帳戶存取權與跨服務存取，以及在 Amazon EC2 執行的應用程式。如需詳細資訊，請參閱《IAM 使用者指南》中的 [IAM 中的快帳戶資源存取](#)。

## 使用政策管理存取權

您可以透過建立政策並將其連接到身分或資源 AWS 來控制 AWS 中的存取。政策定義與身分或資源相關聯的許可。當委託人提出請求時 AWS，會評估這些政策。大多數政策會以 JSON 文件 AWS 形式存放在中。如需進一步了解 JSON 政策文件，請參閱《IAM 使用者指南》中的 [JSON 政策概觀](#)。

管理員會使用政策，透過定義哪些主體可在哪些條件下對哪些資源執行動作，以指定可存取的範圍。

預設情況下，使用者和角色沒有許可。IAM 管理員會建立 IAM 政策並將其新增至角色，供使用者後續擔任。IAM 政策定義動作的許可，無論採用何種方式執行。

## 身分型政策

身分型政策是附加至身分 (使用者、使用者群組或角色) 的 JSON 許可政策文件。這類政策控制身分可對哪些資源執行哪些動作，以及適用的條件。如需了解如何建立身分型政策，請參閱《IAM 使用者指南》中的[透過客戶管理政策定義自訂 IAM 許可](#)。

身分型政策可分為內嵌政策 (直接內嵌於單一身分) 與受管政策 (可附加至多個身分的獨立政策)。如需了解如何在受管政策及內嵌政策之間做選擇，請參閱《IAM 使用者指南》中的[在受管政策與內嵌政策之間選擇](#)。

## 資源型政策

資源型政策是附加到資源的 JSON 政策文件。範例包括 IAM 角色信任政策與 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。您必須在資源型政策中[指定主體](#)。

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來自 IAM 的 AWS 受管政策。

## 其他政策類型

AWS 支援其他政策類型，可設定更多常見政策類型授予的最大許可：

- 許可界限 — 設定身分型政策可授與 IAM 實體的最大許可。如需詳細資訊，請參閱《IAM 使用者指南》中的[IAM 實體許可界限](#)。
- 服務控制政策 (SCP) — 為 AWS Organizations 中的組織或組織單位指定最大許可。如需詳細資訊，請參閱《AWS Organizations 使用者指南》中的[服務控制政策](#)。
- 資源控制政策 (RCP) — 設定您帳戶中資源可用許可的上限。如需詳細資訊，請參閱《AWS Organizations 使用者指南》中的[資源控制政策 \(RCP\)](#)。
- 工作階段政策 — 在以程式設計方式為角色或聯合身分使用者建立臨時工作階段時，以參數形式傳遞的進階政策。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

當多種類型的政策適用於請求時，產生的許可會更複雜而無法理解。若要了解如何 AWS 在涉及多個政策類型時決定是否允許請求，請參閱《IAM 使用者指南》中的[政策評估邏輯](#)。

## Amazon Transcribe 如何使用 IAM

在您使用 IAM 管理對的存取之前 Amazon Transcribe，請先了解哪些 IAM 功能可與 搭配使用 Amazon Transcribe。

IAM 您可以搭配 使用的功能 Amazon Transcribe

IAM 功能	Amazon Transcribe 支援
<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

若要全面了解 Amazon Transcribe 和其他 AWS 服務如何與大多數 IAM 功能搭配使用，請參閱IAM 《使用者指南》中的 [AWS 服務 IAM](#)。

### 的身分型政策 Amazon Transcribe

支援身分型政策：是

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。如需了解如何建立身分型政策，請參閱《IAM 使用者指南》中的 [透過客戶管理政策定義自訂 IAM 許可](#)。

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。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](#)。

## 的身分型政策範例 Amazon Transcribe

若要檢視 Amazon Transcribe 身分型政策的範例，請參閱 [Amazon Transcribe 身分型政策範例](#)。

## 中的資源型政策 Amazon Transcribe

支援資源型政策：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](#)。委託人可以包含帳戶、使用者、角色、聯合身分使用者或 AWS 服務。

如需啟用跨帳戶存取權，您可以在其他帳戶內指定所有帳戶或 IAM 實體作為資源型政策的主體。如需詳細資訊，請參閱《IAM 使用者指南》中的 [IAM 中的快帳戶資源存取](#)。

## 的政策動作 Amazon Transcribe

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。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》中的 [定義的動作 Amazon Transcribe](#)。

中的政策動作在動作之前 Amazon Transcribe 使用 transcribe 字首。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 Amazon Transcribe 身分型政策的範例，請參閱 [Amazon Transcribe 身分型政策範例](#)。

## 的政策資源 Amazon Transcribe

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。最佳實務是使用其 [Amazon Resource Name \(ARN\)](#) 來指定資源。若動作不支援資源層級許可，使用萬用字元 (\*) 表示該陳述式適用於所有資源。

```
"Resource": "*"
```

若要查看 Amazon Transcribe 資源類型及其 ARNs，請參閱《服務授權參考》中的 [定義的資源 Amazon Transcribe](#)。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 Amazon Transcribe 身分型政策的範例，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 元素會根據定義的條件，指定陳述式的執行時機。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。若要查看所有 AWS 全域條件索引鍵，請參閱《IAM 使用者指南》中的 [AWS 全域條件內容索引鍵](#)。

若要查看 Amazon Transcribe 條件索引鍵的清單，請參閱《服務授權參考》中的 [的條件索引鍵 Amazon Transcribe](#)。若要了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 Amazon Transcribe 身分型政策的範例，請參閱 [Amazon Transcribe 身分型政策範例](#)。

## 中的 ACLs 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。

## ABAC 搭配 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，根據稱為標籤的屬性定義許可權。您可以將標籤連接至 IAM 實體 AWS 和資源，然後設計 ABAC 政策，以便在主體的標籤符合資源上的標籤時允許操作。

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 [使用 ABAC 授權定義許可](#)。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 資源的詳細資訊，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](#)。

## 搭配 使用暫時登入資料 Amazon Transcribe

支援臨時憑證：是

臨時登入資料提供 AWS 資源的短期存取權，當您使用聯合或切換角色時，會自動建立。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 使用者指南》中的 [IAM 中的臨時安全憑證與可與 IAM 搭配運作的 AWS 服務](#)。

## 的跨服務主體許可 Amazon Transcribe

支援轉寄存取工作階段 (FAS)：是

轉送存取工作階段 (FAS) 使用呼叫的委託人許可 AWS 服務，並結合 AWS 服務請求向下游服務提出請求。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](#)。

### Warning

變更服務角色的許可可能會中斷 Amazon Transcribe 功能。只有在 Amazon Transcribe 提供指引時，才能編輯服務角色。

## 的服務連結角色 Amazon Transcribe

支援服務連結角色：否

服務連結角色是連結至的一種服務角色 AWS 服務。服務可以擔任代表您執行動作的角色。服務連結角色會出現在您的 [AWS 帳戶](#)，並由服務擁有。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。

如需為其他服務建立或管理服務連結角色的詳細資訊，請參閱 [AWS 使用的服務 IAM](#)。在資料表中尋找服務，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

混淆代理人是由不同實體強制執行動作的實體 (服務或帳戶)。這種類型的冒充可能發生跨帳戶和跨服務。

為了防止混淆代理人，AWS 提供工具，協助您使用已授予存取您 [中資源權限的服務主體](#)，保護所有服務的資料 AWS 帳戶。本節著重於跨服務混淆代理人預防 Amazon Transcribe；不過，您可以在 IAM 使用者指南的[混淆代理人問題](#)區段中進一步了解此主題。

若要限制 IAM 授予 Amazon Transcribe 存取資源的許可，建議您在資源政策 [aws:SourceAccount](#) 中使用全域條件內容金鑰 [aws:SourceArn](#) 和 [aws:SourceAccount](#)。

如果您使用這兩個全域條件內容索引鍵，而 `aws:SourceArn` 值包含 AWS 帳戶 ID，則 `aws:SourceAccount` 中的值在相同政策陳述式中使用時，AWS 帳戶 `aws:SourceArn` 必須使用相同的 AWS 帳戶 ID。

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 中的任何資源 AWS 帳戶 與跨服務存取建立關聯，請使用 `aws:SourceAccount`。

### Note

範混淆代理人問題的最有效方法是使用 `aws:SourceArn` 全域條件內容金鑰，以及資源的完整 ARN。如果不知道資源的完整 ARN，或者如果您指定多個資源，請使用 `aws:SourceArn` 全域條件內容金鑰，同時使用 ARN 未知部分的萬用字元 (\*)。例如：`arn:aws:transcribe::123456789012:*`。

如需假設角色政策的範例，其中顯示如何避免混淆代理人的問題，請參閱 [預防混淆代理人政策](#)。

## 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。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。

如需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 \(主控台\)](#)。

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》中的 [適用於 Amazon Transcribe 的動作、資源和條件索引鍵](#)。

### 主題

- [政策最佳實務](#)
- [使用 AWS 管理主控台](#)
- [IAM 角色所需的許可](#)
- [Amazon S3 加密金鑰所需的許可](#)
- [允許使用者檢視他們自己的許可](#)
- [AWS KMS 加密內容政策](#)
- [預防混淆代理人政策](#)
- [根據標籤檢視轉錄作業](#)

## 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 受管政策並邁向最低權限許可 – 若要開始將許可授予您的使用者和工作負載，請使用將許可授予許多常見使用案例的 AWS 受管政策。它們可在您的 中使用 AWS 帳戶。我們建議您定義特定於使用案例 AWS 的客戶受管政策，以進一步減少許可。如需更多資訊，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 套用許可的更多相關資訊，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。如果透過特定 等使用服務動作 AWS 服務，您也可以使用條件來授予其存取權 CloudFormation。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，可協助您撰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 [使用 IAM Access Analyzer 驗證政策](#)。
- 需要多重要素驗證 (MFA) – 如果您的案例需要 IAM 使用者或 中的根使用者 AWS 帳戶，請開啟 MFA 以提高安全性。如需在呼叫 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 [透過 MFA 的安全 API 存取](#)。

如需 IAM 中最佳實務的相關資訊，請參閱《IAM 使用者指南》中的 [IAM 安全最佳實務](#)。

## 使用 AWS 管理主控台

若要存取 Amazon Transcribe 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 中 Amazon Transcribe 資源的詳細資訊 AWS 帳戶。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，您不需要允許最低主控台許可。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。

為了確保實體 (使用者和角色) 可以使用 [AWS 管理主控台](#)，請將下列其中一個 AWS 受管政策連接到它們。

- `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。它也允許存取儲存貯體名稱包含 `transcribe` 的 Amazon S3 儲存貯體。
- `AmazonTranscribeReadOnlyAccess`：授與 Amazon Transcribe 資源唯讀存取權限，讓您可以取得並列出轉錄作業和自訂詞彙。

#### Note

您可以檢閱受管授權政策，只要登入 IAM AWS 管理主控台，並使用政策名稱進行搜尋。搜尋「轉錄」傳回上述列出的兩個政策 (`AmazonTranscribeReadOnly` 和 `AmazonTranscribeFullAccess`)。

您也可以建立自己的自訂 IAM 政策，以允許 Amazon Transcribe API 動作的許可。您可以將這些自訂政策連接至需要指定授權的實體。

## IAM 角色所需的許可

如果您建立要呼叫 IAM 的角色 Amazon Transcribe，它必須具有存取儲存 Amazon S3 貯體的許可。如果適用，也 KMS key 必須使用以加密儲存貯體的內容。請參考下列章節的範例政策。

### 信任政策

您用來提出轉錄請求的 IAM 實體必須具有信任政策，讓 Amazon Transcribe 能夠擔任該角色。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」。

的信任政策 Amazon Transcribe

### JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "transcribe.amazonaws.com"
        ]
      }
    }
  ],
}
```

```

    "Action": [
      "sts:AssumeRole"
    ],
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "111122223333"
      },
      "ArnLike": {
        "aws:SourceArn": "arn:aws:transcribe:us-west-2:111122223333:*"
      }
    }
  }
}

```

## 即時通話分析的信任政策

### JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "transcribe.streaming.amazonaws.com"
        ]
      },
      "Action": [
        "sts:AssumeRole"
      ],
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "111122223333"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:transcribe:us-west-2:111122223333:*"
        }
      }
    }
  ]
}

```

```
}
```

## Amazon S3 輸入儲存貯體政策

下列政策提供 IAM 角色從指定輸入儲存貯體存取檔案的許可。

### JSON

```
{
  "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 輸出儲存貯體政策

下列政策提供 IAM 角色將檔案寫入指定輸出儲存貯體的許可。

### JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::DOC-EXAMPLE-OUTPUT-BUCKET/*"
      ]
    }
  ]
}
```

```
}
```

## Amazon S3 加密金鑰所需的許可

如果您使用 KMS key 來加密 儲存 Amazon S3 貯體，請在 KMS key 政策中包含下列項目。這可讓您 Amazon Transcribe 存取儲存貯體的內容。如需允許存取的詳細資訊 KMS keys，請參閱《AWS KMS 開發人員指南》中的[允許外部 AWS 帳戶 存取 KMS key](#)。

### JSON

```
{
  "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 使用者檢視附加到他們使用者身分的內嵌及受管政策。此政策包含在主控制台或使用 或 AWS CLI AWS API 以程式設計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

下列政策會授予 IAM 角色「ExampleRole」許可，以針對此特定使用 AWS KMS 解密和加密操作 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](#)。

### JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
      },
      "Action": [
        "kms:Decrypt",

```

```

        "kms:Encrypt",
        "kms:GenerateDataKey*",
        "kms:ReEncrypt*"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "kms:EncryptionContext:color": "indigoBlue"
        }
    }
},
{
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
    },
    "Action": "kms:DescribeKey",
    "Resource": "*"
}
]
}

```

## 預防混淆代理人政策

以下是擔任角色政策的範例，示範如何使用 `aws:SourceArn` 和 `aws:SourceAccount` Amazon Transcribe 來防止混淆代理人問題。如需防止混淆代理人的詳細資訊，請參閱 [預防跨服務混淆代理人](#)。

### JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "transcribe.amazonaws.com"
      },
      "Action": [
        "sts:AssumeRole"
      ],
    }
  ],
}

```

```

        "Condition": {
            "StringEquals": {
                "aws:SourceAccount": "111122223333"
            },
            "ArnLike": {
                "aws:SourceArn": "arn:aws:transcribe:us-
west-2:111122223333:*"
            }
        }
    }
]
}

```

## 根據標籤檢視轉錄作業

您可以在身分型政策中使用條件，以根據標籤控制存取 Amazon Transcribe 資源。此範例會示範如何建立政策，允許檢視轉錄作業。但是，只有在轉錄作業標籤 Owner 的值是該使用者的使用者名稱時，才會授予該權限。此政策也會使用 AWS 管理主控台，授予完成此動作的必要權限。

您可以將此政策連接至您帳戶中的 IAM 實體。如果名為 test-role 的角色嘗試檢視轉錄作業，則轉錄作業必須加上標籤 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 [IAM JSON 政策元素：條件](#)。

如需在 中標記的詳細資訊 Amazon Transcribe，請參閱 [標記 資源](#)。

## JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ListTranscriptionJobsInConsole",
      "Effect": "Allow",
      "Action": "transcribe:ListTranscriptionJobs",
      "Resource": "*"
    },
    {
      "Sid": "ViewTranscriptionJobsIfOwner",
      "Effect": "Allow",
      "Action": "transcribe:GetTranscriptionJob",

```

```
        "Resource": "arn:aws:transcribe:us-west-2:111122223333:transcription-  
job/*",  
        "Condition": {  
            "StringEquals": {"aws:ResourceTag/Owner": "${aws:username}"}  
        }  
    }  
]  
}
```

## 對 Amazon Transcribe 身分和存取進行故障診斷

使用下列資訊來診斷和修正使用 Amazon Transcribe 和 AWS Identity and Access Management (IAM) 時可能遇到的常見問題。

### 主題

- [我無權在 中執行動作 Amazon Transcribe](#)
- [我未獲得執行 iam:PassRole 的授權](#)
- [我想要允許 以外的人員 AWS 帳戶 存取我的 Amazon Transcribe 資源](#)

### 我無權在 中執行動作 Amazon Transcribe

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 管理員。您的管理員提供您的簽署憑證。

### 我未獲得執行 iam:PassRole 的授權

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 可讓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 沒有將角色傳遞給服務的許可。

```
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 管理員。您的管理員提供您的簽署憑證。

## 我想要允許以外的人員 AWS 帳戶 存取我的 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：

- 若要了解 是否 Amazon Transcribe 支援這些功能，請參閱 [Amazon Transcribe 如何使用 IAM](#)。
- 若要了解如何 AWS 帳戶 在您擁有的 資源之間提供存取權，請參閱 [《IAM 使用者指南》中的在您擁有 AWS 帳戶 的另一個 IAM 使用者中提供存取權](#)。
- 若要了解如何將資源的存取權提供給第三方 AWS 帳戶，請參閱 [《IAM 使用者指南》中的將存取權提供給第三方 AWS 帳戶 擁有](#)。
- 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 \(聯合身分\)](#)。
- 如需了解使用角色和資源型政策進行跨帳戶存取之間的差異，請參閱 [《IAM 使用者指南》中的 IAM 中的跨帳戶資源存取](#)。

## 中的資料保護 Amazon Transcribe

AWS [共同責任模型](#) 適用於 中的資料保護 Amazon Transcribe。如此模型所述，AWS 負責保護執行所有的全域基礎設施 AWS 雲端。您負責維護在此基礎設施上託管內容的控制權。您也同時負責所使用 AWS 服務 的安全組態和管理任務。如需資料隱私權的詳細資訊，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，我們建議您保護 AWS 帳戶 登入資料，並使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 設定個別使用者。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 設定 API 和使用者活動記錄 AWS CloudTrail。如需有關使用 CloudTrail 追蹤擷取 AWS 活動的資訊，請參閱AWS CloudTrail 《使用者指南》中的[使用 CloudTrail 追蹤](#)。
- 使用 AWS 加密解決方案，以及其中的所有預設安全控制 AWS 服務。
- 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 AWS 透過命令列界面或 API 存取 時需要 FIPS 140-3 驗證的密碼編譯模組，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 \(FIPS\) 140-3](#)。

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，例如名稱欄位。這包括當您使用 Amazon Transcribe 或其他 AWS 服務 使用 主控台、API AWS CLI或 AWS SDKs時。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

的 Amazon Virtual Private Cloud (Amazon VPC) 端點 Amazon Transcribe 是 VPC 內的邏輯實體，僅允許連線 Amazon Transcribe。Amazon VPC 會將請求路由至 ， Amazon Transcribe 並將回應路由回 VPC。如需詳細資訊，請參閱 [AWS PrivateLink 概念](#)。如需搭配 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

資料加密有助於保護傳輸中與靜態的資料。您可以在傳輸期間使用 Amazon S3受管金鑰或KMS keys靜態金鑰，以及標準 Transport Layer Security (TLS) 來保護您的資料。

### 靜態加密

Amazon Transcribe使用預設Amazon S3金鑰 (SSE-S3) 對放在儲存Amazon S3貯體中的文字記錄進行伺服器端加密。

當您使用 [StartTranscriptionJob](#)操作時，您可以指定自己的 KMS key來加密轉錄任務的輸出。

Amazon Transcribe使用以預設金鑰加密的 Amazon EBS磁碟區。

## 傳輸中加密

Amazon Transcribe使用 TLS 1.2 搭配AWS憑證來加密傳輸中的資料。這包括串流轉錄。

## 金鑰管理

Amazon Transcribe使用 KMS keys為您的資料提供增強加密。使用 Amazon S3，您可以在建立轉錄任務時加密輸入媒體。與 整合AWS KMS可加密來自[StartTranscriptionJob](#)請求的輸出。

如果您未指定 KMS key，則會使用預設Amazon S3金鑰 (SSE-S3) 加密轉錄任務的輸出。

如需的詳細資訊AWS KMS，請參閱 [AWS Key Management Service開發人員指南](#)。

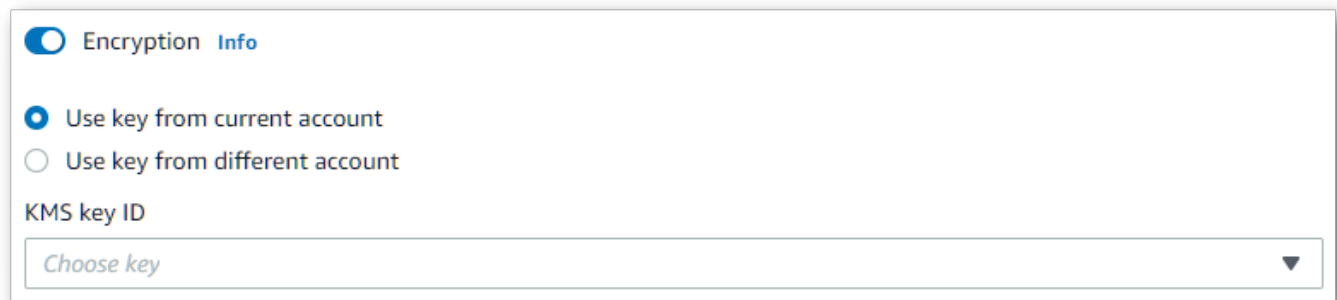
使用 進行金鑰管理AWS 管理主控台

若要加密轉錄任務的輸出，您可以選擇KMS key為提出請求AWS 帳戶的 使用，或從另一個 KMS key 使用 AWS 帳戶。

如果您未指定 KMS key，則會使用預設Amazon S3金鑰 (SSE-S3) 加密轉錄任務的輸出。

啟用輸出加密：

1. 在 輸出資料 下方，選擇 加密。



The screenshot shows a configuration panel for encryption. At the top, there is a toggle for 'Encryption Info' which is turned on. Below this are two radio button options: 'Use key from current account' (which is selected) and 'Use key from different account'. Underneath these options is a label 'KMS key ID' followed by a dropdown menu that currently displays 'Choose key'.

2. 選擇 KMS key是來自AWS 帳戶您目前使用的，還是來自不同的 AWS 帳戶。如果您想要使用來自目前的金鑰AWS 帳戶，請從 KMS keyID 選擇金鑰。如果您使用來自不同的金鑰AWS 帳戶，則必須輸入金鑰的 ARN。若要使用來自不同的金鑰AWS 帳戶，發起人必須具有的kms:Encrypt許可KMS key。請參閱[建立金鑰政策](#)，了解詳細資訊。

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 [StartCallAnalyticsJob](#)、[StartMedicalTranscriptionJob](#)或 [StartTranscriptionJob](#)操作的 `OutputEncryptionKMSKeyId` 參數來指定。

如果使用位於目前AWS 帳戶 中的金鑰，您可以使用四種方式KMS key之一來指定您的：

1. 使用 KMS keyID 本身。例如 1234abcd-12ab-34cd-56ef-1234567890ab。
2. 使用 KMS keyID 的別名。例如 alias/ExampleAlias。
3. 使用 KMS keyID 的 Amazon Resource Name (ARN)。例如 arn:aws:kms:region:account-ID:key/1234abcd-12ab-34cd-56ef-1234567890ab。
4. 使用KMS key別名的 ARN。例如 arn:aws:kms:region:account-ID:alias/ExampleAlias。

如果使用位於與目前 不同的AWS 帳戶 金鑰AWS 帳戶，您可以透過下列兩種方式KMS key之一來指定您的：

1. 使用 KMS keyID 的 ARN。例如 arn:aws:kms:region:account-ID:key/1234abcd-12ab-34cd-56ef-1234567890ab。
2. 使用KMS key別名的 ARN。例如 arn:aws:kms:region:account-ID:alias/ExampleAlias。

請注意，提出請求的實體必須取得許可才可使用指定的 KMS key。

## AWS KMS加密內容

AWS KMS加密內容是純文字、非私密金鑰：值對的映射。此映射代表額外的已驗證資料，稱為加密內容對，可為您的資料提供多一層的安全性。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。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。加密內容不是秘密，它在CloudTrail日誌中以純文字顯示（因此您可以使用它來識別和分類密碼編譯操作）。

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。

若要透過 API 使用輸出加密，設定 [StartTranscriptionJob](#) 操作中的 `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 KMS key ID。

您可以使用 [AWS KMS條件金鑰](#) 搭配 IAM 政策，KMS key 根據 [密碼編譯操作](#) 請求中使用的加密內容來控制對對稱加密的存取。如需加密內容政策的範例，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](#)。

## 選擇不使用您的資料以改善服務

根據預設，會 Amazon Transcribe 存放和使用其已處理的語音輸入，以開發服務並持續改善您的體驗。您可以使用 Amazon Transcribe AWS Organizations 選擇退出政策，選擇不讓內容用於開發和改進。如需如何選擇退出的詳細資訊，請參閱 [AI 服務退出政策](#)。

## 監控 Amazon Transcribe

監控是維護 Amazon Transcribe 和其他 AWS 解決方案的可靠性、可用性和效能的重要部分。AWS 提供下列監控工具，讓您監看 Amazon Transcribe、回報錯誤，並適時採取自動動作：

- Amazon CloudWatch AWS 會即時監控您的 AWS 資源和您在 上執行的應用程式。您可以收集和追蹤指標、建立自訂儀板表，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 用量或其他指標，並在需要時自動啟動新的執行個體。
- Amazon CloudWatch Logs 可以從 Amazon EC2 執行個體和其他來源監控 CloudTrail、存放和存取您的日誌檔案。CloudWatch Logs 可以監控日誌檔案中的資訊，並在達到特定閾值時通知您。您也可以將日誌資料存檔在高耐用性的儲存空間。
- AWS CloudTrail 會擷取由 發出或代表發出的 API 呼叫和相關事件，AWS 帳戶 並將日誌檔案交付至您指定的 儲存 Amazon S3 貯體。您可以識別呼叫的使用者和帳戶 AWS、進行呼叫的來源 IP 地址，以及呼叫的時間。

如需詳細資訊，請參閱「[Amazon CloudWatch 使用者指南](#)」。

Amazon EventBridge 是一種無伺服器服務，使用事件將應用程式元件連接在一起，讓您更輕鬆地建置可擴展的事件驅動型應用程式。可從您自己的應用程式、軟體即服務 (SaaS) 應用程式 AWS 和服務 EventBridge 提供即時資料串流，並將該資料路由到 等目標 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

您可以使用 監控 Amazon Transcribe CloudWatch，這會收集原始資料並將其處理為可讀且近乎即時的指標。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](#)」。

### 搭配 使用 Amazon CloudWatch 指標和維度 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。支援的指標透過 位於 AWS/Transcribe 命名空間 CloudWatch 中。

#### Note

CloudWatch 監控指標是免費的，且不計入 CloudWatch 服務配額。

如需 CloudWatch 指標的詳細資訊，請參閱[使用 Amazon CloudWatch 指標](#)。

## Amazon Transcribe 使用 監控 AWS CloudTrail

Amazon Transcribe 已與 服務整合 AWS CloudTrail，此服務提供 Amazon Transcribe 由 AWS Identity and Access Management (IAM) 使用者或角色，或由 AWS 服務在 中所採取動作的記錄。CloudTrail 擷取 的所有 API 呼叫 Amazon Transcribe。這包括來自 的呼叫，AWS 管理主控台 以及對 Amazon Transcribe APIs 程式碼呼叫，即事件。透過建立線索，您可以將 CloudTrail 事件持續交付至 Amazon S3 儲存貯體 Amazon Transcribe，包括 的事件。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 管理主控台 中的最新事件。使用 所收集的資訊 CloudTrail，您可以查看所提出的每個請求 Amazon Transcribe、提出請求的 IP 地址、提出請求的人員、提出請求的時間，以及其他詳細資訊。

若要進一步了解 CloudTrail，請參閱 [AWS CloudTrail 使用者指南](#)。

## Amazon Transcribe 而且 CloudTrail

CloudTrail 當您建立帳戶 AWS 帳戶 時，會在上啟用。當活動在 中發生時 Amazon Transcribe，該活動會與 CloudTrail 事件 CloudTrail 歷史記錄中的其他 AWS 服務 事件一起記錄在事件中。您可以在 中檢視、搜尋和下載最近的事件 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](#)。

若要取得 中事件的持續記錄 AWS 帳戶，包括的事件 Amazon Transcribe，請建立追蹤。線索是一種組態，可讓 CloudTrail 將事件做為日誌檔案交付至指定的 Amazon S3 bucket. CloudTrail log 檔案，其中包含一或多個日誌項目。事件代表來自任何來源的單一請求。它包含有關請求的動作、動作的日期和時間、請求參數等資訊。CloudTrail log 檔案不是公有 API 呼叫的排序堆疊追蹤，因此不會以任何特定順序顯示。

根據預設，當您在 中建立線索時 AWS 管理主控台，線索會套用至所有 AWS 區域。線索會記錄 AWS 分割區 AWS 區域 中所有的事件，並將日誌檔案傳送到您指定的儲存 Amazon S3 貯體。此外，您可以設定其他 AWS 服務 來進一步分析和處理 CloudTrail 日誌中收集的事件資料。如需詳細資訊，請參閱：

- [建立追蹤的概觀](#)
- [CloudTrail 支援的服務和整合](#)
- [設定的 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 會記錄 API [參考](#)中記錄的所有 Amazon Transcribe 動作。例如，[CreateVocabulary](#)、[GetTranscriptionJob](#)和 [StartTranscriptionJob](#)操作會在 CloudTrail 日誌檔案中產生項目。當 CloudTrail 記錄 Amazon Transcribe API 操作時，CloudTrail 日誌項目會對請求和回應參數中的敏感資訊使用空字串，例如 Amazon S3 URI 值。

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：

- 是否使用根登入資料或 IAM 使用者登入資料提出請求
- 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 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 和多個的 Amazon Transcribe 日誌檔案彙總 AWS 帳戶 到單一儲存 Amazon S3 貯體。如需詳細資訊，請參閱[從多個區域接收 CloudTrail 日誌檔案](#)和[從多個帳戶接收 CloudTrail 日誌檔案](#)。

範例：Amazon Transcribe 日誌檔案項目

線索是一種組態，可讓您將事件做為日誌檔案交付至指定的 Amazon S3 bucket. CloudTrail log 檔案，其中包含一或多個日誌項目。事件代表來自任何來源的單一請求。它包含請求動作做為動作日期和時間的這類資訊，而 request parameters. CloudTrail log 檔案不是公有 API 呼叫的排序堆疊追蹤，因此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。

#### Note

當 CloudTrail 記錄 Amazon Transcribe API 操作時，CloudTrail 日誌項目會對請求和回應參數中的敏感資訊使用空字串，例如 Amazon S3 URI 值。

```
{
  "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": ""
    }
},
"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": ""
        }
    }
},
"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": ""
        },
        "transcript": {
            "transcriptFileUri": ""
        }
    },
    "requestID": "BD8798EACDD16751",
    "eventID": "607b9532-1423-41c7-b048-ec2641693c47",
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
]
```

## Amazon EventBridge 搭配 使用 Amazon Transcribe

使用 Amazon EventBridge，您可以透過在其他 中啟動事件來回應任務 Amazon Transcribe 中的狀態變更 AWS 服務。當轉錄任務變更狀態時，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，在轉錄任務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的通知，包括電子郵件、[聊天應用程式中的 Amazon Q Developer](#) 聊天通知或 [AWS Console Mobile Application](#) 推播通知。您也可以[在主控台通知中心](#) 查看通知。如果您想要設定通知，您可以使用 [AWS 使用者通知](#)。AWS 使用者通知 supports 彙總，這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 管理主控台](#)。定義規則時，請使用 Amazon Transcribe 做為服務名稱。如需如何建立 EventBridge 規則的範例，請參閱 [Amazon EventBridge 規則](#)。

使用前 EventBridge，請注意下列定義：

- 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 EventBridge 監看的傳入事件，並將其路由至目標或目標進行處理。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 事件會盡最大努力發出。如需在 中建立和管理事件的詳細資訊 EventBridge，請參閱 Amazon EventBridge 《使用者指南》中的 [Amazon EventBridge 事件](#)。

以下是轉錄任務的狀態變更為 `COMPLETED` 或 `FAILED` 時 Amazon Transcribe 啟動的 EventBridge 規則範例。

```
{
  "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。對於 Amazon Transcribe，這一律是 `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 會記錄數個 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`：EventBridge 為事件產生的唯一識別符。
- `detail-type`：事件詳細資訊的識別子。例如 Transcribe Job State Change。
- `source`：事件的來源。因此一律 Amazon Transcribe 為 `aws.transcribe`。
- `account`：產生 API 呼叫的帳戶 AWS 帳戶 ID。
- `time`：事件傳送的日期和時間。
- `region`：發出請求 AWS 區域的。
- `resources`：API 呼叫使用的資源。對於 Amazon Transcribe，此欄位一律為空白。
- `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

當任務的狀態從 變更為 `IN_PROGRESSCOMPLETED` 或 `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"),
    "FailureReason": "failure reason", // only present when JobStatus is 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。
- FailureReason：只有在 JobStatus 為 FAILED 時，才會出現此欄位 FAILED，並說明失敗的原因。

- `AnalyticsJobDetails` : 通話分析轉錄任務的詳細資訊，包括略過的分析功能的相關資訊。

## 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://amzn-s3-demo-bucket/my-output-files/my-redacted-file.JSON",
      "TranscriptFileUri": "s3://amzn-s3-demo-bucket/my-output-files/my-file.JSON"
    },
    "Media": {
      "MediaFileUri": "s3://amzn-s3-demo-bucket/my-output-files/my-redacted-file.WAV",
      "RedactedMediaFileUri": "s3://amzn-s3-demo-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。

## AWS HealthScribe 串流後分析事件

當 a AWS HealthScribe 串流後分析操作的狀態變更時，例如從 變更為 IN\_PROGRESS 的 [ClinicalNoteGenerationResult](#) COMPLETED，AWS HealthScribe 會產生包含下列資訊的事件：

```
{
  "version": "0",
  "id": "event ID",
  "detail-type": "MedicalScribe Post Stream Analytics Update",
  "source": "aws.transcribe",
  "account": "111122223333",
  "time": "timestamp",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "SessionId": <SessionID>,
    "UpdateType": "ClinicalNoteGenerationResult",
    "ClinicalNoteGenerationResult": {
      "ClinicalNoteOutputLocation": s3://amzn-s3-demo-bucket/clinical-note-output-files/clinical-notes.JSON,

```

```

    "TranscriptOutputLocation": s3://amzn-s3-demo-bucket/my-output-files/my-
file.JSON,
    "Status": <IN_PROGRESS | COMPLETED | FAILED>,
    "FailureReason": <failure_reason>
  }
}
}

```

- **UpdateType** : 產生事件的串流後分析操作類型。結果物件的內容會根據 `UpdateType` 而有所不同。
- **SessionId** : AWS HealthScribe 串流的識別號碼。使用此 ID 來識別原始串流工作階段，然後尋找產生事件的串流後分析。
- **Status** : 串流後分析操作的狀態。此值可以為 `IN_PROGRESS`、`COMPLETED` 或 `FAILED`。
- **ClinicalNoteOutputLocation** : 輸出 Amazon S3 儲存貯體的 `URIClinicalNoteGenerationResult`。
- **TranscriptOutputLocation** : 文字記錄的 URI。

## 詞彙事件

當 **自訂詞彙** 的狀態從 變更為 `PENDINGREADY` 或 `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。

### Note

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。

## 的合規驗證 Amazon Transcribe

若要了解 是否 AWS 服務 在特定合規計劃範圍內，請參閱[AWS 服務 合規計劃範圍內](#)然後選擇您感興趣的合規計劃。如需一般資訊，請參閱[AWS 合規計劃](#)。

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[在 中下載報告 AWS Artifact](#)。

您使用 時的合規責任 AWS 服務 取決於資料的機密性、您公司的合規目標，以及適用的法律和法規。如需使用 時合規責任的詳細資訊 AWS 服務，請參閱 [AWS 安全文件](#)。

## 中的彈性 Amazon Transcribe

AWS 全域基礎設施是以 AWS 區域 和 可用區域為基礎建置。AWS 區域 提供多個實體分隔和隔離的可用區域，這些區域與低延遲、高輸送量和高度備援聯網連接。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、容錯能力和擴展能力，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 和 可用區域的詳細資訊，請參閱[AWS 全球基礎設施](#)。

## 中的基礎設施安全 Amazon Transcribe

作為受管服務，Amazon Transcribe 受到 AWS 全球網路安全的保護。如需 AWS 安全服務以及如何 AWS 保護基礎設施的資訊，請參閱[AWS 雲端安全](#)。若要使用基礎設施安全的最佳實務來設計您的 AWS 環境，請參閱安全支柱 AWS Well-Architected Framework 中的[基礎設施保護](#)。

您可以使用 AWS 發佈的 API 呼叫，Amazon Transcribe 透過網路存取。使用者端必須支援下列專案：

- 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。

## 中的漏洞分析和**管理 Amazon Transcribe**

組態和 IT 控制是客戶 AWS 與您之間**的共同責任**。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)**

您可以在 VPC 和 之間建立私有連線，Amazon Transcribe 方法是建立**介面 VPC 端點**。介面端點採用 [AWS PrivateLink](#)技術，可讓您在沒有網際網路閘道、NAT 裝置、VPN 連線或 Direct Connect 連線的情況下，用來私下存取 Amazon Transcribe APIs。VPC 中的執行個體不需要公有 IP 地址即可與 Amazon Transcribe APIs通訊。VPC 與 之間的流量 Amazon Transcribe 不會離開 Amazon 網路。

每個介面端點都是由您子網路中的一或多個[彈性網路介面](#)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**

設定**介面 VPC 端點**之前 Amazon Transcribe，請務必檢閱Amazon VPC 《使用者指南》中的[介面端點屬性和限制](#)。

Amazon Transcribe 支援從您的 VPC 呼叫其所有 API 動作。

### 為 建立**介面 VPC 端點 Amazon Transcribe**

您可以使用 AWS 管理主控台 或 為 Amazon Transcribe 服務建立 VPC Amazon VPC 端點 AWS CLI。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的[建立介面端點](#)。

對於 中的**批次轉錄 Amazon Transcribe**，請使用下列服務名稱建立 VPC 端點：

- `com.amazonaws.us-west-2.transcribe`

對於 中的**串流轉錄 Amazon Transcribe**，請使用下列服務名稱建立 VPC 端點：

- `com.amazonaws.us-west-2.transcribestreaming`

如果您為端點啟用私有 DNS，您可以使用 Amazon Transcribe 其預設 DNS 名稱向 提出 API 請求 AWS 區域，例如 `transcribestreaming.us-east-2.amazonaws.com`。

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的[透過介面端點存取服務](#)。

## 為 建立 VPC 端點政策 Amazon Transcribe

您可以將端點政策連接至 VPC 端點，以控制對 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《Amazon VPC 使用者指南》中的[使用 VPC 端點控制對服務的存取](#)。

### 範例：Amazon Transcribe 批次轉錄動作的 VPC 端點政策

以下是在 Amazon Transcribe 中批次轉錄的端點政策的範例。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：Amazon Transcribe 串流轉錄動作的 VPC 端點政策

以下是在 Amazon Transcribe 中串流轉錄的端點政策的範例。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

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 共用的相關資訊，請參閱 Amazon Virtual Private Cloud 指南中的[與其他帳戶共用您的 VPC](#)。

## 的安全最佳實務 Amazon Transcribe

以下最佳實務為一般準則，並不代表完整的安全解決方案。這些最佳實務可能不適用或無法滿足您的環境需求，因此請將其視為實用建議就好，而不要當作是指示。

- 使用資料加密，例如 AWS KMS 加密內容

AWS KMS 加密內容是純文字、非私密金鑰：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)，可提供您的資料額外的安全層。

如需詳細資訊，請參閱 [AWS KMS 加密內容](#)。

- 盡可能使用臨時憑證

盡可能使用暫時憑證，而非諸如存取金鑰等長期憑證。對於需要具有程式設計存取和長期登入資料的 IAM 使用者，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 [輪換存取金鑰](#) 和 [IAM 的安全性最佳實務](#)。

- 針對需要 Amazon Transcribe 存取的應用程式 AWS 和服務使用 IAM 角色

使用 IAM 角色來管理需要存取的應用程式或服務臨時登入資料 Amazon Transcribe。當您使用角色時，您不必將密碼或存取金鑰等長期登入資料分發給 Amazon EC2 執行個體 AWS 或服務。IAM 角色可以提供暫時許可，供應用程式在向 AWS 資源提出請求時使用。

如需詳細資訊，請參閱 [IAM 角色](#) 和 [常見的角色情境：使用者、應用程式和服務](#)。

- 使用標籤式存取控制

您可以使用標籤來控制 中的存取 AWS 帳戶。在 Amazon Transcribe，標籤可以新增至：轉錄作業、自訂詞彙、自訂詞彙篩選和自訂語言模型。

如需詳細資訊，請參閱 [標籤式存取控制](#)。

- 使用 AWS 監控工具

監控是維護 Amazon Transcribe 和 AWS 解決方案可靠性、安全性、可用性和效能的重要部分。您可以使用 監控 Amazon Transcribe CloudTrail。

如需詳細資訊，請參閱 [Amazon Transcribe 使用 監控 AWS CloudTrail](#)。

- 啟用 AWS Config

AWS Config 可以評估、稽核和評估 AWS 資源的組態。使用 AWS Config，您可以檢閱組態的變更，以及 AWS 資源之間的關係。您可調查詳細的資源組態歷史記錄，也可依據內部準則指定的組態以判斷整體合規情況。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 醫療

Amazon Transcribe Medical 是一種自動語音辨識 (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，例如醫生口述筆記、藥物安全監控、遠距醫療預約或醫生與患者對話。Amazon Transcribe 醫療可透過即時串流（透過麥克風）或上傳檔案（批次）的文字記錄來提供。

## ⚠ Important

Amazon Transcribe 醫療不能取代專業醫療建議、診斷或治療。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人員審核和驗證結果。Amazon Transcribe 醫療轉錄只有在經過訓練的醫療專業人員審核準確性和良好的醫療判斷後，才應用於病患照護案例。

Amazon Transcribe Medical 以共同的責任模型運作，因此 AWS 負責保護執行 Amazon Transcribe Medical 的基礎設施，而且您負責管理您的資料。如需詳細資訊，請參閱[共同責任模型](#)。

Amazon Transcribe 醫療提供美國英文 (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。Amazon Transcribe Medical 支援 16,000 Hz 或更高的取樣率。

若要分析您的文字記錄，您可以使用其他 AWS 服務，例如 [Amazon Comprehend Medical](#)。

## 支援的專科

專科	子專科	音訊輸入
心臟科	無	僅串流
神經病學	無	僅串流
腫瘤學	無	僅串流
初級護理	家庭醫學	批次，串流
初級護理	內科	批次，串流
初級護理	婦產科 (OB-GYN)	批次，串流

專科	子專科	音訊輸入
初級護理	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

下列支援呼叫分析 AWS 區域：

區域	轉錄類型
af-south-1(開普敦)	批次
ap-east-1(香港)	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(倫敦)	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(聖保羅)	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(舊金山)	批次
us-west-2(奧勒岡)	批次，串流

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。

## 醫療專業與術語

建立醫學轉錄任務時，請指定來源檔案的語言、醫療專業和音訊類型。輸入美式英文 (en-US) 作為語言，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。

若要儲存轉錄任務的輸出，請選取您已建立的儲存 Amazon S3 貯體。如需儲存 Amazon S3 貯體的詳細資訊，請參閱 [入門 Amazon Simple Storage Service](#)。

以下是在 JSON 中輸入的請求參數最小數量：

```
{
  "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 Medical 可讓您產生替代轉錄。如需詳細資訊，請參閱 [產生替代轉錄](#)。

您也可以啟用發言者分隔或識別音訊中的聲道。如需詳細資訊，請參閱 [啟用發言者分隔](#) 及 [轉錄多聲道音訊](#)。

## 轉錄醫療術語及衡量值

Amazon Transcribe Medical 可以轉錄醫療術語和測量。Amazon Transcribe 醫療輸出口語術語的縮寫。例如，「血壓」會轉錄成 BP。您可以在此頁面的表格中找到 Amazon Transcribe Medical 用於醫療術語和測量的慣例清單。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。

來源音訊中所說的術語	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。

來源音訊中所說的術語	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 掃描	顯示腦出血的病患，需要使用 CT 掃描。

來源音訊中所說的術語	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

## 轉錄數字

Amazon Transcribe Medical 會將數字轉錄為數字，而不是單字。例如，口述「一千兩百四十二」會轉錄為 1242。

數字轉錄將依下列規則進行：

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; 今年是 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>

## 轉錄醫療對話

您可以使用 Amazon Transcribe Medical，使用批次轉錄任務或即時串流來轉錄臨床醫生和患者之間的醫療對話。批次轉錄作業可讓您轉錄音訊檔案。為了確保 Amazon Transcribe Medical 以最高的準確性產生轉錄結果，您必須在轉錄任務或串流中指定臨床醫生的醫療專科。

您可以透過以下醫療專業，轉錄臨床醫生和患者問診：

- 心臟科 — 僅適用於串流轉錄
- 神經病學 — 僅適用於串流轉錄
- 腫瘤學 — 僅適用於串流轉錄
- 初級護理 — 包括以下類型的醫療實務：
  - 家庭醫學
  - 內科
  - 婦產科 (OB-GYN)
  - 小兒科
- 泌尿外科 — 僅適用於串流轉錄

您可以使用醫學自訂詞彙以提高轉錄準確性。如需醫療自訂詞彙運作方式的資訊，請參閱 [使用醫學自訂詞彙提高轉錄準確性](#)。

根據預設，Amazon Transcribe Medical 會傳回可信度最高的轉錄。如果您想要設定為傳回替代轉錄，請參閱 [產生替代轉錄](#)。

有關數字和醫療測量如何在轉錄輸出顯示的資訊，請參閱 [轉錄數字](#) 和 [轉錄醫療術語及衡量值](#)。

### 主題

- [轉錄醫療對話的音訊檔案](#)

- [在即時串流中轉錄醫療對話](#)
- [啟用發言者分隔](#)
- [轉錄多聲道音訊](#)

## 轉錄醫療對話的音訊檔案

使用批次轉錄作業以轉錄醫療對話的音訊檔案。您可以使用此以轉錄臨床醫生與患者的對話。您可以使用 [StartMedicalTranscriptionJob](#) API 或 AWS 管理主控台開始批次轉錄作業。

您使用 [StartMedicalTranscriptionJob](#) API 開始醫學轉錄作業時，請指定 PRIMARYCARE 為 Specialty 參數的值。

### AWS 管理主控台

#### 轉錄臨床醫生與患者的對話 (AWS 管理主控台)

若要使用 AWS 管理主控台 轉錄臨床醫生-患者對話，請建立轉錄任務，然後選擇音訊輸入類型的對話。

1. 登入 [AWS 管理主控台](#)。
2. 在導覽窗格的 Amazon Transcribe Medical 下，選擇轉錄任務。
3. 選擇建立作業。
4. 在指定作業詳細資訊頁面的工作設定下，指定下列項目。
  - a. 名稱 — 轉錄作業的名稱。
  - b. 音訊輸入類型 — 對話
5. 針對其餘欄位，指定音訊檔案 Amazon S3 的位置，以及您要存放轉錄任務輸出的位置。
6. 選擇下一步。
7. 選擇建立。

### API

#### 使用批次轉錄作業 (API) 轉錄醫療對話

- 對於 [StartMedicalTranscriptionJob](#) API，請指定以下項目。
  - a. 對於 MedicalTranscriptionJobName，請指定 AWS 帳戶的唯一名稱。

- b. 對於 LanguageCode，請指定與音訊檔案中所說的語言對應的語言代碼，以及詞彙篩選語言對應的語言。
- c. 對於 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，請指定在音訊檔案 PRIMARYCARE 中說話的臨床醫生的醫療專科。
- e. 對於 Type，請指定 CONVERSATION。
- f. 對於 OutputBucketName，指定 Amazon S3 儲存貯體以存放轉錄結果。

以下是使用適用於 Python (Boto3) 的 AWS SDK 轉錄臨床醫生在PRIMARYCARE專科和患者醫療對話的範例請求。

```
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://amzn-s3-demo-bucket/my-input-files/my-audio-file.flac"
transcribe.start_medical_transcription_job(
    MedicalTranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'amzn-s3-demo-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)
```

以下範例代碼顯示臨床醫生和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

### 使用批次轉錄作業轉錄醫療對話 (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 `example-start-command.json` 的內容。

```
{  
  "MedicalTranscriptionJobName": "my-first-med-transcription-job",  
  "Media": {  
    "MediaFileUri": "s3://amzn-s3-demo-bucket/my-input-files/my-audio-file.flac"  
  },  
  "OutputBucketName": "amzn-s3-demo-bucket",  
  "OutputKey": "my-output-files/",  
  "LanguageCode": "en-US",  
  "Specialty": "PRIMARYCARE",  
  "Type": "CONVERSATION"  
}
```

## 在即時串流中轉錄醫療對話

您可以使用 HTTP/2 或 [WebSocket](#) 通訊協定以轉錄醫療對話的音訊串流。如需如何使用 WebSocket 通訊協定以開始串流的資訊，請參閱 [設定 WebSocket 串流](#)。若要開始 HTTP/2 串流，請使用 [StartMedicalStreamTranscription](#) API。

您可以使用下列醫療專科，轉錄串流音訊：

- 心臟科
- 神經病學
- 腫瘤學
- 初級護理

- 泌尿外科

每個醫療專業包括多種類型的程序和預約。因此，臨床醫生會決定許多不同類型的筆記。請使用下列範例作為準則，協助您指定 WebSocket 請求的 specialty URI 參數值，或 [StartMedicalStreamTranscription](#) API 的 Specialty 參數：

- 對於電生理學或超音波心電圖諮詢，請選擇 CARDIOLOGY。
- 對於腫瘤內科、腫瘤外科或腫瘤放射學諮詢，請選擇 ONCOLOGY。
- 對於提供中風 (暫時性缺血性或腦血管性) 的患者諮詢的醫生，請選擇 NEUROLOGY。
- 對於尿失禁的諮詢，請選擇 UROLOGY。
- 對於每年檢查或緊急護理問診，請選擇 PRIMARYCARE。
- 對於住院者住院問診，請選擇 PRIMARYCARE。
- 對於生產、輸卵管結紮、IUD 插入或墮胎的諮詢，請選擇 PRIMARYCARE。

## AWS 管理主控台

### 轉錄串流醫療對話 (AWS 管理主控台)

若要使用 AWS 管理主控台 在即時串流中轉錄臨床醫生與患者對話，請選擇轉錄醫療對話、啟動串流，以及開始對麥克風說話的選項。

1. 登入 [AWS 管理主控台](#)。
2. 在導覽窗格的 Amazon Transcribe Medical 下，選擇即時轉錄。
3. 搜尋對話。
4. 對於醫療專科，請選擇臨床醫生的專科。
5. 選擇開始串流。
6. 對著麥克風說話。

### 在 HTTP/2 串流轉錄醫療對話

以下是 HTTP/2 請求的參數語法。

若要轉錄醫療對話的 HTTP/2 串流，請使用 [StartMedicalStreamTranscription](#) API 並指定以下項目：

- LanguageCode — 語言代碼。有效值為 en-US。

- MediaEncoding — 輸入音訊使用的編碼。有效值為 pcm、ogg-opus 和 flac。
- Specialty — 醫療專業人員的專科。
- Type – CONVERSATION

若要提高即時串流中特定詞彙的轉錄準確性，請使用自訂詞彙。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 [使用醫學自訂詞彙提高轉錄準確性](#)。

若要標示來自不同發言者的語音，請將 ShowSpeakerLabel 參數設定為 true。如需詳細資訊，請參閱 [啟用發言者分隔](#)。

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 串流轉錄醫療對話

您可以使用 WebSocket 請求以轉錄醫療對話。您提出 WebSocket 請求時，您會建立預先簽署的 URI。此 URI 包含在您的應用程式和 Amazon Transcribe Medical 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 WebSocket 串流](#)。

使用下列範本以建立您的預先簽署 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，請使用自訂詞彙。若要啟用自訂字彙，請 `vocabulary-name` 將的值設定為您要使用的自訂詞彙的名稱。如需詳細資訊，請參閱 [使用醫學自訂詞彙提高轉錄準確性](#)。

若要標示來自不同發言者的語音，請將 `show-speaker-label` 參數設定為 `true`。如需詳細資訊，請參閱 [啟用發言者分隔](#)。

如需建立預先簽署 URI 的詳細資訊，請參閱 [設定 WebSocket 串流](#)。

## 啟用發言者分隔

若要在 Amazon Transcribe Medical 中啟用發言者分割，請使用發言者日記。這讓您可以查看在轉錄輸出中患者說的內容以及臨床醫生說的內容。

當您啟用發言者日記時，Amazon Transcribe Medical 會使用每個發言者的唯一識別符來標記每個發言者表達用語。表達用語是語音單位，通常會以靜音區分其他表達用語。在批次轉錄中，臨床醫生的表達用語會收到 `spk_0` 的標示而患者會收到 `spk_1` 的標示。

如果一個發言者的表達用語與另一位發言者的表達用語重疊，Amazon Transcribe Medical 會按照開始時間在轉錄中排序。輸入音訊中重疊的表達用語在轉錄輸出中不會重疊。

您使用批次轉錄作業或即時串流轉錄音訊檔案時，您可以啟用發言者日記功能。

### 主題

- [在批次轉錄中啟用發言者分隔](#)
- [在即時串流中啟用發言者分隔](#)

## 在批次轉錄中啟用發言者分隔

您可以在批次轉錄作業中使用 [StartMedicalTranscriptionJob](#) API 或 AWS 管理主控台。這讓您可以在臨床醫生和患者對話中，進行每個發言者的分隔，並確定在轉錄輸出中發言者說的內容。

### AWS 管理主控台

若要使用 AWS 管理主控台 在轉錄任務中啟用發言者日記，您可以啟用音訊識別，然後啟用發言者分割。

1. 登入 [AWS 管理主控台](#)。
2. 在導覽窗格的 Amazon Transcribe Medical 下，選擇轉錄任務。
3. 選擇建立作業。

4. 在指定作業詳細資訊頁面上，提供轉錄作業的相關資訊。
5. 選擇下一步。
6. 啟用音訊識別。
7. 對於音訊識別類型，請選擇發言者分隔。
8. 對於最大發言者數量，請輸入您認為在音訊檔案中說話的最大發言者數量。
9. 選擇建立。

## API

### 使用批次轉錄作業 (API) 啟用發言者分隔

- 對於 [StartMedicalTranscriptionJob](#) API，請指定以下項目。
  - a. 對於 `MedicalTranscriptionJobName`，請指定在 AWS 帳戶中唯一的名稱。
  - 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。
  - c. 對於 `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`，請指定在音訊檔案中說話的臨床醫生的醫療專科。
  - e. 對於 `Type`，請指定 `CONVERSATION`。
  - f. 對於 `OutputBucketName`，指定儲存 Amazon S3 貯體以存放轉錄結果。
  - 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發言者數量。

下列請求使用 適用於 Python (Boto3) 的 AWS SDK 啟動已啟用發言者分割之主要照護臨床醫生患者對話的批次轉錄任務。

```
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://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
transcribe.start_medical_transcription_job(
```

```

MedicalTranscriptionJobName = job_name,
Media={
    'MediaFileUri': job_uri
},
OutputBucketName = 'amzn-s3-demo-bucket',
OutputKey = 'my-output-files/',
LanguageCode = 'en-US',
Specialty = 'PRIMARYCARE',
Type = 'CONVERSATION',
OutputBucketName = 'amzn-s3-demo-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)

```

下列範例程式碼示範啟用發言者分隔的轉錄作業的轉錄結果。

```

{
  "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

### 轉錄執行初級護理的臨床醫生與患者對話的音訊檔案 (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 `example-start-command.json` 的內容。

```
{  
  "MedicalTranscriptionJobName": "my-first-med-transcription-job",  
  "Media": {  
    "MediaFileUri": "s3://amzn-s3-demo-bucket/my-input-files/my-audio-file.flac"  
  },  
  "OutputBucketName": "amzn-s3-demo-bucket",  
  "OutputKey": "my-output-files/",  
  "LanguageCode": "en-US",  
  "Specialty": "PRIMARYCARE",  
  "Type": "CONVERSATION",  
  "Settings": {  
    "ShowSpeakerLabels": true,  
    "MaxSpeakerLabels": 2  
  }  
}
```

## 在即時串流中啟用發言者分隔

若要在即時串流中分割發言者並標記其語音，請使用 AWS 管理主控台 或串流請求。發言者分隔最適合兩個至五個發言者的串流。雖然 Amazon Transcribe Medical 可以在串流中分割超過五個發言者，但如果超過該數字，則分割區的準確性會降低。

若要開始 HTTP/2 請求，請使用 [StartMedicalStreamTranscription](#) API。若要開始 WebSocket 請求，請使用預先簽署的 URI。URI 包含設定應用程式與 Amazon Transcribe Medical 設定雙向通訊所需的資訊。

### 在麥克風語音中啟用發言者分隔 (AWS 管理主控台)

您可以使用 AWS 管理主控台 來啟動臨床醫生與患者對話的即時串流，或即時對麥克風說話的口述。

1. 登入 [AWS 管理主控台](#)。
2. 在導覽窗格中，針對 Amazon Transcribe 醫療選擇即時轉錄。
3. 對於音訊輸入類型，選擇您要轉錄的醫療語音類型。
4. 對於其他設定，選擇發言者分隔。
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 串流中啟用發言者分隔

若要在醫療對話的 HTTP/2 串流中啟用發言者區隔，請使用 [StartMedicalStreamTranscription](#) API 並指定下列項目：

- 對於 LanguageCode，指定與串流中的語言相對應的語言代碼。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科。
- 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 請求中啟用發言者分隔

若要使用 API 以分隔 WebSocket 串流中的發言者，請使用下列格式以建立預先簽署的 URI，開始 WebSocket 請求並設定 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
```

下列程式碼示範串流請求的截斷範例回應。

```
{
  "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 Medical 會根據自然語音區段來中斷傳入的音訊串流，例如喇叭變更或音訊暫停。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。只有完全轉錄的區段會顯示發言者標籤。

下列清單顯示串流轉錄輸出中物件和參數的組織。

## Transcript

每個語音區段都有自己的 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。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單字和標點符號的相關資訊。當您啟用發言者分割時，每個單字都有完整轉錄語音區段的 Speaker 標籤。Amazon Transcribe Medical 使用此標籤為串流中的每個發言者指派唯一的整數。值為 speaker-change 的 Type 參數代表使用者已經停止說話，而另一個使用者即將開始。

## Transcript

每個項目物件都包含轉錄的語音區段作為 Transcript 欄位的值。

如需 WebSocket 請求的詳細資訊，請參閱 [設定 WebSocket 串流](#)。

## 轉錄多聲道音訊

如果您的音訊檔案或串流有多個頻道，您可以使用頻道識別來轉錄每個頻道的語音。Amazon Transcribe Medical 會分別轉錄每個頻道的語音。它將每個聲道的獨立轉錄與單一轉錄輸出結合。

使用聲道識別以識別音訊中不同的聲道，並從每個聲道轉錄語音。在通話者和客服人員案例等情況下啟用此功能。使用此功能可區分通話者與執行藥物安全監控的聯絡中心的錄音或串流中的客服人員。

您可以為批次處理和即時串流，啟用聲道識別。下列清單描述如何為每個方法啟用。

- 批次轉錄 – AWS 管理主控台 和 [StartMedicalTranscriptionJob](#) API
- 串流轉錄 — WebSocket 串流和 [StartMedicalStreamTranscription](#) API

## 轉錄多聲道音訊檔案

當您轉錄音訊檔案時，Amazon Transcribe Medical 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。每個單字都有開始時間和結束時間。如果聲道上的使用者與不同聲道上的某個人說話，則每個聲道項目的開始時間和結束時間會重疊，且個人彼此交談時間也會重疊。

依預設，您可以轉錄有兩個聲道的音訊檔案。如果您需要轉錄有兩個以上聲道的檔案，可以要求提高配額。如需請求提高配額的詳細資訊，請參閱 [AWS 服務配額](#)。

若要在批次轉錄任務中轉錄多聲道音訊，請使用 AWS 管理主控台 或 [StartMedicalTranscriptionJob](#) API。

## AWS 管理主控台

若要使用 AWS 管理主控台 在批次轉錄任務中啟用頻道識別，您可以啟用音訊識別，然後啟用頻道識別。頻道識別是 中音訊識別的子集 AWS 管理主控台。

1. 登入 [AWS 管理主控台](#)。
2. 在導覽窗格的 Amazon Transcribe Medical 下，選擇轉錄任務。
3. 選擇建立作業。
4. 在指定作業詳細資訊頁面上，提供轉錄作業的相關資訊。
5. 選擇下一步。
6. 啟用音訊識別。
7. 對於音訊識別類型，請選擇聲道識別。
8. 選擇建立。

## API

### 轉錄多聲道音訊檔案 (API)

- 對於 [StartMedicalTranscriptionJob](#) API，請指定以下項目。
  - a. 對於 TranscriptionJobName，指定 AWS 帳戶的唯一名稱。
  - 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。有效值為 en-US。
  - c. 對於 Media 物件的 MediaFileUri 參數，指定您要轉錄的媒體檔案名稱。
  - d. 對於 Settings 物件，設定 ChannelIdentification 為 true。

以下是使用 適用於 Python (Boto3) 的 AWS SDK 的範例請求。

```
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://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
transcribe.start_medical_transcription_job(
    MedicalTranscriptionJobName = job_name,
```

```
Media = {
  'MediaFileUri': job_uri
},
OutputBucketName = 'amzn-s3-demo-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

### 使用批次轉錄作業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 example-start-command.json 的程式碼。

```
{
  "MedicalTranscriptionJobName": "my-first-med-transcription-job",
  "Media": {
    "MediaFileUri": "s3://amzn-s3-demo-bucket/my-input-files/my-audio-
file.flac"
  },
}
```

```
"OutputBucketName": "amzn-s3-demo-bucket",
"OutputKey": "my-output-files/",
"LanguageCode": "en-US",
"Specialty": "PRIMARYCARE",
"Type": "CONVERSATION",

  "Settings":{
    "ChannelIdentification": true
  }
}
```

下列程式碼示範在兩個聲道有對話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 串流中的不同聲道轉錄音訊。

依預設，您可以使用兩個聲道轉錄串流。如果您需要轉錄有兩個以上聲道的串流，可以要求提高配額。如需請求提高配額的詳細資訊，請參閱 [AWS Service Quotas](#)。

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 — 您串流音訊中聲道的數量。

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

若要分隔 WebSocket 串流中的發言者，請使用下列格式以建立預先簽署的 URI 並開始 WebSocket 請求。指定 `enable-channel-identification` 為 `true` 和 `number-of-channels` 串流中的聲道數量。預先簽署的 URI 包含設定應用程式與 Amazon Transcribe Medical 之間的雙向通訊所需的資訊。

```
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 操作常見的參數都列在 [通用參數](#) 區段中。

如需 WebSocket 請求的詳細資訊，請參閱 [設定 WebSocket 串流](#)。

### 多聲道串流輸出

對於 HTTP/2 和 WebSocket 請求，串流轉錄的輸出相同。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` 標記，指出該語音所屬的聲道。

## 轉錄醫療聽寫

您可以使用 Amazon Transcribe Medical，使用批次轉錄任務或即時串流轉錄臨床醫生指定的醫療備註。批次轉錄作業可讓您轉錄音訊檔案。您可以在轉錄任務或串流中指定臨床醫生的醫療專科，以確保 Amazon Transcribe Medical 以最高的準確性產生轉錄結果。

您可以轉錄以下專業的醫療聽寫：

- 心臟科 — 僅適用於串流轉錄
- 神經病學 — 僅適用於串流轉錄
- 腫瘤學 — 僅適用於串流轉錄
- 初級護理 — 包括以下類型的醫療實務：
  - 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 — 僅適用於串流轉錄

您可以使用自訂詞彙提高轉錄準確性。如需醫療自訂詞彙運作方式的資訊，請參閱 [使用醫學自訂詞彙提高轉錄準確性](#)。

根據預設，Amazon Transcribe Medical 會傳回可信度最高的轉錄。如果您想要設定為傳回替代轉錄，請參閱 [產生替代轉錄](#)。

有關數字和醫療測量如何在轉錄輸出顯示的資訊，請參閱 [轉錄數字](#) 和 [轉錄醫療術語及衡量值](#)。

## 主題

- [轉錄醫療聽寫的音訊檔案](#)
- [在即時串流中轉錄醫療聽寫](#)

## 轉錄醫療聽寫的音訊檔案

使用批次轉錄作業以轉錄醫療對話的音訊檔案。您可以使用此以轉錄臨床醫生與患者的對話。您可以使用 [StartMedicalTranscriptionJob](#) API 或 AWS 管理主控台開始批次轉錄作業。

您使用 [StartMedicalTranscriptionJob](#) API 開始醫學轉錄作業時，請指定 PRIMARYCARE 為 Specialty 參數的值。

### AWS 管理主控台

#### 轉錄臨床醫生與患者的對話 (AWS 管理主控台)

若要使用 AWS 管理主控台 轉錄臨床醫生-患者對話，請建立轉錄任務，然後選擇音訊輸入類型的對話。

1. 登入 [AWS 管理主控台](#)。
2. 在導覽窗格的 Amazon Transcribe Medical 下，選擇轉錄任務。
3. 選擇建立作業。
4. 在指定作業詳細資訊頁面的工作設定下，指定下列項目。
  - a. 名稱 — 轉錄作業的名稱。
  - b. 音訊輸入類型 — 聽寫
5. 針對其餘欄位，指定音訊檔案 Amazon S3 的位置，以及您要存放轉錄任務輸出的位置。

6. 選擇下一步。
7. 選擇建立。

## API

### 使用批次轉錄作業 (API) 轉錄醫療對話

- 對於 [StartMedicalTranscriptionJob](#) API，請指定以下項目。
  - a. 對於 `MedicalTranscriptionJobName`，請指定 AWS 帳戶的唯一名稱。
  - b. 對於 `LanguageCode`，請指定與音訊檔案中所說的語言對應的語言代碼，以及詞彙篩選語言對應的語言。
  - c. 在 `Media` 對象的 `MediaFileUri` 參數中，指定您要轉錄的音訊檔案名稱。
  - d. 對於 `Specialty`，請指定在音訊檔案中說話的臨床醫生的醫療專科。
  - e. 對於 `Type`，請指定 `DICTATION`。
  - f. 對於 `OutputBucketName`，指定 Amazon S3 儲存貯體以存放轉錄結果。

以下是使用適用於 Python (Boto3) 的 AWS SDK 轉錄PRIMARYCARE專科臨床醫生醫療聽寫的範例請求。

```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe')
job_name = "my-first-med-transcription-job"
job_uri = "s3://amzn-s3-demo-bucket/my-input-files/my-audio-file.flac"
transcribe.start_medical_transcription_job(
    MedicalTranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'amzn-s3-demo-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

### 在批次轉錄作業中啟用發言者分隔 (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 `example-start-command.json` 的內容。

```
{  
  "MedicalTranscriptionJobName": "my-first-med-transcription-job",  
  "Media": {  
    "MediaFileUri": "s3://amzn-s3-demo-bucket/my-input-files/my-audio-file.flac"  
  },  
  "OutputBucketName": "amzn-s3-demo-bucket",  
  "OutputKey": "my-output-files/",  
  "LanguageCode": "en-US",  
  "Specialty": "PRIMARYCARE",  
  "Type": "DICTATION"  
}
```

## 在即時串流中轉錄醫療聽寫

使用 WebSocket 串流，將醫療聽寫轉錄為音訊串流。您也可以使用 AWS 管理主控台 轉錄您或其他人直接對麥克風說話的語音。

對於 HTTP/2 或 WebSocket 串流，您可以轉錄以下醫療專科的音訊：

- 心臟科
- 腫瘤學
- 神經病學
- 初級護理
- 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，臨床醫生會決定許多不同類型的筆記。請使用下列範例作為準則，協助您指定 WebSocket 請求的 specialty URI 參數值，或 [StartMedicalStreamTranscription](#) API 的 Specialty 參數：

- 對於電生理或超音波心電圖程序後的聽寫，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。
- 對於醫生口述顯示腦炎診斷的筆記，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 光影像的結果的聽寫，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，請使用自訂詞彙。若要啟用自訂詞彙，請將 vocabulary-name 的值設定為您要使用的自訂詞彙的名稱。

使用 轉錄口述到麥克風的口述 AWS 管理主控台

若要使用 AWS 管理主控台 轉錄醫療口述的串流音訊，請選擇轉錄醫療口述、啟動串流，以及開始對麥克風說話的選項。

轉錄醫療聽寫的串流音訊 (AWS 管理主控台)

1. 登入 [AWS 管理主控台](#)。
2. 在導覽窗格的 Amazon Transcribe Medical 下，選擇即時轉錄。
3. 選擇聽寫。
4. 對於醫療專科，請選擇在串流說話的臨床醫生的醫學專科。
5. 選擇開始串流。

## 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 串流，請使用 [StartMedicalStreamTranscription](#) API 並指定以下項目：

- LanguageCode — 語言代碼。有效值為 en-US。
- MediaEncoding — 輸入音訊使用的編碼。有效值為 pcm、ogg-opus 和 flac。
- Specialty — 醫療專業人員的專科。
- 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 串流媒體請求以轉錄醫療聽寫

若要使用 WebSocket 請求在即時串流中轉錄醫療聽寫，您需要建立預先簽署的 URI。此 URI 包含設定應用程式與 Amazon Transcribe Medical 之間的音訊串流所需的資訊。如需建立 WebSocket 請求的詳細資訊，請參閱 [設定 WebSocket 串流](#)。

使用下列範本以建立您的預先簽署 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 WebSocket 串流](#)。

## 使用醫學自訂詞彙提高轉錄準確性

若要改善 Amazon Transcribe Medical 中的轉錄準確性，請建立並使用一或多個醫療自訂詞彙。自訂詞彙是特定領域的單字或片語的集合。此集合有助於改善 Amazon Transcribe Medical 在轉錄這些字詞或片語時的效能。

當您使用 Amazon Transcribe Medical 時，您必須對自己的資料完整性負責。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)。

為了達到最佳效果，請建立獨立的小型自訂詞彙，每個詞彙都有助於轉錄特定的錄音。與建立用於所有錄音的大型自訂詞彙相較，您在轉錄準確性上感受到更大幅的改進。

根據預設，您的 中最多可以有 100 個自訂詞彙 AWS 帳戶。自訂詞彙的大小不得超過 50 KB。如需有關請求增加自訂詞彙數量的資訊 AWS 帳戶，請參閱 [AWS 服務配額](#)。

自訂詞彙提供美式英文版本 (en-US)。

### 主題

- [為您的醫療自訂詞彙建立文字檔案](#)
- [使用文字檔案建立醫學自訂詞彙](#)
- [使用醫學自訂詞彙轉錄音訊檔案](#)
- [使用醫學自訂詞彙轉錄即時串流](#)
- [Amazon Transcribe Medical 的字元集](#)

## 為您的醫療自訂詞彙建立文字檔案

若要建立自訂詞彙，您可以建立 UTF-8 格式的文字檔案。在此檔案中，您建立一個四欄表格，每欄指定一個欄位。每個欄位都會告訴 Amazon Transcribe Medical 網域特定詞彙的發音方式，或如何在轉錄中顯示這些詞彙。您可以將包含這些欄位的文字檔案存放在 Amazon S3 儲存貯體中。

### 了解如何格式化文字檔案

若要建立醫療自訂詞彙，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。

以下是表格四欄的名稱：

- **Phrase** – 必填欄，必填值
- **IPA** – 必填欄，可選填值
- **SoundsLike** – 必填欄，可選填值
- **DisplayAs** – 必填欄，可選填值

建立自訂詞彙時，請確定：

- 使用單一 Tab 字元分隔每個資料欄。如果您嘗試使用空格或多個 Tab 字元分隔資料欄，則會 Amazon Transcribe 擲出錯誤訊息。
- 確定欄中的每個值後沒有空格或空白格。

確定您為每一欄輸入的值：

- 不超過 256 個字元，連字號包括在內
- 僅使用允許字元集的字元，請參閱 [Amazon Transcribe Medical 的字元集](#)

## 輸入表格欄的值

下列資訊顯示如何為表格的四個欄指定值：

- **Phrase** – 應可辨識的單字或片語。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 [Amazon Transcribe Medical 的字元集](#)。如果您未指定資料 **DisplayAs** 欄，Amazon Transcribe Medical 會使用輸出檔案中資料 **Phrase** 欄的內容。

- **IPA**(必填欄，可選填值) – 若要指定單字或片語的發音，您可以在此欄中包含 [國際音標 \(IPA\)](#) 中的字元。IPA 欄不能包含前置或結尾空格，而且您必須使用單一空格以區隔輸入中的每個

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 [Amazon Transcribe Medical 的字元集](#)。美式英文是 Amazon Transcribe Medical 中唯一可用的語言。

- **SoundsLike**(必填欄，可選填值) – 您可以將單字或片語分成較小的區段，並使用語言的標準正字法，為每個區段提供發音，以模仿單字的發音。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 [Amazon Transcribe Medical 的字元集](#)。

- **DisplayAs** (必填欄，可選填值)– 定義單字或片語輸出時的外觀。例如，如果單字或片語為 **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。

如果您未指定資料 DisplayAs 欄，Amazon Transcribe Medical 會使用輸出中輸入檔案的資料 Phrase 欄。

您可以在 DisplayAs 欄中使用任何 UTF-8 字元。

您僅可在 IPA 和 DisplayAs 欄中的值納入空格。

若要建立自訂詞彙的文字檔案，請將文字檔案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 `的檔案儲存在您使用 Amazon Transcribe Medical 建立自訂詞彙 AWS 區域 的相同 Amazon S3 儲存貯.txt`體中。

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。它們只是為了描述這些範例。

片語	IPA	SoundsLike	DisplayAs
acute-respiratory-distress-syndrome			acute respiratory distress syndrome

片語	IPA	SoundsLike	DisplayAs
A.L.L.	eɪ ɛ   ɛ l		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 l	ALL

DisplayAs	SoundsLike	IPA	片語
acute respiratory distress syndrome			acute-respiratory-distress-syndrome
ALL		eɪ ɛ   ɛ l	A.L.L.
	ay-tree-o-ven-trick-ular-node-al-re-entr-ant-tack-ih-card-ia		atrioventricular-nodal-reentrant-tachycardia

## 使用文字檔案建立醫學自訂詞彙

若要建立自訂詞彙，您必須備妥文字檔案，其中包含字詞或片語的集合。Amazon Transcribe Medical 使用此文字檔案來建立自訂詞彙，您可以用來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 Medical 主控台建立自訂詞彙。

## AWS 管理主控台

若要使用 AWS 管理主控台 建立自訂詞彙，請提供文字檔案的 Amazon S3 URI，其中包含您的單字或片語。

1. 登入 [AWS 管理主控台](#)。
2. 在導覽窗格的 Amazon Transcribe Medical 下，選擇自訂詞彙。
3. 對於詞彙設定下的名稱，選擇自訂詞彙的名稱。
4. 在 Amazon S3 中指定音訊檔案或影片檔案的位置：
  - 對於詞彙設定下的 S3 詞彙輸入檔案位置，指定 Amazon S3 URI，識別用於建立自訂詞彙之文字檔案。
  - 對於 S3 中的詞彙輸入檔案位置，選擇瀏覽 S3 以瀏覽文字檔案並選擇。
5. 選擇建立詞彙。

您可以在 AWS 管理主控台中查看自訂詞彙的處理狀態。

## API

### 建立醫療自訂詞彙 (API)

- 對於 [StartTranscriptionJob](#) API，請指定以下項目。
  - a. 對於 LanguageCode，請指定 en-US。
  - b. 針對 VocabularyFileUri，指定您用來定義自訂詞彙的文字檔案 Amazon S3 位置。
  - c. 對於 VocabularyName，指定自訂詞彙的名稱。您指定的名稱在 中必須是唯一的 AWS 帳戶。

若要查看自訂詞彙的處理狀態，使用 [GetMedicalVocabulary](#) API。

以下是使用 適用於 Python (Boto3) 的 AWS SDK 建立自訂詞彙的範例請求。

```
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://amzn-s3-demo-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

在批次轉錄作業中啟用發言者分隔 (AWS CLI)

- 執行下列程式碼。

```
aws transcribe create-medical-vocabulary \
--vocabulary-name my-first-vocabulary \
--vocabulary-file-uri s3://amzn-s3-demo-bucket/my-vocabularies/my-vocabulary-
file.txt \
--language-code en-US
```

## 使用醫學自訂詞彙轉錄音訊檔案

使用 [StartMedicalTranscriptionJob](#) 或 AWS 管理主控台 啟動轉錄任務，該任務使用自訂詞彙來改善轉錄準確性。

### AWS 管理主控台

1. 登入 [AWS 管理主控台](#)。
2. 在導覽窗格的 Amazon Transcribe Medical 下，選擇轉錄任務。
3. 選擇建立作業。
4. 在指定作業詳細資訊頁面上，提供轉錄作業的相關資訊。
5. 選擇下一步。

6. 在自訂下，啟用自訂詞彙。
7. 在詞彙選擇下，選擇自訂詞彙。
8. 選擇建立。

## API

使用批次轉錄作業 (API) 在音訊檔案中啟用發言者分隔

- 對於 [StartMedicalTranscriptionJob](#) API，請指定以下項目。
  - a. 對於 `MedicalTranscriptionJobName`，請指定在 AWS 帳戶中唯一的名稱。
  - b. 對於 `LanguageCode`，請指定與音訊檔案中所說的語言對應的語言代碼，以及詞彙篩選語言對應的語言。
  - c. 對於 `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`，請指定在音訊檔案中說話的臨床醫生的醫療專科。
  - e. 對於 `Type`，指定音訊檔案是對話或聽寫。
  - f. 對於 `OutputBucketName`，指定 Amazon S3 儲存貯體以存放轉錄結果。
  - g. 對於 `Settings` 物件，請指定下列項目：
    - `VocabularyName` — 您的自訂詞彙的名稱。

下列請求使用 適用於 Python (Boto3) 的 AWS SDK 啟動具有自訂詞彙的批次轉錄任務。

```
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://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
transcribe.start_medical_transcription_job(
    MedicalTranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'amzn-s3-demo-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)
```

## 使用醫學自訂詞彙轉錄即時串流

若要提高即時串流中的轉錄準確性，您可以使用 HTTP/2 或 WebSocket 串流以使用自訂詞彙。若要開始 HTTP/2 請求，請使用 [StartMedicalStreamTranscription](#) API。您可以使用 AWS 管理主控台、[StartMedicalStreamTranscription](#) API 或使用 WebSocket 通訊協定，即時使用自訂詞彙。

轉錄您向麥克風口說話的聽寫 (AWS 管理主控台)

若要使用 AWS 管理主控台 轉錄醫療口述的串流音訊，請選擇轉錄醫療口述、啟動串流，以及開始對麥克風說話的選項。

轉錄醫療聽寫的串流音訊 (AWS 管理主控台)

1. 登入 [AWS 管理主控台](#)。
2. 在導覽窗格的 Amazon Transcribe Medical 下，選擇即時轉錄。
3. 對於醫療專科，請選擇在串流說話的臨床醫生的醫學專科。
4. 對於音訊輸入類型，選擇對話或聽寫。
5. 對於其他設定，請選擇自訂詞彙。
  - 對於詞彙選擇，請選擇自訂詞彙。
6. 選擇開始串流。

## 7. 對著麥克風說話。

在 HTTP/2 串流中啟用發言者分隔

以下是 HTTP/2 請求的參數語法。

```
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
```

參數描述：

- 主機：使用 AWS 區域 您正在呼叫的 更新 AWS 區域 ( 上述範例中的「us-west-2」)。如需有效清單 AWS 區域，請參閱 [AWS 區域和端點](#)。
- 授權：這是自動產生的欄位。若要進一步了解如何建立簽章，請參閱 [使用簽章第 4 版簽署 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：這是自動產生的欄位。若要進一步了解如何計算簽章，請參閱 [使用簽章第 4 版簽署 AWS 請求](#)。
- x-amz-date：簽章的建立日期與時間。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](#)。
- x-amzn-transcribe-session-id：串流工作階段的名稱。
- x-amzn-transcribe-language-code：用於輸入音訊的編碼。請參閱 [StartMedicalStreamTranscription](#) 或 [支援的語言和特定語言功能](#)，了解有效值的清單。

- `x-amzn-transcribe-media-encoding`：用於輸入音訊的編碼。有效值為 `pcm`、`ogg-opus` 和 `flac`。
- `x-amzn-transcribe-sample-rate`：輸入音訊的範例速率（以 Hertz 為單位）。Amazon Transcribe 支援從 8,000 Hz 到 48,000 Hz 的範圍。劣質的音訊，例如電話音訊，通常約為 8,000 Hz。優質的音訊，通常介於 16,000 Hz 至 48,000 Hz 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`：您要與轉錄搭配使用的詞彙名稱。
- `x-amzn-transcribe-specialty`：正在轉錄的醫療專科。
- `x-amzn-transcribe-type`：選擇是聽寫或對話。
- `x-amzn-transcribe-show-speaker-label`：要啟用日記，此值必須是 `true`。
- `content-type`：請勿變更此欄位；請使用上述範例中顯示的內容。

在 WebSocket 請求中啟用發言者分隔

若要使用 API 以分隔 WebSocket 串流中的發言者，請使用下列格式以建立預先簽署的 URI，開始 WebSocket 請求並設定 `vocabulary-name` 為自訂詞彙的名稱。

```
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
```

## Amazon Transcribe Medical 的字元集

若要在 Amazon Transcribe Medical 中使用自訂詞彙，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	Code	字元	Code
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
l	006C	g	0261
ˌ	006C 0329	ɪ	026A

字元	Code	字元	Code
m	006D	ɹ	0279
n	006E	ʃ	0283
ŋ	006E 0329	ʊ	028A
ou	006F 028A	ʌ	028C
p	0070	ɹ	028D
s	0073	ʒ	0292
t	0074	ɔ̃	02A4
u	0075	tʃ	02A7
v	0076	θ	03B8

## 識別轉錄中的個人健康資訊 (PHI)

使用個人健康資訊識別，在轉錄結果中標記個人健康資訊 (PHI)。透過查看標籤，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作業以識別 PHI。

您可以使用自己的後處理以修訂轉錄輸出中識別出的 PHI。

使用個人健康資訊識別以識別以下類型的 PHI：

- 個人 PHI：
  - 名稱 — 全名或姓氏和首字母
  - Gender
  - 年齡
  - 電話號碼
  - 與患者直接相關的日期 (不包括年份)
  - 電子郵件位址

- 地理位置 PHI :
  - 實體地址
  - 郵遞區號
  - 醫療中心或診所的名稱
- 帳戶 PHI :
  - 傳真號碼
  - 社會安全號碼 (SSN)
  - 健康保險受益人編號
  - 帳號 :
  - 證書或許可證號碼
- 車輛 PHI:
  - 車輛識別碼 (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) 地址編號

Amazon Transcribe Medical 是符合 1996 年健康保險流通與責任法案 (HIPAA) 資格的服務。如需詳細資訊，請參閱 [Amazon Transcribe 醫療](#)。如需識別音訊檔案中 PHI 的資訊，請參閱 [識別音訊檔案中的 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

- [識別音訊檔案中的 PHI](#)
- [識別即時串流中的 PHI](#)

## 識別音訊檔案中的 PHI

使用批次轉錄作業以轉錄音訊檔案，並識別其中的個人健康資訊 (PHI)。當您啟用個人健康資訊 (PHI) 識別時，Amazon Transcribe 醫療會標記在轉錄結果中識別的 PHI。如需 Amazon Transcribe Medical 可以識別之 PHI 的相關資訊，請參閱 [識別轉錄中的個人健康資訊 \(PHI\)](#)。

您可以使用 [StartMedicalTranscriptionJob](#) API 或 AWS 管理主控台，開始批次轉錄作業。

## AWS 管理主控台

若要使用 AWS 管理主控台 轉錄臨床醫生-患者對話，請建立轉錄任務，然後選擇音訊輸入類型的對話。

### 轉錄音訊檔案並識別 PHI (AWS 管理主控台)

1. 登入 [AWS 管理主控台](#)。
2. 在導覽窗格的 Amazon Transcribe Medical 下，選擇轉錄任務。
3. 選擇建立作業。
4. 在指定作業詳細資訊頁面的工作設定下，指定下列項目。
  - a. 名稱 – 您的唯一轉錄任務的名稱 AWS 帳戶。
  - b. 音訊輸入類型 — 對話或聽寫。
5. 針對其餘欄位，指定音訊檔案 Amazon S3 的位置，以及您要存放轉錄任務輸出的位置。
6. 選擇下一步。
7. 在音訊設定下，選擇 PHI 識別。
8. 選擇建立。

## API

### 使用批次轉錄作業 (API) 轉錄音訊檔案並識別 PHI

- 對於 [StartMedicalTranscriptionJob](#) API，請指定以下項目。
  - a. 對於 `MedicalTranscriptionJobName`，指定您 AWS 帳戶中唯一的名稱。
  - b. 對於 `LanguageCode`，指定與您音訊檔案中所說語言相對應的語言代碼。
  - c. 對於 `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`，請指定在音訊檔案 `PRIMARYCARE` 中說話的臨床醫生的醫療專科。
  - e. 對於 `Type`，指定 `CONVERSATION` 或 `DICTATION`。
  - f. 對於 `OutputBucketName`，指定 Amazon S3 儲存貯體以存放轉錄結果。

以下是使用適用於 Python (Boto3) 的 AWS SDK 轉錄音訊檔案並識別病患 PHI 的範例請求。

```
from __future__ import print_function
```

```
import time
import boto3
transcribe = boto3.client('transcribe')
job_name = "my-first-transcription-job"
job_uri = "s3://amzn-s3-demo-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 = 'amzn-s3-demo-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)
```

以下範例代碼顯示識別 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": [{
      "id": 0,
      "start_time": "0.0",
      "end_time": "0.37",
      "alternatives": [{
        "confidence": "0.9993",
```

```
        "content": "The"
    ]],
    "type": "pronunciation"
}, {
    "id": 1,
    "start_time": "0.37",
    "end_time": "0.44",
    "alternatives": [{
        "confidence": "0.9981",
        "content": "patient's"
    }],
    "type": "pronunciation"
}, {
    "id": 2,
    "start_time": "0.44",
    "end_time": "0.52",
    "alternatives": [{
        "confidence": "1.0",
        "content": "name"
    }],
    "type": "pronunciation"
}, {
    "id": 3,
    "start_time": "0.52",
    "end_time": "0.92",
    "alternatives": [{
        "confidence": "1.0",
        "content": "is"
    }],
    "type": "pronunciation"
}, {
    "id": 4,
    "start_time": "0.92",
    "end_time": "0.9989",
    "alternatives": [{
        "confidence": "1.0",
        "content": "Bertrand"
    }],
    "type": "pronunciation"
}, {
    "id": 5,
    "alternatives": [{
        "confidence": "0.0",
        "content": "."
    }]
```

```
    ]],
    "type": "punctuation"
  ]],
  "entities": [{
    "content": "Bertrand",
    "category": "PHI*-Personal*",
    "startTime": 0.92,
    "endTime": 1.2,
    "confidence": 0.9989
  }],
  "audio_segments": [
    {
      "id": 0,
      "transcript": "The patient's name is Bertrand.",
      "start_time": "0.0",
      "end_time": "0.9989",
      "items": [
        0,
        1,
        2,
        3,
        4,
        5
      ]
    }
  ]
},
"status": "COMPLETED"
}
```

## AWS CLI

使用批次轉錄作業，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://amzn-s3-demo-bucket/my-input-files/my-audio-file.flac" \  
--output-bucket-name amzn-s3-demo-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 串流中識別個人健康資訊 (PHI)。當您啟用 PHI 識別時，Amazon Transcribe 醫療會標記其在轉錄結果中識別的 PHI。如需 Amazon Transcribe Medical 可以識別之 PHI 的相關資訊，請參閱 [識別轉錄中的個人健康資訊 \(PHI\)](#)。

### 識別向麥克風說話時聽寫的 PHI

若要使用 AWS 管理主控台 轉錄麥克風挑選的語音並識別任何 PHI，請選擇口述做為音訊輸入類型、啟動串流，然後開始對電腦上的麥克風說話。

使用 識別口述中的 PHI AWS 管理主控台

1. 登入 [AWS 管理主控台](#)。
2. 在導覽窗格中，選擇即時轉錄。
3. 對於音訊輸入類型，選擇聽寫。
4. 對於其他設定，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。

### 識別 HTTP/2 串流中的 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，指定 en-US。
- 對於 MediaSampleHertz，指定音訊的取樣率。
- 對於 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，請使用下列格式以建立預先簽署的 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 操作通用的參數都列在 [通用參數](#) 區段中。

## 產生替代轉錄

當您使用 Amazon Transcribe Medical 時，您會取得可信度最高的轉錄。不過，您可以設定 Amazon Transcribe Medical 以傳回其他可信度較低的轉錄。

使用替代轉錄以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 管理主控台 或 [StartMedicalTranscriptionJob](#) API 產生替代轉錄。

### AWS 管理主控台

若要使用 AWS 管理主控台 產生替代轉錄，請在設定任務時啟用替代結果。

1. 登入 [AWS 管理主控台](#)。
2. 在導覽窗格的 Amazon Transcribe Medical 下，選擇轉錄任務。
3. 選擇建立作業。
4. 在指定作業詳細資訊頁面上，提供轉錄作業的相關資訊。

5. 選擇下一步。
6. 啟用替代結果。
7. 對於最大替代，輸入介於 2 到 10 之間的整數值，以取得您要在輸出中顯示的替代轉錄數量上限。
8. 選擇建立。

## API

使用批次轉錄作業，分隔音訊檔案中每位發言者的文字 (API)

- 對於 [StartMedicalTranscriptionJob](#) API，請指定以下項目。
  - a. 對於 `MedicalTranscriptionJobName`，請指定在 AWS 帳戶中唯一的名稱。
  - b. 對於 `LanguageCode`，請指定與音訊檔案中所說的語言對應的語言代碼，以及詞彙篩選語言對應的語言。
  - c. 在 `Media` 物件的 `MediaFileUri` 參數中，指定您要轉錄的音訊檔案位置。
  - d. 對於 `Specialty`，請指定在音訊檔案中說話的臨床醫生的醫療專科。
  - e. 對於 `Type`，指定要轉錄醫學對話或聽寫。
  - f. 對於 `OutputBucketName`，指定 Amazon S3 儲存貯體以存放轉錄結果。
  - 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，指示轉錄輸出中所需的替代轉錄數量。

下列請求使用 適用於 Python (Boto3) 的 AWS SDK 啟動轉錄任務，該任務最多會產生兩個替代轉錄。

```
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://amzn-s3-demo-bucket/my-input-files/my-audio-file.flac
transcribe.start_medical_transcription_job(
    MedicalTranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
```

```
OutputBucketName = 'amzn-s3-demo-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

在音訊檔案中轉錄初級護理臨床醫生和患者間對話的音訊檔案 (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 `example-start-command.json` 的內容。

```
{
    "MedicalTranscriptionJobName": "my-first-transcription-job",
    "LanguageCode": "en-US",
    "Specialty": "PRIMARYCARE",
    "Type": "CONVERSATION",
```

```
"OutputBucketName": "amzn-s3-demo-bucket",
"Media": {
  "MediaFileUri": "s3://amzn-s3-demo-bucket/my-input-files/my-audio-
file.flac"
},
"Settings": {
  "ShowAlternatives": true,
  "MaxAlternatives": 2
}
}
```

## Amazon Transcribe 醫療和界面 VPC 端點 (AWS PrivateLink)

您可以建立介面 VPC 端點，在 VPC 和 Amazon Transcribe Medical 之間建立私有連線。介面端點採用 [AWS PrivateLink](#) 技術，可讓您在沒有網際網路閘道、NAT 裝置、VPN 連線或 Direct Connect 連線的情況下私密存取 Amazon Transcribe Medical APIs。VPC 中的執行個體不需要公有 IP 地址，即可與 Amazon Transcribe Medical APIs 通訊。VPC 和 Amazon Transcribe Medical 之間的流量不會離開 Amazon 網路。

每個介面端點都是由您子網路中的一或多個[彈性網路介面](#)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 Medical VPC 端點的考量事項

在您設定 Amazon Transcribe Medical 的介面 VPC 端點之前，請務必檢閱 Amazon VPC 《使用者指南》中的[介面端點屬性和限制](#)。

Amazon Transcribe Medical 支援從您的 VPC 呼叫其所有 API 動作。

### 建立 Amazon Transcribe Medical 的介面 VPC 端點

您可以使用 AWS 管理主控台 或 為 Amazon Transcribe Medical 服務建立 VPC 端點 AWS CLI。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的[建立介面端點](#)。

對於 Amazon Transcribe Medical 中的批次轉錄，請使用下列服務名稱建立 VPC 端點：

- com.amazonaws.us-west-2.transcribe

對於 Amazon Transcribe Medical 中的串流轉錄，請使用下列服務名稱建立 VPC 端點：

- `com.amazonaws.us-west-2.transcribestreaming`

如果您為端點啟用私有 DNS，您可以使用其預設 DNS 名稱向 Amazon Transcribe Medical 提出 API 請求 AWS 區域，例如 `transcribestreaming.us-east-2.amazonaws.com`。

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的[透過介面端點存取服務](#)。

## 為 Amazon Transcribe Medical 串流建立 VPC 端點政策

您可以將端點政策連接至控制 Amazon Transcribe Medical 存取的 VPC 端點。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：Amazon Transcribe 醫療串流轉錄動作的 VPC 端點政策

以下是 Medical 中串流轉錄的端點政策範例 Amazon Transcribe。連接到端點時，此政策會授予所有資源上所有委託人的所列 Amazon Transcribe Medical 動作的存取權。

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Allow",
      "Action": [
        "transcribe:StartMedicalStreamTranscription",
      ],
      "Resource": "*"
    }
  ]
}
```

範例：Amazon Transcribe 醫療批次轉錄動作的 VPC 端點政策

以下是 Amazon Transcribe Medical 中批次轉錄的端點政策範例。連接到端點時，此政策會授予所有資源上所有委託人的所列 Amazon Transcribe Medical 動作的存取權。

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Allow",
      "Action": [
        "transcribe:StartMedicalTranscriptionJob"
      ],
      "Resource": "*"
    }
  ]
}
```

## 共用子網路

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 共用的相關資訊，請參閱 Amazon Virtual Private Cloud 指南中的[與其他帳戶共用您的 VPC](#)。

# AWS HealthScribe

AWS HealthScribe 是一種符合 HIPAA 資格的機器學習 (ML) 功能，結合了語音辨識和生成式 AI，可轉錄患者與臨床醫生對話並產生 easy-to-review 的臨床備註。AWS HealthScribe 可協助醫療保健軟體供應商建置臨床應用程式，以減少文件負擔並改善諮詢體驗。此服務會自動提供豐富的對話文字記錄、識別發言者角色、分類對話、擷取醫學術語，並產生初步臨床備註。AWS HealthScribe 結合這些功能，無需整合和最佳化個別 AI 服務，讓您加速實作。

常用案例：

- 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 產生的結果很概率，可能不一定會因為各種因素而準確，包括音訊品質、背景雜訊、發言者清晰度、醫學術語的複雜性、內容特定的語言細微差別、和 [機器學習和生成式 AI 的性質](#)。AWS HealthScribe 旨在用於臨床醫生和醫學骨架的輔助角色。AWS HealthScribe 輸出應僅用於患者護理案例，包括但不限於做為電子運作狀態記錄的一部分，經過訓練的醫療專業人員審核後，醫療判斷的準確性和實施。AWS HealthScribe 輸出不能取代專業醫療建議，診斷、或處理方式和並非旨在修復、處理、緩解、防止、或診斷任何疾病或健康狀況。

主題

- [安全](#)
- [服務可用性](#)
- [技術需求](#)
- [支援的醫療專科](#)
- [工作流程](#)
- [AWS HealthScribe 文字記錄檔案](#)
- [AWS HealthScribe 臨床文件檔案](#)

- [AWS HealthScribe 轉錄任務](#)
- [AWS HealthScribe 串流](#)
- [for AWS HealthScribe 靜態資料加密](#)

## 安全

AWS HealthScribe 在共同的責任模型下運作，因此 AWS 負責保護執行 AWS HealthScribe 的基礎設施，而且您負責管理您的資料。若要了解詳細資訊，請參閱 [共同責任模型](#)。

根據預設，AWS HealthScribe 提供靜態加密，以保護使用 Amazon S3 受管金鑰的敏感客戶資料。當您建立 AWS HealthScribe 轉錄任務或啟動串流時，您可以指定客戶受管金鑰。這會新增第二層加密。如需詳細資訊，請參閱 [for AWS HealthScribe 靜態資料加密](#)。

## 服務可用性

AWS HealthScribe 在美國東部（維吉尼亞北部）區域提供。

## 技術需求

- 支援的語言：美式英文 (en-US)
- 建議的音訊格式：無失真音訊（例如 FLAC 或 WAV）
- 編碼：PCM 16 位元
- 取樣率：16,000 Hz 或更高

## 支援的醫療專科

AWS HealthScribe 目前支援下列專業：

- 敏感免疫學
- 心臟科
- 皮膚病學
- 神經內科
- 消化道病學
- 血液/腫瘤學
- 感染性疾病

- Nephrology
- 神經病學
- OBGYN
- 腫瘤學
- 眼科學
- 骨科
- 耳鼻喉科
- 痛苦醫學
- 小兒科
- 初級護理
- 精神科
- 肺病學
- 風濕病學
- 外科
- 泌尿外科

## 工作流程

AWS HealthScribe 工作流程包括轉錄任務和串流。在您執行轉錄任務或完成串流後，AWS HealthScribe 會產生轉錄檔案，並提供每個對話回合的turn-by-turn轉錄輸出和洞見。此外，還會產生臨床文件檔案，其中包含摘要和證據連結。如需更多資訊，請參閱[AWS HealthScribe 文字記錄檔案](#)及[AWS HealthScribe 臨床文件檔案](#)。

- 轉錄任務 – 透過轉錄任務，AWS HealthScribe 會分析來自 Amazon S3 儲存貯體的已完成醫療諮詢媒體檔案。以下是 AWS HealthScribe 轉錄任務特定的 API 操作。
  - [StartMedicalScribeJob](#)
  - [ListMedicalScribeJobs](#)
  - [GetMedicalScribeJob](#)
  - [DeleteMedicalScribeJob](#)

如需詳細資訊，包括程式碼範例，請參閱 [AWS HealthScribe 轉錄任務](#)。

- 串流 – AWS HealthScribe 串流是一種以 HTTP2 為基礎的即時雙向服務，在一個頻道上接受音訊串流，並在另一個頻道上提供音訊轉錄。

以下是 AWS HealthScribe 串流特定的 API 操作：

- [StartMedicalScribeStream](#)
- [GetMedicalScribeStream](#)

如需詳細資訊，包括程式碼範例，請參閱 [AWS HealthScribe 串流](#)。

## AWS HealthScribe 文字記錄檔案

在文字記錄檔案中，除了具有單字層級時間戳記的標準turn-by-turn轉錄輸出之外，AWS HealthScribe 還為您提供：

- 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。

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\_0、CLINICIAN\_1 和 PATIENT\_0、PATIENT\_1。
- 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。
  - 造訪流程管理 — 與閒聊或轉換相關的資訊。
- 洞見 — 擷取存在於對話中的臨床相關實體 (ClinicalEntity)。AWS HealthScribe 會偵測 [Amazon Comprehend Medical](#) 支援的所有臨床實體。

如需轉錄任務的文字記錄範例，請參閱 中的文字記錄輸出 [轉錄任務輸出範例](#)。如需來自串流文字記錄範例，請參閱 中的文字記錄輸出 [串流轉錄輸出範例](#)。

# AWS HealthScribe 臨床文件檔案

AWS HealthScribe 可以使用下列其中一個範本進行臨床備註摘要。預設值為 HISTORY\_AND\_PHYSICAL。

- HISTORY\_AND\_PHYSICAL：提供臨床文件重要章節的摘要。區段的範例包括主投訴、目前疾病歷史記錄、系統審查、過去醫療歷史記錄、評估和計劃。
- GIRPP：根據患者朝向目標的進度提供摘要。區段的範例包括目標、介入、回應、進度和計劃。
- BIRP：專注於病患的行為模式和回應。區段的範例包括行為、介入、回應和計劃。
- SIRP：強調治療的情況內容。區段的範例包括情況、介入、回應和計劃。
- DAP：提供臨床文件的簡化格式。區段的範例包括資料、評估和計劃。
- BH\_SOAP：以行為狀態為重心的文件格式。區段的範例包括主觀、目標、評估和計畫。
- PH\_SOAP：以身體運作狀態為重心的文件格式。區段的範例包括主觀、目標、評估和計畫。

若要指定要使用的範本，請執行下列動作：

- 對於轉錄任務，指定要在 [StartMedicalScribeJob](#) API 操作 NoteTemplate 之的 [ClinicalNoteGenerationSettings](#) Settings 的 中使用的範本。
- 對於串流，您可以指定要在 [MedicalScribeConfigurationEvent](#) NoteTemplate 之的 [ClinicalNoteGenerationSettings](#) PostStreamAnalyticsSettings 的 中使用的範本。

## Medical Scribe 內容

MedicalScribeContext 物件包含內容資訊，用於自訂具有患者特定詳細資訊的臨床備註產生。

MedicalScribeContext 物件包含下列元件：

- PatientContext - 包含用於自訂臨床備註產生的病患特定資訊。其中包含：
  - 代名詞 - 您想要提供的患者偏好代名詞，做為產生臨床備註的內容。在產生的臨床輸出中參照病患時，會使用這些代名詞。

如需如何設定醫療轉錄內容的資訊，請參閱 [MedicalScribeContext](#)。

### 主題

- [HISTORY\\_AND\\_PHYSICAL 範本區段](#)

- [GIRPP 範本區段](#)
- [BIRP 範本區段](#)
- [SIRP 範本區段](#)
- [DAP 範本區段](#)
- [BH\\_SOAP 範本區段](#)
- [PH\\_SOAP 範本區段](#)

## HISTORY\_AND\_PHYSICAL 範本區段

HISTORY\_AND\_PHYSICAL 洞見範本包含下列區段。

區段	Description
主訴	病患造訪臨床醫生原因的簡短描述。
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。
PHYSICAL_EXAMINATION	記錄臨床醫生對患者身體系統和生命體徵進行理學檢查的結果。
PAST_FAMILY_HISTORY	有關在患者系列中執行之運作狀態的資訊。
PAST_SOCIAL_HISTORY	有關患者社交生活、習慣、職業和影響健康的環境因素的詳細資訊。

區段	Description
DIAGNOSTIC_TESTING	實驗室測試、影像研究和其他診斷程序的結果和解釋。

中出現的每個句子都Summary包含 EvidenceLinks ，可在摘要的文字記錄中提供SegmentId相關對話的。這有助於使用者驗證應用程式中摘要的準確性。如同可解釋性，為 AI 產生的洞見提供可追蹤性和透明度，與負責任的 AI 原則一致。將這些參考與摘要備註一起提供給臨床醫生或醫學人員，有助於培養信任，並鼓勵在臨床環境中安全地使用 AI。

如需轉錄任務臨床文件檔案的範例，請參閱 中的臨床文件檔案範例 [轉錄任務輸出範例](#)。如需來自串流之臨床文件檔案的範例，請參閱 中的臨床文件檔案範例 [串流轉錄輸出範例](#)。

## GIRPP 範本區段

GIRPP 洞見範本包含下列各節。

區段	Description
目標	需要透過處理解決的已識別問題、挑戰或行為。
介入	臨床醫生用來協助患者解決已識別目標的特定處理、方法或技術。
回應	患者如何回應介入，包括他們的參與程度、反應和意見回饋。
進度	臨床醫生對朝治療目標移動的評估，包括觀察患者改善或障礙。
計畫	接下來的處理步驟，包括未來的介入、家庭作業指派和推薦。

## BIRP 範本區段

BIRP 洞見範本包含下列各節。

區段	Description
Behavior (行為)	患者呈現的問題及其對治療的反應。
介入	臨床醫生用來協助患者解決已識別目標的特定處理、方法或技術。
回應	患者如何回應介入，包括他們的參與程度、反應和意見回饋。
計畫	接下來的處理步驟，包括未來的介入、家庭作業指派和推薦。

## SIRP 範本區段

SIRP 洞見範本包含下列各節。

區段	Description
情形	患者提出的問題及其尋求治療的目標。
介入	臨床醫生用來協助患者解決已識別目標的特定處理、方法或技術。
回應	患者如何回應介入，包括他們的參與程度、反應和意見回饋。
計畫	接下來的處理步驟，包括未來的介入、家庭作業指派和推薦。

## DAP 範本區段

DAP 洞見範本包含下列區段。

區段	Description
資料	病患尋求治療的原因和有關病患的資訊。

區段	Description
評估	臨床醫生對患者情況的診斷。
計畫	接下來的處理步驟，包括未來的介入、家庭作業指派和推薦。

## BH\_SOAP 範本區段

BH\_SOAP 洞見範本包含下列區段。

區段	Description
主觀	用戶端的處理目標及其體驗，以及現有和過去的問題。
目標	有關用戶端的資料和事實。
評估	臨床醫生對患者情況的診斷。
計畫	接下來的處理步驟，包括未來的介入、家庭作業指派和推薦。

## PH\_SOAP 範本區段

PH\_SOAP 洞察範本包含下列區段。

區段	Description
主觀	用戶端的處理目標及其體驗，以及現有和過去的問題。
目標	有關用戶端的資料和事實。
評估	臨床醫生對患者情況的診斷。

區段	Description
計畫	接下來的處理步驟，包括未來的介入、家庭作業指派和推薦。

## AWS HealthScribe 轉錄任務

An AWS HealthScribe 轉錄任務會處理來自 Amazon S3 儲存貯體的媒體檔案。處理媒體檔案時，它會轉錄患者與臨床醫師的對話，並分析醫療諮詢以產生兩個 JSON 輸出檔案：[文字記錄檔案](#)和[臨床文件檔案](#)。

以下是 AWS HealthScribe 轉錄任務特有的 API 操作：

- [StartMedicalScribeJob](#)
- [ListMedicalScribeJobs](#)
- [GetMedicalScribeJob](#)
- [DeleteMedicalScribeJob](#)

## 啟動 AWS HealthScribe 轉錄任務

您可以使用 CLI 或 AWS SDKs 啟動 AWS HealthScribe AWS 任務。

### 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://amzn-s3-demo-bucket/my-input-files/my-media-file.flac \  
--output-bucket-name amzn-s3-demo-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://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"  
  },  
  "OutputBucketName": "amzn-s3-demo-bucket",  
  "DataAccessRoleArn": "arn:aws:iam::111122223333:role/ExampleRole",  
  "Settings": {  
    "ShowSpeakerLabels": false,  
    "ChannelIdentification": true  
  },  
  "ChannelDefinitions": [  
    {  
      "ChannelId": 0,  
      "ParticipantRole": "CLINICIAN"  
    }, {  
      "ChannelId": 1,  
      "ParticipantRole": "PATIENT"  
    }  
  ]  
}
```

適用於 Python (Boto3) 的 AWS SDK

下列範例使用 適用於 Python (Boto3) 的 AWS SDK 提出 [start\\_Medical\\_scribe\\_job](#) 請求。如需詳細資訊，請參閱 [StartMedicalScribeJob](#)。

```
from __future__ import print_functionimport timeimport boto3  
transcribe = boto3.client('transcribe', 'us-west-2')  
job_name = "my-first-medical-scribe-job"  
job_uri = "s3://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
```

```
transcribe.start_medical_scribe_job(  
    MedicalScribeJobName = job_name,  
    Media = {  
        'MediaFileUri': job_uri  
    },  
    OutputBucketName = 'amzn-s3-demo-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 管理主控台目前不支援 AWS HealthScribe 任務。

## 轉錄任務輸出範例

除了文字記錄之外，StartMedicalScribeJob 請求還會產生獨立的臨床文件檔案。這兩個檔案都是 JSON 格式，並存放在您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

An AWS HealthScribe 轉錄檔案（來自StartMedicalScribeJob請求）的格式如下：

```
{
  "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": 0.17,
    "EndAudioTime": 0.9,
    "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://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
  },
  "OutputBucketName": "amzn-s3-demo-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"
        }
      ],
      "SummarizedSegment": "Weight loss"
    }
  ]
},
{
  "SectionName": "PLAN",
  "Summary": [
    {
      "EvidenceLinks": [
        {
          "SegmentId": "uuid31"
        },
        {
          "SegmentId": "uuid32"
        },
        {
          "SegmentId": "uuid33"
        },
        {
          "SegmentId": "uuid34"
        }
      ],
    }
  ],
}
```

```
        "SummarizedSegment": "For the condition of Weight loss: The patient was  
        given a 30-day supply of phentermine and was advised to follow up in 30 days."  
    }  
  ]  
}  
]  
}
```

## AWS HealthScribe 串流

透過 AWS HealthScribe 串流，您可以即時轉錄醫療對話。AWS HealthScribe 串流是一種以 HTTP2 為基礎的即時雙向服務，在一個頻道上接受音訊串流，並在另一個頻道上提供音訊轉錄。串流完成後，AWS HealthScribe 會分析串流內容，並產生文字記錄 JSON 檔案和臨床備註 JSON 檔案。

若要開始串流，請使用 [StartMedicalScribeStream](#) API 操作。此 API 會啟動 HTTP2 型雙向頻道，讓您用來串流音訊事件。

當您啟動串流時，請先在 中指定串流組態 `MedicalScribeConfigurationEvent`。此事件包括頻道定義、加密設定和串流後分析設定，例如彙總文字記錄和產生臨床備註的輸出組態。

開始串流音訊後，您可以管理串流，如下所示：

- 完成後，若要使用串流後分析開始處理結果，請傳送 `MedicalScribeSessionControlEvent` 具有 `Type` 的 `END_OF_SESSION`，AWS HealthScribe 會開始分析。
- 若要暫停串流，請完成輸入串流，而不傳送 `MedicalScribeSessionControlEvent`。
- 若要繼續暫停的串流，請使用 `StartMedicalScribeStream` API 操作並指定相同的 `SessionId`。這是您最初啟動串流 `SessionId` 時使用的。

### 主題

- [準則和要求](#)
- [ResourceAccessRoleArn 角色許可](#)
- [Starting AWS HealthScribe 串流轉錄](#)

## 準則和要求

以下是 AWS HealthScribe 串流的指導方針和要求：

- 傳送音訊事件之前，您必須先在 中指定串流組態 [MedicalScribeConfigurationEvent](#)。
- 若要執行串流後分析，ResourceAccessRoleArn 中的 MedicalScribeConfigurationEvent 必須具有正確的許可。如需詳細資訊，請參閱 [ResourceAccessRoleArn 角色許可](#)。
- 您可以在初始串流建立後 5 小時內繼續工作階段任意次數。
- 您可以跨所有串流請求，透過工作階段串流最多 2 小時的音訊。
- 根據預設，AWS HealthScribe 提供靜態加密，以保護使用 Amazon S3 受管金鑰的敏感客戶資料。當您啟動串流時，您可以為第二層加密指定 AWS KMS 金鑰。您的 ResourceAccessRoleArn 必須具有使用 AWS KMS 金鑰的許可。如需詳細資訊，請參閱 [for AWS HealthScribe 靜態資料加密](#)。
- 您可以使用 AWS HealthScribe 串流搭配 AWS SDKs，不包括適用於 Python (Boto3) 的 SDK 和適用於 PHP 的 SDK。
- 如果在結束串流後發生 LimitExceededException 例外狀況，您可以重新啟動工作階段，但仍會產生串流後分析。若要重新啟動串流，請使用 [StartMedicalScribeStream](#) API，並使用相同的 SessionID。然後傳送 MedicalScribeSessionControlEvent 具有 Type 的 END\_OF\_SESSION，AWS HealthScribe 會開始分析。

## ResourceAccessRoleArn 角色許可

若要執行串流後分析，ResourceAccessRoleArn 中的 MedicalScribeConfigurationEvent 必須能夠存取您的 Amazon S3 輸出儲存貯體，而且如果您提供它，則必須能夠存取您的 AWS KMS 金鑰。此外，角色的信任政策必須授予 transcribe.streaming.amazonaws.com 服務擔任角色的許可。

以下是授予 Amazon S3 儲存貯體許可和 AWS KMS 金鑰許可的 IAM 政策範例。如需詳細資訊，請參閱 [for AWS HealthScribe 靜態資料加密](#)。

### JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket",
```

```

        "arn:aws:s3:::amzn-s3-demo-bucket/*"
    ],
    "Effect": "Allow"
  },
  {
    "Action": [
      "kms:DescribeKey",
      "kms:Decrypt",
      "kms:Encrypt",
      "kms:GenerateDataKey*"
    ],
    "Resource": "arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
    "Effect": "Allow"
  }
]
}

```

信任政策範例如下。

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "transcribe.streaming.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}

```

## Starting AWS HealthScribe 串流轉錄

下列程式碼範例示範如何使用 AWS SDKs 設定 AWS HealthScribe 串流轉錄。

## 主題

- [適用於 Java 2.x 的 SDK](#)
- [串流轉錄輸出範例](#)

## 適用於 Java 2.x 的 SDK

下列範例使用適用於 Java 的 SDK 2.x 來設定串流並提出 [StartMedicalScribeStream](#) 請求。

```
package org.example;

import io.reactivex.rxjava3.core.BackpressureStrategy;
import io.reactivex.rxjava3.core.Flowable;
import org.reactivestreams.Publisher;
import org.reactivestreams.Subscriber;
import software.amazon.awssdk.auth.credentials.AwsCredentialsProvider;
import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
import software.amazon.awssdk.core.SdkBytes;
import software.amazon.awssdk.regions.Region;
import
    software.amazon.awssdk.services.transcribestreaming.TranscribeStreamingAsyncClient;
import
    software.amazon.awssdk.services.transcribestreaming.model.ClinicalNoteGenerationSettings;
import software.amazon.awssdk.services.transcribestreaming.model.LanguageCode;
import software.amazon.awssdk.services.transcribestreaming.model.MediaEncoding;

import
    software.amazon.awssdk.services.transcribestreaming.model.MedicalScribeInputStream;
import
    software.amazon.awssdk.services.transcribestreaming.model.MedicalScribePostStreamAnalyticsSettings;
import
    software.amazon.awssdk.services.transcribestreaming.model.MedicalScribeSessionControlEventType;
import
    software.amazon.awssdk.services.transcribestreaming.model.MedicalScribeTranscriptEvent;
import
    software.amazon.awssdk.services.transcribestreaming.model.MedicalScribeTranscriptSegment;
import
    software.amazon.awssdk.services.transcribestreaming.model.StartMedicalScribeStreamRequest;
import
    software.amazon.awssdk.services.transcribestreaming.model.StartMedicalScribeStreamResponseHandler;
import
    software.amazon.awssdk.services.transcribestreaming.model.medicalscribeinputstream.DefaultConf
```

```
import software.amazon.awssdk.http.nio.netty.NettyNioAsyncHttpClient;

import javax.sound.sampled.AudioFormat;
import javax.sound.sampled.AudioInputStream;
import javax.sound.sampled.AudioSystem;
import javax.sound.sampled.DataLine;
import javax.sound.sampled.LineUnavailableException;
import javax.sound.sampled.TargetDataLine;
import java.io.BufferedInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.UncheckedIOException;
import java.util.Arrays;
import java.util.concurrent.CompletableFuture;

public class HealthScribeStreamingDemoApp {
    private static final int CHUNK_SIZE_IN_BYTES = 6400;
    private static final int SAMPLE_RATE = 16000;
    private static final Region REGION = Region.US_EAST_1;
    private static final String sessionId = "1234abcd-12ab-34cd-56ef-123456SAMPLE";
    private static final String bucketName = "amzn-s3-demo-bucket";
    private static final String resourceAccessRoleArn =
"arn:aws:iam::123456789012:role/resource-access-role";
    private static TranscribeStreamingAsyncClient client;

    public static void main(String args[]) {

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

        try {
            StartMedicalScribeStreamRequest request =
StartMedicalScribeStreamRequest.builder()
                .languageCode(LanguageCode.EN_US.toString())
                .mediaSampleRateHertz(SAMPLE_RATE)
                .mediaEncoding(MediaEncoding.PCM.toString())
                .sessionId(sessionId)
                .build();

            MedicalScribeInputStream endSessionEvent =
MedicalScribeInputStream.sessionControlEventBuilder()
```

```
        .type(MedicalScribeSessionControlEventType.END_OF_SESSION)
        .build();

        CompletableFuture<Void> result = client.startMedicalScribeStream(
            request,
            new AudioStreamPublisher(getStreamFromMic(),
getConfigurationEvent(),endSessionEvent),
            getMedicalScribeResponseHandler());
        result.get();
        client.close();
    } catch (Exception e) {
        System.err.println("Error occurred: " + e.getMessage());
        e.printStackTrace();
    }
}

private static AudioInputStream getStreamFromMic() throws LineUnavailableException
{
    // Signed PCM AudioFormat with 16kHz, 16 bit sample size, mono
    AudioFormat format = new AudioFormat(SAMPLE_RATE, 16, 1, true, false);
    DataLine.Info info = new DataLine.Info(TargetDataLine.class, format);

    if (!AudioSystem.isLineSupported(info)) {
        System.out.println("Line not supported");
        throw new LineUnavailableException("The audio system microphone line is not
supported.");
    }
    TargetDataLine line = (TargetDataLine) AudioSystem.getLine(info);
    int bufferSize = (CHUNK_SIZE_IN_BYTES / format.getFrameSize()) *
format.getFrameSize();
    line.open(format);
    line.start();

    // Create a wrapper class that can be closed when Enter is pressed
    AudioInputStream audioStream = new AudioInputStream(line);

    // Start a thread to monitor for Enter key
    System.out.println("Recording... Press Enter to stop");
    Thread monitorThread = new Thread(() -> {
        try {
            System.in.read();
            line.stop();
            line.close();
        } catch (IOException e) {
```

```

        e.printStackTrace();
    }
});
monitorThread.setDaemon(true); // Set as daemon thread so it doesn't prevent
JVM shutdown
monitorThread.start();

return new AudioInputStream(
    new BufferedInputStream(new AudioInputStream(line)),
    format,
    AudioSystem.NOT_SPECIFIED
);
}

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

private static StartMedicalScribeStreamResponseHandler
getMedicalScribeResponseHandler() {

    return StartMedicalScribeStreamResponseHandler.builder()
        .onResponse(r -> {
            System.out.println("Received Initial response");
        })
        .onError(Throwable::printStackTrace)
        .onComplete(() -> {
            System.out.println("=== All records streamed successfully ===");
        })
        .subscriber(event -> {
            if (event instanceof MedicalScribeTranscriptEvent) {
                MedicalScribeTranscriptSegment segment =
                ((MedicalScribeTranscriptEvent) event).transcriptSegment();
                if (segment != null && segment.content() != null && !
segment.content().isEmpty()) {
                    System.out.println(segment.content());
                }
            }
        })
        .build();
}

private static DefaultConfigurationEvent getConfigurationEvent() {

```

```

        MedicalScribePostStreamAnalyticsSettings postStreamSettings =
MedicalScribePostStreamAnalyticsSettings
    .builder()
    .clinicalNoteGenerationSettings(
        ClinicalNoteGenerationSettings.builder()
            .outputBucketName(bucketName)
            .build()
        )
    .build();
    return (DefaultConfigurationEvent)
MedicalScribeInputStream.configurationEventBuilder()
    .resourceAccessRoleArn(resourceAccessRoleArn)
    .postStreamAnalyticsSettings(postStreamSettings)
    .build();
}

private static class AudioStreamPublisher implements
Publisher<MedicalScribeInputStream> {
    private final InputStream audioInputStream;
    private final MedicalScribeInputStream configEvent;
    private final MedicalScribeInputStream endSessionEvent;

    private AudioStreamPublisher(AudioInputStream audioInputStream,
        MedicalScribeInputStream configEvent,
        MedicalScribeInputStream endSessionEvent) {
        this.audioInputStream = audioInputStream;
        this.configEvent = configEvent;
        this.endSessionEvent = endSessionEvent;
    }

    @Override
    public void subscribe(Subscriber<? super MedicalScribeInputStream> subscriber)
    {
        createAudioFlowable()
            .doOnComplete(() -> {
                try {
                    audioInputStream.close();
                } catch (IOException e) {
                    throw new UncheckedIOException(e);
                }
            })
            .subscribe(subscriber);
    }
}

```

```
private Flowable<MedicalScribeInputStream> createAudioFlowable() {
    // Start with config event
    Flowable<MedicalScribeInputStream> configFlow = Flowable.just(configEvent);

    // Create audio chunk flowable
    Flowable<MedicalScribeInputStream> audioFlow = Flowable.create(emitter -> {
        byte[] buffer = new byte[CHUNK_SIZE_IN_BYTES];
        int bytesRead;

        try {
            while (!emitter.isCancelled() && (bytesRead =
audioInputStream.read(buffer)) > 0) {
                byte[] audioData = bytesRead < buffer.length
                    ? Arrays.copyOfRange(buffer, 0, bytesRead)
                    : buffer;

                MedicalScribeInputStream audioEvent =
MedicalScribeInputStream.audioEventBuilder()
                    .audioChunk(SdkBytes.fromByteArray(audioData))
                    .build();

                emitter.onNext(audioEvent);
            }
            emitter.onComplete();
        } catch (IOException e) {
            emitter.onError(e);
        }
    }, BackpressureStrategy.BUFFER);

    // End with session end event
    Flowable<MedicalScribeInputStream> endFlow =
Flowable.just(endSessionEvent);

    // Concatenate all flows
    return Flowable.concat(configFlow, audioFlow, endFlow);
}
}
```

## 串流轉錄輸出範例

串流完成後，AWS HealthScribe 會分析串流內容，並產生文字記錄 JSON 檔案和臨床備註 JSON 檔案。下列是每個輸出類型的範例：

## 範例文字記錄輸出

以下是來自串流工作階段的 a AWS HealthScribe 文字記錄檔案範例。

```
{
  "Conversation": {
    "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"
  }],

```

```
"ConversationId": "sampleConversationUUID",
"LanguageCode": "en-US",
"SessionId": "sampleSessionUUID",
"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": "uuid4"
  }, {
    "BeginAudioTime": 6.96,
    "Content": "Yes.",
```

```

        "EndAudioTime": 7.88,
        "ParticipantDetails": {
            "ParticipantRole": "CLINICIAN_0"
        },
        "SectionDetails": {
            "SectionName": "SUBJECTIVE"
        },
        "SegmentId": "uuid5"
    }, {
        "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": "uuid6"
    }
}
}
}

```

## 範例臨床文件輸出

以下是來自串流工作階段的 a AWS HealthScribe 臨床文件洞察檔案範例。

```

{
  "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"
        }
      ],
      "SummarizedSegment": "Weight loss"
    }
  ]
},
{
  "SectionName": "PLAN",
  "Summary": [
    {
      "EvidenceLinks": [
        {
          "SegmentId": "uuid31"
        },
        {
          "SegmentId": "uuid32"
        },
        {
          "SegmentId": "uuid33"
        },
        {
          "SegmentId": "uuid34"
        }
      ]
    }
  ]
}
```

```
    }
  ],
  "SummarizedSegment": "For the condition of Weight loss: The patient was
given a 30-day supply of phentermine and was advised to follow up in 30 days."
}
]
}
],
"SessionId": "sampleSessionUUID"
}
}
```

## for AWS HealthScribe 靜態資料加密

根據預設，AWS HealthScribe 提供靜態加密，以保護使用 AWS HealthScribe 受管 AWS Key Management Service (AWS KMS) 金鑰的敏感客戶資料。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。此外，它還可讓您建置符合嚴格加密合規和法規要求的安全應用程式。當您建立 AWS HealthScribe 轉錄任務或啟動串流時，您可以指定客戶受管金鑰。這會新增第二層加密。

- AWS HealthScribe 受管 AWS KMS 金鑰 — AWS HealthScribe 預設會使用 AWS HealthScribe 受管 AWS Key Management Service (AWS KMS) 金鑰自動加密中繼檔案。您無法停用此加密層，或選擇替代加密類型。您無法檢視、管理或使用金鑰，或稽核其使用方式。不過，您不需要採取任何動作或變更任何程式，即可保護加密您資料的金鑰。
- 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受管金鑰，以透過現有的 AWS 擁有的加密新增第二層加密。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 政策和授予
  - 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 會使用擁有 AWS 的金鑰自動啟用靜態加密，免費保護個人身分識別資料。不過，使用客戶受管金鑰會產生 AWS KMS 費用。如需定價的詳細資訊，請參閱 [AWS Key Management Service 定價](#)。

如需詳細資訊 AWS KMS，請參閱 [什麼是 AWS Key Management Service](#)。

**主題**

- [建立客戶受管金鑰](#)
- [指定 AWS HealthScribe 的客戶受管金鑰](#)
- [AWS KMS 加密內容](#)
- [監控 AWS HealthScribe 的加密金鑰](#)

## 建立客戶受管金鑰

您可以使用 AWS 管理主控台或 AWS KMS APIs 來建立對稱客戶受管金鑰。若要建立對稱客戶受管金鑰，請遵循《AWS Key Management Service 開發人員指南》中的 [建立對稱客戶受管金鑰](#) 的步驟。

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 [管理對客戶受管金鑰的存取](#)。

## AWS KMS AWS HealthScribe 的關鍵政策

如果您在與 [StartMedicalScribeJob](#) 或 ResourceAccessRole [StartMedicalScribeStream](#) 請求DataAccessRole中指定為 IAM 的角色相同的帳戶中使用金鑰，則不需要更新金鑰政策。若要在與 DataAccessRole（用於轉錄任務）或 ResourceAccessRole（用於串流）不同的帳戶中使用客戶受管金鑰，您必須信任金鑰政策中的個別角色，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](#) — 提供客戶受管金鑰詳細資訊，以允許 AWS HealthScribe 驗證金鑰

以下是可用來授予 ResourceAccessRole 跨帳戶許可的金鑰政策範例，以使用客戶受管金鑰進行 AWS HealthScribe 串流。若要將此政策用於轉錄任務，請更新 Principal 以使用 DataAccessRole ARN，並移除或修改加密內容。

## JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAccessForKeyAdministrators",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:root"
      },
      "Action": [
        "kms:*"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowAccessToResourceAccessRoleForMedicalScribe",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/ResourceAccessRole"
      },
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:GenerateDataKey*"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:EncryptionContext:aws:us-east-1:transcribe:medical-scribe:session-id": "1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
      }
    },
    {
      "Sid": "AllowAccessToResourceAccessRoleForDescribeKey",
      "Effect": "Allow",
```

```

    "Principal": {
      "AWS": "arn:aws:iam::111122223333:role/ResourceAccessRole"
    },
    "Action": "kms:DescribeKey",
    "Resource": "*"
  }
]
}

```

## 存取角色的 IAM 政策許可

連接至 `DataAccessRole` 或 `ResourceAccessRole` 的 IAM 政策必須授予執行必要 AWS KMS 動作的許可，無論客戶受管金鑰和角色是否位於相同或不同的帳戶中。此外，角色的信任政策必須授予 AWS HealthScribe 擔任角色的許可。

下列 IAM 政策範例顯示如何授予 AWS HealthScribe 串流的 `ResourceAccessRole` 許可。若要將此政策用於轉錄任務，請將取

代 `transcribe.streaming.amazonaws.com` 為 `transcribe.streaming.amazonaws.com` 並移除或修改加密內容。

## JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:GenerateDataKey*"
      ],
      "Resource": "arn:aws:kms:us-west-2:111122223333:key/KMS-Example-KeyId",
      "Condition": {
        "StringEquals": {
          "kms:ViaService": "transcribe.streaming.amazonaws.com",
          "kms:EncryptionContext:aws:us-east-1:transcribe:medical-scribe:session-id": "1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
      }
    }
  ]
}

```

```

    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:DescribeKey"
      ],
      "Resource": "arn:aws:kms:us-west-2:111122223333:key/KMS-Example-KeyId",
      "Condition": {
        "StringEquals": {
          "kms:ViaService": "transcribe.streaming.amazonaws.com"
        }
      }
    }
  ]
}

```

以下是 ResourceAccessRole 的信任政策範例。對於 DataAccessRole，將取代 `transcribe.streaming.amazonaws.com` 為 `transcribe.amazonaws.com`。

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "transcribe.streaming.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "111122223333"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:transcribe:us-west-2:111122223333:*"
        }
      }
    }
  ]
}

```

```
]
}
```

如需在[政策中指定許可](#)或對[金鑰存取進行疑難排解](#)的詳細資訊，請參閱 AWS Key Management Service 開發人員指南。

## 指定 AWS HealthScribe 的客戶受管金鑰

您可以將客戶受管金鑰指定為轉錄任務或串流的第二層加密。

- 對於轉錄任務，您可以在 [StartMedicalScribeJob](#) API 操作的 [OutputEncryptionKMSKeyId](#) 中指定金鑰。
- 對於串流，您可以在 `MedicalScribeConfigurationEvent` 的 [MedicalScribeEncryptionSettings](#) 中指定金鑰。 [MedicalScribeConfigurationEvent](#)

## AWS KMS 加密內容

AWS KMS 加密內容是純文字、非私密金鑰：值對的映射。此映射代表額外的已驗證資料，稱為加密內容對，可為您的資料提供多一層的安全性。AWS HealthScribe 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。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。加密內容不是秘密，它在 CloudTrail 日誌中以純文字顯示（因此您可以使用它來識別和分類密碼編譯操作）。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](#) 參數必須參考對稱 AWS KMS 金鑰 ID。

對於串流，您可以在 [MedicalScribeConfigurationEvent](#) `KmsEncryptionContext` 的 [MedicalScribeEncryptionSettings](#) 中指定的鍵值對。 [MedicalScribeEncryptionSettings](#) 您可以使用 [AWS KMS 條件金鑰](#) 搭配 IAM 政策，根據 [密碼編譯操作](#) 請求中使用的加密內容來控制對對稱加密 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](#)。

## AWS HealthScribe 加密內容

AWS HealthScribe 在所有 AWS Key Management Service 密碼編譯操作中使用相同的加密內容。加密內容是字串對字串的映射，可以自訂為您想要的任何項目。

```
"encryptionContext": {
  "ECKey": "ECValue"
  ...
}
```

對於 AWS HealthScribe 串流，以下是服務產生的預設加密內容。它會將此內容套用到您提供的任何加密內容上。

```
"encryptionContext": {
  "aws:<region>:transcribe:medical-scribe:session-id":
  "1234abcd-12ab-34cd-56ef-123456SAMPLE"
}
```

對於 AWS HealthScribe 轉錄任務，以下是服務產生的預設加密內容。它會將此內容套用到您提供的任何加密內容上。

```
"encryptionContext": {
  "aws:<region>:transcribe:medical-scribe:job-name": "<job-name>",
  "aws:<region>:transcribe:medical-scribe:start-time-epoch-ms": "<job-start-time>"
}
```

如果您不提供任何加密內容，則所有 AWS KMS 密碼編譯操作只會使用服務產生的加密內容。

### 使用加密內容監控 AWS HealthScribe

當您使用對稱客戶受管金鑰來加密靜態 in AWS HealthScribe 中的資料時，您也可以使用稽核記錄和日誌中的加密內容來識別客戶受管金鑰的使用方式。加密內容也會出現在 AWS CloudTrail 或 CloudWatch Logs 產生的日誌中。

### 使用加密內容控制對客戶受管金鑰的存取

您也可以在金鑰政策和 IAM 政策中，使用加密內容來控制對對稱客戶受管金鑰的存取。

以下是授予特定加密內容之客戶受管金鑰存取權的金鑰政策陳述式範例。此政策陳述式中的 條件要求 KMS 金鑰使用具有指定加密內容的加密內容限制條件。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAccessToResourceAccessRoleForMedicalScribe",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/ResourceAccessRole"
      },
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:GenerateDataKey*"
      ],
      "Resource": "arn:aws:kms:us-west-2:111122223333:key/KMS-Example-  
KeyId",
      "Condition": {
        "StringEquals": {
          "kms:EncryptionContext:aws:us-east-1:transcribe:medical-  
scribe:session-id": "1234abcd-12ab-34cd-56ef-123456SAMPLE",
          "kms:EncryptionContext:EKKey": "ECValue"
        }
      }
    },
    {
      "Sid": "AllowAccessToResourceAccessRoleForDescribeKey",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/ResourceAccessRole"
      },
      "Action": "kms:DescribeKey",
      "Resource": "arn:aws:kms:us-west-2:111122223333:key/KMS-Example-  
KeyId"
    }
  ]
}
```

## 監控 AWS HealthScribe 的加密金鑰

當您搭配 AWS HealthScribe 使用 AWS Key Management Service 客戶受管金鑰時，您可以使用 AWS CloudTrail 或 CloudWatch 日誌來追蹤 AWS HealthScribe 傳送的請求 AWS KMS。

下列範例是 CloudTrail Encrypt 和 Decrypt 事件，您可以使用這些事件來監控 AWS HealthScribe 如何使用您的客戶受管金鑰。

### 加密

```
{
  "eventVersion": "1.09",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
    "arn": "arn:aws:sts::123456789012:assumed-role/Admin/Sampleuser01",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::123456789012:assumed-role/Admin/Sampleuser01",
        "accountId": "123456789012",
        "userName": "Admin"
      }
    },
    "attributes": {
      "creationDate": "2024-08-16T01:10:05Z",
      "mfaAuthenticated": "false"
    }
  },
  "invokedBy": "transcribe.streaming.amazonaws.com"
},
"eventTime": "2024-08-16T01:10:05Z",
"eventSource": "kms.amazonaws.com",
"eventName": "Encrypt",
"awsRegion": "us-east-1",
"sourceIPAddress": "transcribe.streaming.amazonaws.com",
"userAgent": "transcribe.streaming.amazonaws.com",
"requestParameters": {
  "encryptionContext": {
    "aws:us-east-1:transcribe:medical-scribe:session-id": "1234abcd-12ab-34cd-56ef-123456SAMPLE"
  }
}
```

```

    },
    "encryptionAlgorithm":"SYMMETRIC_DEFAULT",
    "keyId":"1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "responseElements":null,
  "requestID":"cbe0ac33-8cca-49e5-9bb5-dc2b8dfcb389",
  "eventID":"1b9fedde-aa96-48cc-9dd9-a2cce2964b3c",
  "readOnly":true,
  "resources":[
    {
      "accountId":"123456789012",
      "type":"AWS::KMS::Key",
      "ARN":"arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    }
  ],
  "eventType":"AwsApiCall",
  "managementEvent":true,
  "recipientAccountId":"123456789012",
  "eventCategory":"Management"
}

```

## 解密

```

{
  "eventVersion":"1.09",
  "userIdentity":{
    "type":"AssumedRole",
    "principalId":"AROAIQDTESTANDEXAMPLE:Sampleuser01",
    "arn":"arn:aws:sts::123456789012:assumed-role/Admin/Sampleuser01",
    "accountId":"123456789012",
    "accessKeyId":"AKIAIOSFODNN7EXAMPLE3",
    "sessionContext":{
      "sessionIssuer":{
        "type":"Role",
        "principalId":"AROAIQDTESTANDEXAMPLE:Sampleuser01",
        "arn":"arn:aws:sts::123456789012:assumed-role/Admin/Sampleuser01",
        "accountId":"123456789012",
        "userName":"Admin"
      },
      "attributes":{

```

```
        "creationDate":"2024-08-16T20:47:04Z",
        "mfaAuthenticated":"false"
    }
},
    "invokedBy":"transcribe.streaming.amazonaws.com"
},
"eventTime":"2024-08-16T20:47:04Z",
"eventSource":"kms.amazonaws.com",
"eventName":"Decrypt",
"awsRegion":"us-east-1",
"sourceIPAddress":"transcribe.streaming.amazonaws.com",
"userAgent":"transcribe.streaming.amazonaws.com",
"requestParameters":{
    "keyId":"mrk-de27f019178f4fbf86512ab03ba860be",
    "encryptionAlgorithm":"SYMMETRIC_DEFAULT",
    "encryptionContext":{
        "aws:us-east-1:transcribe:medical-scribe:session-
id":"1234abcd-12ab-34cd-56ef-123456SAMPLE"
    }
},
"responseElements":null,
"requestID":"8b7fb865-48be-4e03-ac3d-e7bee3ba30a1",
"eventID":"68b7a263-d410-4701-9e2b-20c196628966",
"readOnly":true,
"resources":[
    {
        "accountId":"123456789012",
        "type":"AWS::KMS::Key",
        "ARN":"arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    }
],
"eventType":"AwsApiCall",
"managementEvent":true,
"recipientAccountId":"123456789012",
"eventCategory":"Management"
}
```

# 的文件歷史記錄 Amazon Transcribe

- 最新文件更新日期：2023 年 11 月 13 日

下表說明每個版本的重要變更 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="#">新功能</a>	AWS HealthScribe 現在支援臨床備註摘要的新範本：BIRP、SIRP、DAP、BH_SOAP 和 PH_SOAP。如需詳細資訊，請參閱 <a href="#">AWS HealthScribe 臨床文件檔案</a> 。	2025 年 5 月 30 日
<a href="#">新功能</a>	AWS HealthScribe 現在支援臨床備註摘要的 GIRPP 範本。如需詳細資訊，請參閱 <a href="#">AWS HealthScribe 臨床文件檔案</a> 。	2025 年 2 月 4 日
<a href="#">新功能</a>	AWS HealthScribe 現在支援即時轉錄醫療對話的串流。	2025 年 1 月 29 日
<a href="#">章節更新</a>	更新 8000 頻率的 ar-SA 語言支援。	2025 年 1 月 16 日
<a href="#">章節更新</a>	更新 8000 頻率的 ar-SA 語言支援。	2025 年 1 月 16 日
<a href="#">章節更新</a>	使用我們在 AWS GovCloud (US) (US-West、us-gov-west-1) AWS GovCloud (US) (US-East、us-gov-east-1) 或非洲 (Cape Town、af-south-1) 區域不支援的語言代碼更新支援的語言區段。	2025 年 1 月 14 日

<a href="#">章節更新</a>	使用名為 "audio_segments" 的新欄位擴展 Transcribe Batch json 輸出。	2024 年 7 月 15 日
<a href="#">功能更新</a>	將日記中的發言者上限更新為 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>	自訂詞彙欄的更新 - IPA 和 SoundsLike。	2024 年 4 月 30 日
<a href="#">功能更新</a>	Amazon Transcribe Call Analytics 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 最佳實務已更新。	2023 年 2 月 13 日
<a href="#">新的語言</a>	Amazon Transcribe 現在支援越南文和瑞典文。	2022 年 12 月 6 日
<a href="#">新功能</a>	Amazon Transcribe 現在支援即時通話分析。	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 參考現在是獨立的指南。	2022 年 4 月 1 日
<a href="#">新增章節</a>	包含針對 Amazon Transcribe Medical Amazon Transcribe 和 Amazon Transcribe Call Analytics 的新比較資料表。	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>	簡介章節現在展示 Amazon Transcribe 了使用案例。	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 現在可以為您的影片檔案產生字幕。	2021 年 9 月 16 日
<a href="#">新功能</a>	Amazon Transcribe 現在支援用於串流的 PII 修訂和識別。	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 AWS 帳戶 資源增加安全層級。	2021 年 9 月 10 日
<a href="#">新的語言</a>	Amazon Transcribe 現在支援非洲文、丹麥文、中文（繁體）、泰文、紐西蘭英文和南非英文。	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 現在支援批次轉錄任務的通話分析。	2021 年 8 月 4 日
<a href="#">新功能</a>	Amazon Transcribe 現在支援使用具有批次自訂語言模型的自訂詞彙。	2021 年 5 月 12 日
<a href="#">新功能</a>	Amazon Transcribe 現在支援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 現在支援澳洲英文、英式英文、印地文和美國西班牙文的自訂語言模型。	2021 年 3 月 19 日

<a href="#">新功能</a>	Amazon Transcribe 現在支援用於串流音訊轉錄的 OGG/OPUS 和 FLAC 轉碼器。	2020 年 11 月 24 日
<a href="#">新的語言</a>	Amazon Transcribe 新增支援義大利文和德文串流音訊轉錄。	2020 年 11 月 4 日
<a href="#">AWS 區域 擴展</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 新增對批次轉錄中介面 VPC 端點的支援。	2020 年 10 月 9 日
<a href="#">新功能</a>	Amazon Transcribe 新增支援串流中的頻道識別。	2020 年 9 月 17 日
<a href="#">新功能</a>	Amazon Transcribe 新增對批次轉錄中自動語言識別的支援。	2020 年 9 月 15 日
<a href="#">新功能</a>	Amazon Transcribe 新增對串流中發言者分割的支援。	2020 年 8 月 19 日
<a href="#">新功能</a>	Amazon Transcribe 新增對自訂語言模型的支援。	2020 年 8 月 5 日
<a href="#">新功能</a>	Amazon Transcribe 新增對串流中介面 VPC 端點的支援。	2020 年 6 月 26 日
<a href="#">新功能</a>	Amazon Transcribe 新增支援串流中的詞彙篩選。	2020 年 5 月 20 日
<a href="#">新功能</a>	Amazon Transcribe 新增支援自動修訂個人身分識別資訊。	2020 年 2 月 26 日

<a href="#">新功能</a>	Amazon Transcribe 新增建立自訂字詞詞彙以篩選轉錄的支援。	2019 年 12 月 20 日
<a href="#">新功能</a>	Amazon Transcribe 新增對將轉錄任務排入佇列的支援。	2019 年 12 月 19 日
<a href="#">新的語言</a>	Amazon Transcribe 新增對海灣阿拉伯文、希伯來文、日文、馬來文、瑞士德文、特拉克文和土耳其文的支援。	2019 年 11 月 21 日
<a href="#">AWS 區域 擴展</a>	Amazon Transcribe 現可於亞太區域 ( 東京 ) (ap-north-east-1) 使用。	2019 年 11 月 21 日
<a href="#">新功能</a>	Amazon Transcribe 新增對替代轉錄的支援。	2019 年 11 月 20 日
<a href="#">新的語言</a>	Amazon Transcribe 新增對荷蘭文、波斯文、印尼文、愛爾蘭英文、葡萄牙文、蘇格蘭英文、坦米爾文和威爾斯英文的支援。	2019 年 11 月 12 日
<a href="#">新的語言</a>	Amazon Transcribe 現在支援澳洲英文 (en-AU) 的串流轉錄。	2019 年 10 月 25 日
<a href="#">AWS 區域 擴展</a>	Amazon Transcribe 現已在中國 ( 北京 ) (cn-north-1) 和中國 ( 寧夏 ) (cn-northwest-1) 提供。	2019 年 10 月 9 日

<a href="#">新功能</a>	Amazon Transcribe 可讓您提供自己的 KMS key 來加密轉錄輸出檔案。如需詳細資訊，請參閱 <a href="#">StartStreamTranscription</a> API 的 <a href="#">OutputEncryptionKMSKeyId</a> 參數。	2019 年 9 月 24 日
<a href="#">新的語言</a>	Amazon Transcribe 新增對中文（國語）、簡化、中國大陸和俄文的支援。	2016 年 8 月 23 日
<a href="#">新功能</a>	Amazon Transcribe 新增使用 WebSocket 通訊協定串流音訊轉錄的支援。	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 擴展</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 新增對現代標準阿拉伯文的支援。	2019 年 5 月 28 日
<a href="#">新功能</a>	Amazon Transcribe 現在會將數字單字轉錄為美式英文的數字。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 新增對印地文和印度英文的支援。	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 新增對西班牙文的支援。	2019 年 4 月 19 日

<a href="#">AWS 區域 擴展</a>	Amazon Transcribe 現可於歐洲（法蘭克福）(eu-central-1) 和亞太區域（首爾）(ap-north-east-2) 使用。	2019 年 4 月 18 日
<a href="#">新的語言</a>	Amazon Transcribe 新增對英國英文、法文和加拿大法文串流轉錄的支援。	2019 年 4 月 5 日
<a href="#">新功能</a>	適用於 Ruby V3 的 AWS SDK 現在支援 Amazon Transcribe	2019 年 3 月 25 日
<a href="#">新功能</a>	Amazon Transcribe 允許自訂詞彙，這是 Amazon Transcribe 您想要在音訊輸入中辨識的特定字詞清單。	2019 年 3 月 25 日
<a href="#">新的語言</a>	Amazon Transcribe 新增對德文和韓文的支援。	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 擴展</a>	Amazon Transcribe 現已在南美洲（聖保羅）(sa-east-1) 提供。	2019 年 2 月 7 日
<a href="#">AWS 區域 擴展</a>	Amazon Transcribe 現可於亞太區域（孟買）(ap-south-1)、亞太區域（新加坡）(ap-southeast-1)、歐洲（倫敦）(eu-west-2) 和歐洲（巴黎）(eu-west3) 使用。	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 現在支援音訊串流的轉錄。	2018 年 11 月 19 日
<a href="#">新的語言</a>	Amazon Transcribe 新增對澳洲英文、英國英文和加拿大法文的支援。	2018 年 11 月 15 日
<a href="#">AWS 區域 擴展</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>	新增 AWS CloudTrail 和 Amazon CloudWatch Events 整合。	2018 年 6 月 28 日
<a href="#">新功能</a>	Amazon Transcribe 新增對自訂詞彙的支援。	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。