



Entwicklerhandbuch

Amazon Transcribe



Amazon Transcribe: Entwicklerhandbuch

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Die Handelsmarken und die Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irregeführt werden könnten oder Amazon in schlechtem Licht dargestellt oder diskreditiert werden könnte. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

Was ist Amazon Transcribe?	1
Amazon Transcribe und HIPAA-Eignung	1
Preisgestaltung	2
Regionale Verfügbarkeit und Quotas	2
Verfügbare Features	5
Unterstützte Sprachen	8
Unterstützte Programmiersprachen	20
Zeichensätze	21
Abchasisch	24
Afrikaans	27
Arabisch	27
Asturisch	29
Aserbaidshisch	29
Armenisch	30
Baschkirisch	31
Baskisch	34
Belarussisch	34
Bengalisch	35
Bosnisch	37
Bulgarisch	38
Katalanisch	39
Zentralkurdisch	40
Chinesisch – vereinfacht	41
Chinesisch – traditionell	42
Chinesisch, Kantonesisch	43
Kroatisch	43
Tschechisch	44
Dänisch	45
Niederländisch	45
Englisch	46
Estnisch	47
Farsi	47
Finnisch	48
Französisch	49

Galizisch	50
Georgisch	51
Deutsch	52
Griechisch	52
Gujarati	54
Hausa	56
Hebräisch	56
Hindi	57
Ungarisch	60
Isländisch	60
Indonesisch	61
Italienisch	61
Japanisch	62
Kabylish	62
Kannada	63
Kasachisch	65
Kinyarwanda	67
Koreanisch	67
Kirgisisch	68
Lettisch	70
Litauisch	71
Luganda	72
Mazedonisch	72
Malaiisch	74
Malayalam	75
Maltesisch	77
Marathi	77
Meadow Mari	79
Mongolisch	82
Norwegisches Bokmål	84
Odia/Oriya	85
Paschto	87
Polnisch	89
Portugiesisch	89
Pandschabi	91
Rumänisch	92

Russisch	93
Serbisch	94
Singhalesisch	97
Slowakisch	99
Slowenisch	100
Somali	100
Spanisch	101
Sundanesisch	102
Swahili	102
Schwedisch	103
Tagalog/Filipino	103
Tamil	104
Tatarisch	105
Telugu	108
Thailändisch	110
Türkisch	112
Ukrainisch	113
Uigurisch	114
Usbekisch	116
Vietnamesisch	118
Walisisch	122
Wolof	123
Zulu	124
Funktionsweise	125
Dateneingabe und -ausgabe	125
Medienformate	126
Audiokanäle	127
Samplerate	128
Ausgabe	128
Transkription von Zahlen	132
Erste Schritte	133
Melden Sie sich an für ein AWS-Konto	134
Installieren von AWS CLI und SDKs	134
Konfiguration IAM Anmeldedaten	135
Erstellen eines Amazon S3 bucket	135
Erstellen eines IAM policy	136

Transkribieren mit dem AWS-Managementkonsole	138
Transkribieren mit dem AWS CLI	147
Aufnahme einer neuen Tätigkeit im Bereich der Transkription	148
Abrufen des Status eines Transkriptionsauftrags	149
Auflistung Ihrer Transkriptionsaufträge	151
Löschen Ihres Transkriptionsauftrags	151
Transkribieren mit dem AWS SDKs	152
Arbeiten mit AWS SDKs	164
Transkribieren mit HTTP oder WebSockets	165
Streaming-Transkriptionen	167
Best Practices	168
Umgang mit LimitExceededException Fehlern	169
Streaming und Teilergebnisse	170
Partial-result Stabilisierung	171
Einrichten einer Streaming-Transkription	176
Ereignis-Stream-Kodierung	190
Datenframes	193
Auftragswarteschlangen	194
Aktivieren der Auftragswarteschlange	194
Taggen von Ressourcen	199
Tag-based Zugriffskontrolle	200
Hinzufügen von Tags zu Ihrem Amazon Transcribe Ressourcen	201
Sprecherpartitionierung (Diarisierung)	205
Aufteilung der Sprecher in einer Batch-Transkription	206
Partitionierung der Sprecher in einer Streaming-Transkription	209
Beispielausgabe	211
Transkribieren von Mehrkanal-Audio	218
Verwendung der Kanalidentifizierung in einer Batch-Transkription	219
Verwenden der Kanalidentifizierung in einer Streaming-Transkription	223
Beispielausgabe	224
Identifizierung von Sprachen	233
Batch-Sprachidentifizierung	233
Identifizierung von Sprachen in mehrsprachigen Audiodateien	234
Verbesserung der Genauigkeit der Sprachidentifizierung	235
Kombination der Sprachenidentifikation mit anderen Amazon Transcribe features	236
Sprachidentifizierung mit Batch-Transkriptionen	237

Identifizierung von Streaming-Sprachen	243
Identifizierung von Sprachen in mehrsprachigen Audiodateien	245
Sprachidentifizierung mit Streaming-Medien nutzen	245
Alternative Transkriptionen	251
Anforderung alternativer Transkriptionen	253
Verbesserung der Transkriptionsgenauigkeit	258
Benutzerdefinierte Vokabulare	259
Benutzerdefinierte Vokabulartabellen versus Listen	260
Erstellen eines benutzerdefinierten Vokabulars mithilfe einer Tabelle	261
Erstellen eines benutzerdefinierten Vokabulars mithilfe einer Liste	273
Verwenden eines benutzerdefinierten Vokabulars	276
Benutzerdefinierte Sprachmodelle	283
Datenquellen	283
Trainings- versus Tuningdaten	284
Erstellen eines benutzerdefinierten Sprachmodells	285
Verwenden eines benutzerdefinierten Sprachmodells	291
Wörter filtern	299
Erstellen eines Wortschatzfilters	300
Erstellen benutzerdefinierter Wortschatzfilter	301
Einen benutzerdefinierten Vokabularfilter verwenden	306
Verwenden eines benutzerdefinierten Vokabularfilters in einer Batch-Transkription	276
Verwenden eines benutzerdefinierten Wortschatzfilters in einer Streaming-Transkription	280
Erkennen von toxischer Sprache	314
Erkennen toxischer Sprache	315
Verwenden der Erkennung toxischer Sprache in einer Batch-Transkription	315
Beispielausgabe	319
Schwärzung von Transkripten	322
Schwärzen von PII in Ihrem Batch-Job	323
Schwärzen oder Identifizieren von PII in einem Echtzeit-Datenstrom	332
Beispielausgabe	340
Beispiel für eine geschwärzte Ausgabe (Batch)	340
Beispiel für eine geschwärzte Streaming-Ausgaben	343
Beispiel für die Ausgabe der PII-Identifikation	344
Erstellen von Untertiteln	347
Erzeugen von Untertiteldateien	348
Analyse von Callcenter-Audio	351

Häufige Anwendungsfälle	351
Überlegungen und zusätzliche Informationen	353
Regionale Verfügbarkeit und Quotas	354
Post-call Analytik	355
Post-call Einblicke	356
Kategorien erstellen	359
Beginn einer Transkription	370
Post-call Ergebnisse aus Analysen	377
Aktivieren der generativen Anrufzusammenfassung	390
Real-time Rufen Sie Analytics auf	395
Real-time Einblicke	396
Kategorien erstellen	400
Post-call Analytik mit Transkriptionen in Echtzeit	407
Beginn einer Transkription	414
Real-time Rufen Sie die Analytics-Ausgabe auf	423
Anrufe transkribieren Amazon Chime	430
Codebeispiele	432
Grundlagen	433
Aktionen	433
Szenarien	504
Erstellen einer Amazon-Transcribe-Streaming-App	505
Text in Sprache und zurück in Text konvertieren	506
Erstellen und Verfeinern eines benutzerdefinierten Vokabulars	507
Audio transkribieren und Auftragsdaten abrufen	517
Sicherheit	528
Identitäts- und Zugriffsverwaltung	529
Zielgruppe	529
Authentifizierung mit Identitäten	529
Verwalten des Zugriffs mit Richtlinien	531
Wie Amazon Transcribe funktioniert mit IAM	533
Confused-Deputy-Prävention	539
Identity-based Beispiele für politische Maßnahmen	540
Fehlerbehebung	550
Datenschutz	552
Inter-network Datenschutz im Verkehr	553
Datenverschlüsselung	553

Abmeldung von der Verwendung Ihrer Daten zur Serviceverbesserung	556
Überwachung Amazon Transcribe	557
Überwachung mit CloudWatch	558
Überwachung Amazon Transcribe mit CloudTrail	558
Verwenden Amazon EventBridge mit Amazon Transcribe	563
Compliance-Validierung	572
Ausfallsicherheit	572
Sicherheit der Infrastruktur	572
Schwachstellenanalyse und -management	573
VPC-Endpunkte (AWS PrivateLink)	573
Gemeinsam genutzte Subnetze	575
Bewährte Methoden für die Gewährleistung der Sicherheit	576
Amazon Transcribe Medizinisch	578
Regionale Verfügbarkeit und Quotas	579
Medizinische Fachgebiete	581
Transkribieren von medizinischen Begriffen und Messungen	582
Transkription von Zahlen	584
Transkribieren eines medizinischen Gesprächs	586
Transkribieren einer Audiodatei	587
Transkribieren eines Echtzeit-Streams	592
Aktivieren der Sprecherpartitionierung	595
Transkribieren von Mehrkanal-Audio	606
Transkribieren eines medizinischen Diktats	613
Transkribieren einer Audiodatei	615
Transkribieren eines medizinischen Streaming-Diktats	619
Erstellung und Verwendung von benutzerdefinierten medizinischen Vokabularen	622
Erstellen einer Textdatei für Ihr benutzerdefiniertes medizinisches Vokabular	623
Verwendung einer Textdatei zur Erstellung eines benutzerdefinierten medizinischen Vokabulars	628
Transkribieren einer Audiodatei unter Verwenden eines benutzerdefinierten medizinischen Vokabulars	630
Transkription eines Echtzeit-Streams unter Verwenden eines benutzerdefinierten medizinischen Vokabulars	632
Zeichensatz für Amazon Transcribe Medical	635
Identifizierung von PHI in einer Transkription	637
Identifizierung von PHI in einer Audiodatei	638

Identifizierung von PHI in einem Echtzeit-Stream	643
Erstellen alternativer Transkriptionen	645
VPC-Endpunkte (AWS PrivateLink)	648
Überlegungen zu Amazon Transcribe Medizinische VPC-Endpunkte	648
Erstellen eines Schnittstellen-VPC-Endpunkts für Amazon Transcribe Medical	649
Erstellen einer VPC-Endpunkttrichtlinie für Amazon Transcribe Medizinisches Streaming	649
Gemeinsam genutzte Subnetze	651
AWS HealthScribe	652
Sicherheit	653
Service-Verfügbarkeit	654
Technische Anforderungen	654
Unterstützte medizinische Fachgebiete	654
Arbeitsabläufe	655
Transkriptionsdatei	656
Datei mit klinischer Dokumentation	657
Kontext von Medical Scribe	658
Abschnitte mit den Vorlagen HISTORY_AND_PHYSICAL	658
Abschnitte mit GIRPP-Vorlagen	660
Abschnitte mit BIRP-Vorlagen	661
Abschnitte mit SIRP-Vorlagen	661
Abschnitte mit DAP-Vorlagen	662
Abschnitte der BH_SOAP-Vorlage	662
Abschnitte der PH_SOAP-Vorlage	663
Transkriptionsaufträge	663
Starten eines AWS HealthScribe Transkriptionsjob	664
Streaming	679
Richtlinien und Anforderungen	680
ResourceAccessRoleArn Rollen, Berechtigungen	680
Wird gestartet AWS HealthScribe Streaming-Transkription	682
Datenverschlüsselung im Ruhezustand für AWS HealthScribe	699
Einen vom Kunden verwalteten Schlüssel erstellen	700
Angabe eines vom Kunden verwalteten Schlüssels für AWS HealthScribe	704
AWS KMS Verschlüsselungskontext	705
Überwachen Sie Ihre Verschlüsselungsschlüssel für AWS HealthScribe	708
Dokumentverlauf	712
AWS Glossar	726

..... dccxxvii

Was ist Amazon Transcribe?

Amazon Transcribe ist ein automatischer Spracherkennungsdienst, der Modelle für maschinelles Lernen verwendet, um Audio in Text umzuwandeln. Sie können ihn Amazon Transcribe als eigenständigen Transkriptionsdienst verwenden oder um jeder Anwendung Sprache-zu-Text-Funktionen hinzuzufügen.

Mit Amazon Transcribe können Sie die Genauigkeit für Ihren speziellen Anwendungsfall durch Sprachanpassung verbessern, Inhalte filtern, um den Datenschutz oder eine zielgruppengerechte Sprache zu gewährleisten, Inhalte in Mehrkanal-Audio analysieren, die Sprache einzelner Sprecher partitionieren und vieles mehr.

Sie können Medien in Echtzeit transkribieren (Streaming) oder Mediendateien transkribieren, die sich in einem Bucket befinden (Batch). Amazon S3 Welche Sprachen für die einzelnen Transkriptionstypen unterstützt werden, ist der [Unterstützte Sprachen und sprachspezifische Funktionen](#)-Tabelle zu entnehmen.

Topics

- [Amazon Transcribe und HIPAA-Eignung](#)
- [Preisgestaltung](#)
- [Regionale Verfügbarkeit und Quotas](#)

Siehe [Was ist Amazon Transcribe?](#) für eine kurze Videotour durch diesen Service.

Weitere Informationen hierzu finden Sie unter [Wie Amazon Transcribe funktioniert](#) und [Erste Schritte mit Amazon Transcribe](#).

Tip

Informationen über die Amazon Transcribe -API finden Sie in der [API-Referenz](#).

Amazon Transcribe und HIPAA-Eignung

Amazon Transcribe ist durch die HIPAA-Eignung und BAA abgedeckt AWS, nach der BAA-Kunden alle gespeicherten und während der Übertragung befindlichen PHI verschlüsseln müssen, wenn sie

verwendet werden. Die automatische PHI-Identifizierung ist ohne zusätzliche Kosten und in allen Regionen, in denen das Unternehmen tätig ist, verfügbar. Amazon Transcribe Weitere Informationen finden Sie unter [HIPAA-Zulässigkeit und BAA](#).

Preisgestaltung

Amazon Transcribe ist ein nutzungsabhängiger Service. Die Preise basieren auf transkribierten Audiosekunden, die monatlich in Rechnung gestellt werden.

Die Nutzung wird in Schritten von einer Sekunde abgerechnet, wobei pro Anfrage mindestens 15 Sekunden berechnet werden. Beachten Sie, dass für Funktionen wie die Schwärzung von PII-Inhalten und benutzerdefinierte Sprachmodelle zusätzliche Gebühren anfallen.

[Informationen zu den jeweiligen Kosten finden Sie unter Preise. AWS-RegionAmazon Transcribe](#)

Regionale Verfügbarkeit und Quotas

Amazon Transcribe wird in den folgenden Bereichen unterstützt AWS-Regionen:

Region	Transkriptionstyp
af-south-1 (Kapstadt)	Batch, Streaming
ap-east-1 (Hongkong)	Batch
ap-northeast-1 (Tokyo)	Batch, Streaming
ap-northeast-2 (Seoul)	Batch, Streaming
ap-south-1 (Mumbai)	Batch, Streaming
ap-southeast-1 (Singapur)	Batch, Streaming
ap-southeast-2 (Sydney)	Batch, Streaming
ap-Southeast-5 (Malaysia)	Streamen
ap-Southeast-7 (Thailand)	Streamen
ca-central-1 (Kanada, Zentral)	Batch, Streaming

Region	Transkriptionstyp
eu-central-1 (Frankfurt)	Batch, Streaming
eu-central-2 (Zürich)	Batch, Streaming
eu-nord-1 (Stockholm)	Batch
eu-west-1 (Irland)	Batch, Streaming
eu-west-2 (London)	Batch, Streaming
eu-west-3 (Paris)	Batch
me-south-1 (Bahrain)	Batch
mx-central-1 (Mexiko)	Streamen
sa-east-1 (São Paulo)	Batch, Streaming
us-east-1 (N. Virginia)	Batch, Streaming
us-east-2 (Ohio)	Batch, Streaming
us-gov-east-1 (,) GovCloud US-East	Batch, Streaming
us-gov-west-1 (,) GovCloud US-West	Batch, Streaming
us-west-1 (San Francisco)	Batch
us-west-2 (Oregon)	Batch, Streaming

 Important

Die regionale Unterstützung unterscheidet sich für Amazon Transcribe Medical, und Call Analytics.

Um die Endpunkte für jede unterstützte Region zu erhalten, siehe [Dienstendpunkte](#) in der Allgemeinen AWS -Referenz.

Eine Liste der Quotas, die sich auf Ihre Transkriptionen beziehen, finden Sie in der Allgemeinen AWS -Referenz unter [Service Quotas](#). Einige Quotas können auf Anfrage geändert werden. Steht in der Spalte Anpassbar Ja, können Sie eine Erhöhung beantragen. Wählen Sie dazu den angegebenen Link.

Amazon Transcribe features

Um Ihnen bei der Entscheidung zu helfen, welche Amazon Transcribe Lösung am besten zu Ihrem Anwendungsfall passt, bietet die folgende Tabelle einen Funktionsvergleich.

Beachten Sie, dass sich „Batch“ und 'post-call' auf die Transkription einer Datei beziehen, die sich in einem Amazon S3 -Bucket befindet, während sich „Streaming“ und 'real-time' auf die Transkription von Medien in Echtzeit beziehen.

Feature	Amazon Transcribe	Amazon Transcribe Medical ¹	Amazon Transcribe Rufen Sie Analytics an

Konfigurationsoptionen

Alternative Transkriptionen	Batch, Streaming	Batch, Streaming	Nein
Kanalidentifizierung	Batch, Streaming	Batch, Streaming	post-call, real-time
Auftragswarteschlangen	Batch	Nein	post-call
Sprachidentifizierung	Batch, Streaming	Nein	post-call
Multi-language Identifikation	Batch, Streaming	Nein	Nein
Sprecherdiarisierung	Batch, Streaming	Batch, Streaming	post-call
Transkribieren von Ziffern ²	Batch, Streaming	Batch, Streaming	post-call, real-time

Gesprächsanalytik

Anrufmerkmale	Nein	Nein	post-call
Anrufzusammenfassung ²	Nein	Nein	post-call

Feature	Amazon Transcribe	Amazon Transcribe Medical ¹	Amazon Transcribe Rufen Sie Analytics an
Benutzerdefinierte Kategorisierung	Nein	Nein	post-call
Real-time Kategorie Ereignisse	Nein	Nein	real-time
Real-time Problemerkennung ²	Nein	Nein	real-time
Real-time Stimmung des Sprechers	Nein	Nein	real-time
Sprecherstimmung	Nein	Nein	post-call
Sprachanpassung			
Benutzerdefinierte Sprachmodelle ²	Batch, Streaming	Nein	post-call, real-time
Benutzerdefinierte Vokabulare	Batch, Streaming	Batch, Streaming	post-call, real-time
Organisation der Ressourcen			
Tagging	Batch	Batch	post-call
Sensible Daten			
Identifizierung persönlicher Gesundheitsinformationen ²	Nein	Batch, Streaming	Nein

Feature	Amazon Transcribe	Amazon Transcribe Medical ¹	Amazon Transcribe Rufen Sie Analytics an
Identifizierung persönlich identifizierbarer Informationen ²	Streamen	Nein	real-time
Audio schwärzen ²	Nein	Nein	post-call, real-time
Schwärzung von Transkripten ²	Batch, Streaming	Nein	post-call, real-time
Wortschatzfilter	Batch, Streaming	Nein	post-call, real-time
Video			
Untertitel	Batch	Nein	Nein

 ¹ Amazon Transcribe Medical ist nur in US-Englisch verfügbar.

² Dieses Feature ist nicht für alle Sprachen verfügbar; weitere Einzelheiten finden Sie in der Tabelle [Unterstützte Sprachen und sprachspezifische Funktionen](#) .

Unterstützte Sprachen und sprachspezifische Funktionen

Die Sprachen, die von unterstützt werden, Amazon Transcribe sind in der folgenden Tabelle aufgeführt. Außerdem sind die sprachspezifischen Funktionen aufgeführt. Bitte vergewissern Sie sich, dass die Funktion, die Sie verwenden möchten, für die Sprache Ihres Mediums unterstützt wird, bevor Sie mit der Transkription fortfahren.

Die vollständige Liste der Amazon Transcribe Funktionen finden Sie in der [Funktionsübersicht](#).

In der folgenden Tabelle bezieht sich „Batch“ auf das Transkribieren einer Mediendatei in einem Amazon S3 Bucket und „Streaming“ auf das Transkribieren von gestreamten Medien in Echtzeit. Bei Call Analytics-Transkriptionen 'post-call' bezieht sich dies auf das Transkribieren einer Mediendatei, die sich in einem Amazon S3 Bucket befindet, und 'real-time' bezieht sich auf das Transkribieren von gestreamten Medien in Echtzeit.

Sprache	Sprachcode	Dateneingabe	Transkription von Zahlen	Akronyme	Benutzerdefinierte Sprachmodelle	Schwärzen	Call Analytics * –
Abchasisch	ab-GE	Batch	Nein	Batch	Nein	Nein	Nein
Afrikaans	af-ZA	Batch, Streaming	Nein	Batch, Streaming	Nein	Nein	Nein
Albanisch	sq-AL	Streaming*	Nein	streamen*	Nein	Nein	Nein
Amharisch	am-ET	streamen*	Nein	streamen*	Nein	Nein	Nein
Arabisch, Golf	ar-AE	Batch, Streaming	Batch	Nein	Nein	Nein	post-call
Arabisch, Moderner Standard	ar-SA	Batch, Streaming	Nein	Nein	Nein	Nein	Nein

Sprache	Sprachcode	Dateneingabe	Transkription von Zahlen	Akronyme	Benutzerdefinierte Sprachmodelle	Schwärzen	Call Analytics * -
Armenisch	hy-AM	Batch, Streaming *	Nein	Batch, Streaming *	Nein	Nein	Nein
Asturisch	ast-ES	Batch	Nein	Batch	Nein	Nein	Nein
Aserbaidschanisch	az-AZ	Batch	Nein	Batch	Nein	Nein	Nein
Baschkirisch	ba-RU	Batch	Nein	Batch	Nein	Nein	Nein
Baskisch	eu-ES	Batch, Streaming	Nein	Batch, Streaming	Nein	Nein	Nein
Belarussisch	be-BY	Batch, Streaming *	Nein	Batch, Streaming *	Nein	Nein	Nein
Bengalisch	bn-IN	Batch, Streaming *	Batch	Batch, Streaming *	Nein	Nein	Nein
Bosnisch	bs-BA	Batch, Streaming *	Nein	Batch, Streaming *	Nein	Nein	Nein
birmanisch	my-MM	streamen*	Nein	streamen*	Nein	Nein	Nein
Bulgarisch	bg-BG	Batch, Streaming *	Nein	Batch, Streaming *	Nein	Nein	Nein

Sprache	Sprachcode	Dateneingabe	Transkription von Zahlen	Akronyme	Benutzerdefinierte Sprachmodelle	Schwärzen	Call Analytics * -
Katalanisch	ca-ES	Batch, Streaming	Streamen	Batch, Streaming	Nein	Nein	Nein
Zentralkurdisch, Iran	ckb-IR	Batch, Streaming *	Nein	Batch, Streaming *	Nein	Nein	Nein
Zentralkurdisch, Irak	ckb-IQ	Batch, Streaming *	Nein	Batch, Streaming *	Nein	Nein	Nein
Chinesisch, Kantonesisch	zh-HK (yue-HK)	Batch, Streaming	Nein	Nein	Nein	Nein	Nein
Chinesisch – vereinfacht	zh-CN	Batch, Streaming	Nein	Nein	Nein	Nein	post-call
Chinesisch – traditionell	zh-TW	Batch, Streaming	Nein	Nein	Nein	Nein	Nein
Kroatisch	hr-HR	Batch, Streaming	Nein	Batch, Streaming	Nein	Nein	Nein
Tschechisch	cs-CZ	Batch, Streaming	Nein	Batch, Streaming	Nein	Nein	Nein
Dänisch	da-DK	Batch, Streaming	Batch	Batch, Streaming	Nein	Nein	Nein

Sprache	Sprachcode	Dateneingabe	Transkription von Zahlen	Akronyme	Benutzerdefinierte Sprachmodelle	Schwärzen	Call Analytics *-
Niederländisch	nl-NL	Batch, Streaming	Batch	Batch, Streaming	Nein	Nein	Nein
Englisch , Australisch	en-AU	Batch, Streaming	Batch, Streaming	Batch, Streaming	Batch, Streaming	Streamen	post-call, real-time
Englisch , Britisch	en-GB	Batch, Streaming	Batch, Streaming	Batch, Streaming	Batch, Streaming	Streamen	post-call, real-time
Englisch , Indisch	en-IN	Batch, Streaming	Batch, Streaming	Batch, Streaming	Nein	Streamen	post-call
Englisch , Irisch	en-IE	Batch, Streaming	Batch, Streaming	Batch, Streaming	Nein	Streamen	post-call
Englisch , Neuseeland	en-NZ	Batch, Streaming	Batch, Streaming	Batch, Streaming	Nein	Streamen	Nein
Englisch , Schottisch	en-AB	Batch, Streaming	Batch, Streaming	Batch, Streaming	Nein	Streamen	post-call
Englisch , Südafrika	en-ZA	Batch, Streaming	Batch, Streaming	Batch, Streaming	Nein	Streamen	Nein
Englisch , US	en-US	Batch, Streaming	Batch, Streaming	Batch, Streaming	Batch, Streaming	Batch, Streaming	post-call, real-time
Englisch , Walisisch	en-WL	Batch, Streaming	Batch, Streaming	Batch, Streaming	Nein	Streamen	post-call
Estnisch	et-EE	Batch	Nein	Batch	Nein	Nein	Nein

Sprache	Sprachcode	Dateneingabe	Transkription von Zahlen	Akronyme	Benutzerdefinierte Sprachmodelle	Schwärzen	Call Analytics *-
Estnisch	et-ET	Batch, Streaming*	Nein	Batch, Streaming*	Nein	Nein	Nein
Farsi	fa-IR	Batch, Streaming	Nein	Nein	Nein	Nein	Nein
Farsi , afghanisch	fa-AF	streamen*	Nein	streamen*	Nein	Nein	Nein
Finnisch	fi-FI	Batch, Streaming	Nein	Batch, Streaming	Nein	Nein	Nein
Französisch	fr-FR	Batch, Streaming	Batch, Streaming	Batch, Streaming	Nein	Streamen	post-call, real-time
Französisch , Kanadisch	fr-CA	Batch, Streaming	Batch, Streaming	Batch, Streaming	Nein	Streamen	post-call, real-time
Galizisch	gl-ES	Batch, Streaming	Nein	Batch, Streaming	Nein	Nein	Nein
Georgisch	ka-GE	Batch, Streaming*	Nein	Batch, Streaming*	Nein	Nein	Nein
Deutsch	de-DE	Batch, Streaming	Batch, Streaming	Batch, Streaming	Batch, Streaming	Streamen	post-call, real-time
Deutsch , Schweizerisch	de-CH	Batch, Streaming	Batch, Streaming	Batch, Streaming	Nein	Streamen	post-call

Sprache	Sprachcode	Dateneingabe	Transkription von Zahlen	Akronyme	Benutzerdefinierte Sprachmodelle	Schwärzen	Call Analytics *-
Griechisch	el-GR	Batch, Streaming	Nein	Batch, Streaming	Nein	Nein	Nein
Gujarati	gu-IN	Batch, Streaming*	Nein	Batch, Streaming*	Nein	Nein	Nein
Haitianisches Kreolisch	ht-HT	streamen*	Nein	streamen*	Nein	Nein	Nein
Hausa	ha-NG	Batch	Nein	Batch	Nein	Nein	Nein
Hebräisch	he-IL	Batch, Streaming	Batch	Nein	Nein	Nein	Nein
Hindi , Indisch	hi-IN	Batch, Streaming	Batch	Batch, Streaming	Batch	Nein	post-call
Ungarisch	hu-HU	Batch, Streaming*	Nein	Batch, Streaming*	Nein	Nein	Nein
Isländisch	is-IS	Batch, Streaming*	Nein	Batch, Streaming*	Nein	Nein	Nein
Indonesisch	id-ID	Batch, Streaming	Batch	Batch, Streaming	Nein	Nein	Nein
Italienisch	it-IT	Batch, Streaming	Batch, Streaming	Batch, Streaming	Nein	Streamen	post-call, real-time

Sprache	Sprachcode	Dateneingabe	Transkription von Zahlen	Akronyme	Benutzerdefinierte Sprachmodelle	Schwärzen	Call Analytics *-
Japanisch	ja-JP	Batch, Streaming	Batch, Streaming	Nein	Batch, Streaming	Nein	post-call
Javanesisch	jv-ID	streamen*	Nein	streamen*	Nein	Nein	Nein
Kabylish	kab-DZ	Batch, Streaming*	Nein	Batch, Streaming*	Nein	Nein	Nein
Kannada	kn-IN	Batch, Streaming*	Nein	Batch, Streaming*	Nein	Nein	Nein
Kasachisch	kk-KZ	Batch, Streaming*	Nein	Batch, Streaming*	Nein	Nein	Nein
Khmer	km-KH	streamen*	Nein	streamen*	Nein	Nein	Nein
Kinyarwanda	rw-RW	Batch	Nein	Batch	Nein	Nein	Nein
Koreanisch	ko-KR	Batch, Streaming	Nein	Batch, Streaming	Nein	Nein	post-call
Kirgisisch	ky-KG	Batch	Nein	Batch	Nein	Nein	Nein
Lettisch	lv-LV	Batch, Streaming	Nein	Batch, Streaming	Nein	Nein	Nein
Litauisch	lt-LT	Batch, Streaming*	Nein	Batch, Streaming*	Nein	Nein	Nein

Sprache	Sprachcode	Dateneingabe	Transkription von Zahlen	Akronyme	Benutzerdefinierte Sprachmodelle	Schwärzen	Call Analytics *-
Luganda	lg-IN	Batch, Streaming*	Nein	Batch, Streaming*	Nein	Nein	Nein
Mazedonisch	mk-MK	Batch, Streaming*	Nein	Batch, Streaming*	Nein	Nein	Nein
Malaiisch	ms-MY	Batch, Streaming	Nein	Batch, Streaming	Nein	Nein	Nein
Malayalam	ml-IN	Batch, Streaming*	Nein	Batch, Streaming*	Nein	Nein	Nein
Maltesisch	mt-MT	Batch	Nein	Batch	Nein	Nein	Nein
Marathi	mr-IN	Batch, Streaming*	Nein	Batch, Streaming*	Nein	Nein	Nein
Meadow Mari	mhr-RU	Batch	Nein	Batch	Nein	Nein	Nein
Mongolisch	mn-MN	Batch	Nein	Batch	Nein	Nein	Nein
Nepali	ne-NP	streamen*	Nein	streamen*	Nein	Nein	Nein
Norwegisches Bokmål	no-NO	Batch, Streaming	Batch	Batch, Streaming	Nein	Nein	Nein

Sprache	Sprachcode	Dateneingabe	Transkription von Zahlen	Akronyme	Benutzerdefinierte Sprachmodelle	Schwärzen	Call Analytics *-
Odia/Oriya	or-IN	Batch, Streaming*	Nein	Batch, Streaming*	Nein	Nein	Nein
Paschto	ps-AF	Batch, Streaming*	Nein	Batch, Streaming*	Nein	Nein	Nein
Polnisch	pl-PL	Batch, Streaming	Batch	Batch, Streaming	Nein	Nein	Nein
Portugiesisch	pt-PT	Batch, Streaming	Batch, Streaming	Batch, Streaming	Nein	Streamen	post-call
Portugiesisch, Brasilianisch	pt-BR	Batch, Streaming	Batch, Streaming	Batch, Streaming	Nein	Streamen	post-call, real-time
Pandschabi	pa-IN	Batch, Streaming*	Nein	Batch, Streaming*	Nein	Nein	Nein
Rumänisch	ro-RO	Batch, Streaming	Batch	Batch, Streaming	Nein	Nein	Nein
Russisch	ru-RU	Batch, Streaming	Batch	Batch, Streaming	Nein	Nein	Nein
Serbisch	sr-RS	Batch, Streaming	Nein	Batch, Streaming	Nein	Nein	Nein

Sprache	Sprachcode	Dateneingabe	Transkription von Zahlen	Akronyme	Benutzerdefinierte Sprachmodelle	Schwärzen	Call Analytics *-
Singhalesisch	si-LK	Batch, Streaming*	Nein	Batch, Streaming*	Nein	Nein	Nein
Slowakisch	sk-SK	Batch, Streaming	Nein	Batch, Streaming	Nein	Nein	Nein
Slowenisch	sl-SI	Batch, Streaming*	Nein	Batch, Streaming*	Nein	Nein	Nein
Somali	so-SO	Batch, Streaming	Batch	Batch, Streaming	Nein	Nein	Nein
Spanisch	es-ES	Batch, Streaming	Batch, Streaming	Batch, Streaming	Nein	Streamen	post-call
Spanisch, Mexikaner	es-MX	Streaming*	Nein	streamen*	Nein	Nein	Nein
Spanisch, US	es-US	Batch, Streaming	Batch, Streaming	Batch, Streaming	Batch, Streaming	Batch, Streaming	post-call, real-time
Sundanesisch	su-ID	Batch, Streaming*	Nein	Batch, Streaming*	Nein	Nein	Nein
Swahili, Kenia	sw-KE	Batch, Streaming*	Batch	Batch, Streaming*	Nein	Nein	Nein
Swahili, Burundi	sw-BI	Batch, Streaming*	Batch	Batch, Streaming*	Nein	Nein	Nein

Sprache	Sprachcode	Dateneingabe	Transkription von Zahlen	Akronyme	Benutzerdefinierte Sprachmodelle	Schwärzen	Call Analytics * -
Swahili, Ruanda	sw-RW	Batch, Streaming *	Batch	Batch, Streaming *	Nein	Nein	Nein
Swahili, Tansania	sw-TZ	Batch, Streaming *	Batch	Batch, Streaming *	Nein	Nein	Nein
Swahili, Uganda	sw-UG	Batch, Streaming *	Batch	Batch, Streaming *	Nein	Nein	Nein
Schwedisch	sv-SE	Batch, Streaming	Batch	Batch, Streaming	Nein	Nein	Nein
Tagalog/Filipino	tl-PH	Batch, Streaming	Batch	Batch, Streaming	Nein	Nein	Nein
Tamil	ta-IN	Batch, Streaming *	Batch	streamen*	Nein	Nein	Nein
Tatarisch	tt-RU	Batch	Nein	Batch	Nein	Nein	Nein
Telugu	te-IN	Batch, Streaming *	Nein	streamen*	Nein	Nein	Nein
Thailändisch	th-TH	Batch, Streaming	Batch	Batch, Streaming	Nein	Nein	Nein
Türkisch	tr-TR	Batch, Streaming *	Batch	Batch, Streaming *	Nein	Nein	Nein

Sprache	Sprachcode	Dateneingabe	Transkription von Zahlen	Akronyme	Benutzerdefinierte Sprachmodelle	Schwärzen	Call Analytics * -
Ukrainisch	uk-UA	Batch, Streaming	Batch	Batch, Streaming	Nein	Nein	Nein
Uigurisch	ug-CN	Batch	Nein	Batch	Nein	Nein	Nein
Usbekisch	uz-UZ	Batch, Streaming *	Nein	Batch, Streaming *	Nein	Nein	Nein
Vietnamesisch	vi-VN	Batch, Streaming	Batch	Batch, Streaming	Nein	Nein	Nein
Walisisch	cy-WL	Batch, Streaming *	Nein	Batch, Streaming *	Nein	Nein	Nein
Wolof	wo-SN	Batch	Nein	Batch	Nein	Nein	Nein
Zulu	zu-ZA	Batch, Streaming	Nein	Batch, Streaming	Nein	Nein	Nein

*Streaming für diese Sprachen ist in den folgenden Sprachen nicht verfügbar AWS-Regionen: af-south-1 (Kapstadt), ap-northeast-1 (Tokio), ap-southeast-5 (Malaysia), ap-southeast-7 (Thailand) und cn-northwest-1 (Ningxia).

* Die folgenden Call-Analytics-Erkenntnisse werden nur in bestimmten englischen Dialekten unterstützt:

- [Anrufzusammenfassung](#): en-* (alle englischen Dialekte)
- [Erkennung von Problemen](#): en-AU, en-GB, en-US

Unterstützte Programmiersprachen

Amazon Transcribe AWS unterstützt die folgenden SDKs:

Batch-Transkriptionen	Streaming-Transkriptionen
.NET	.NET (Transcribe Medical und HealthScribe wird nicht unterstützt)
AWS Befehlszeilenschnittstelle (CLI)	Die CLI wird für das Streaming nicht unterstützt.
C++	C++
Go	Go
Java V2	Java V2
JavaScript	JavaScript V3
PHP V3	Das SDK wird für Streaming nicht unterstützt.
Python Boto3	Python-Streaming-SDK für Amazon Transcribe
Rubin V3	Rubin V3
Rust	Rust

Informationen zur Verwendung von SDKs mit Amazon Transcribe finden Sie unter [Transkribieren mit dem AWS SDKs](#)

Weitere Informationen zu allen verfügbaren AWS SDKs und Builder-Tools finden Sie unter [Tools to Build On. AWS](#)

Tip

SDK-Codebeispiele finden Sie in diesen GitHub Repositorys:

- [AWS Codebeispiele](#)

- [Amazon Transcribe Beispiele](#)

Zeichensätze für benutzerdefinierte Vokabularen und Wortschatzfilter

Für jede Amazon Transcribe unterstützte Sprache gibt es einen bestimmten Satz von Zeichen, die erkannt Amazon Transcribe werden können. Wenn Sie ein benutzerdefiniertes Vokabular oder einen Wortschatzfilter erstellen, verwenden Sie nur die im Zeichensatz Ihrer Sprache aufgeführten Zeichen. Wenn Sie nicht unterstützte Zeichen verwenden, schlägt Ihr benutzerdefiniertes Vokabular oder Ihr Wortschatzfilter fehl.

Important

Stellen Sie sicher, dass Ihre benutzerdefinierte Vokabulardatei nur die unterstützten Unicode-Codepunkte und Codepunktfolgen verwendet, die in den folgenden Zeichensätzen aufgeführt sind.

Viele Unicode-Zeichen können in gängigen Schriftarten identisch erscheinen, auch wenn sie unterschiedliche Codepunkte verwenden. Es werden nur die in diesem Leitfaden aufgeführten Codepunkte unterstützt. Das französische Wort *déjà* kann beispielsweise mit vorkomponierten Zeichen (bei denen ein Unicode-Wert ein akzentuiertes Zeichen darstellt) oder mit dekomponierten Zeichen (bei denen zwei Unicode-Werte ein akzentuiertes Zeichen darstellen, ein Wert für das Grundzeichen und ein weiterer für den Akzent) wiedergegeben werden.

- Vorkomponierte Version: 0064 **00E9** 006A **00E0** (wird als *déjà* wiedergegeben)
- Dekomponierte Version: 0064 **0065** **0301** 006A **0061** **0300** (wird als *déjà* wiedergegeben)

Themen

- [Abchasischer Zeichensatz](#)
- [Afrikaans-Zeichensatz](#)
- [Arabischer Zeichensatz](#)
- [Asturischer Zeichensatz](#)
- [Aserbaidshanischer Zeichensatz](#)

- [Armenischer Zeichensatz](#)
- [Baschkirischer Zeichensatz](#)
- [Baskischer Zeichensatz](#)
- [Belarussischer Zeichensatz](#)
- [Bengalischer Zeichensatz](#)
- [Bosnischer Zeichensatz](#)
- [Bulgarischer Zeichensatz](#)
- [Katalanischer Zeichensatz](#)
- [Zentralkurdischer Zeichensatz](#)
- [Chinesisch, Mandarin \(Festlandchina\), vereinfachter Zeichensatz](#)
- [Chinesisch, Mandarin \(Taiwan\), traditioneller Zeichensatz](#)
- [Chinesisch, Kantonesisch \(Hongkong\), Traditioneller Zeichensatz](#)
- [Kroatischer Zeichensatz](#)
- [Tschechischer Zeichensatz](#)
- [Dänischer Zeichensatz](#)
- [Niederländischer Zeichensatz](#)
- [Englischer Zeichensatz](#)
- [Estnischer Zeichensatz](#)
- [Farsi-Zeichensatz](#)
- [Finnischer Zeichensatz](#)
- [Französischer Zeichensatz](#)
- [Galizischer Zeichensatz](#)
- [Georgischer Zeichensatz](#)
- [Deutscher Zeichensatz](#)
- [Griechischer Zeichensatz](#)
- [Gujarati-Zeichensatz](#)
- [Hausa-Zeichensatz](#)
- [Hebräischer Zeichensatz](#)
- [Hindi-Zeichensatz](#)
- [Ungarischer Zeichensatz](#)

- [Isländischer Zeichensatz](#)
- [Indonesischer Zeichensatz](#)
- [Italienischer Zeichensatz](#)
- [Japanischer Zeichensatz](#)
- [Kabyllischer Zeichensatz](#)
- [Kannada-Zeichensatz](#)
- [Kasachischer Zeichensatz](#)
- [Kinyarwanda-Zeichensatz](#)
- [Koreanischer Zeichensatz](#)
- [Kirgisischer Zeichensatz](#)
- [Lettischer Zeichensatz](#)
- [Litauischer Zeichensatz](#)
- [Luganda-Zeichensatz](#)
- [Mazedonischer Zeichensatz](#)
- [Malaiischer Zeichensatz](#)
- [Malayalam-Zeichensatz](#)
- [Maltesischer Zeichensatz](#)
- [Marathi-Zeichensatz](#)
- [Meadow Mari-Zeichensatz](#)
- [Mongolischer Zeichensatz](#)
- [Norwegischer Bokmål-Zeichensatz](#)
- [Odia/Oriya Zeichensatz](#)
- [Paschtu-Zeichensatz](#)
- [Polnischer Zeichensatz](#)
- [Portugiesischer Zeichensatz](#)
- [Pandschabi-Zeichensatz](#)
- [Rumänischer Zeichensatz](#)
- [Russischer Zeichensatz](#)
- [Serbischer Zeichensatz](#)
- [Singhalesischer Zeichensatz](#)

- [Slowakischer Zeichensatz](#)
- [Slowenischer Zeichensatz](#)
- [Somalischer Zeichensatz](#)
- [Spanischer Zeichensatz](#)
- [Sundanesischer Zeichensatz](#)
- [Swahili-Zeichensatz](#)
- [Schwedischer Zeichensatz](#)
- [Tagalog/Filipino Zeichensatz](#)
- [Tamilischer Zeichensatz](#)
- [Tatarischer Zeichensatz](#)
- [Telugu-Zeichensatz](#)
- [Thailändischer Zeichensatz](#)
- [Türkischer Zeichensatz](#)
- [Ukrainischer Zeichensatz](#)
- [Uigurischer Zeichensatz](#)
- [Usbekischer Zeichensatz](#)
- [Vietnamesischer Zeichensatz](#)
- [Walisischer Zeichensatz](#)
- [Wolof-Zeichensatz](#)
- [Zulu-Zeichensatz](#)

Abchasischer Zeichensatz

Für ein abchasisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
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	жџ	0497
к	043A	џ	0499
л	043B	ќ	049B
м	043C	к	049F
н	043D	т	04A1
о	043E	ћ	04A3
п	043F	н	04A5
р	0440	ѣ	04A9
с	0441	џ	04AB
т	0442	т	04AD
у	0443	у	04AF

Zeichen	Code	Zeichen	Code
ф	0444	ყ	04B1
х	0445	ჩ	04B3
ц	0446	ც	04B5
ч	0447	ჭ	04B7
ш	0448	ჰ	04BB
щ	0449	ჲ	04BD
ъ	044A	ჳ	04BF
ы	044B	#	04CA
ь	044C	ა	04D1
э	044D	ბ	04D3
ю	044E	გ	04D7
я	044F	დ	04D9
#	0450	ე	04E1
ë	0451	წ	04E3
ђ	0452	ჭ	04E7
í	0453	ჲ	04E9
є	0454	ჳ	04EF
s	0455	ჴ	04F1
i	0456	ჵ	04F3
ï	0457	#	04F7

Zeichen	Code	Zeichen	Code
j	0458	ÿ	04F9
#	0525		

Afrikaans-Zeichensatz

Für ein benutzerdefiniertes Afrikaans-Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
á	00E1	ï	00EF
è	00E8	ó	00F3
é	00E9	ô	00F4
ê	00EA	ö	00F6
ë	00EB	ú	00FA
í	00ED	û	00FB
î	00EE	ü	00FC

Arabischer Zeichensatz

Für ein arabisches benutzerdefiniertes Vokabular können Sie die folgenden Unicode-Zeichen im Feld `Phrase` verwenden. Sie können auch das Zeichen Bindestrich (-) verwenden, um Wörter zu trennen.

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

Asturischer Zeichensatz

Für ein asturisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
á	00E1	ñ	00F1
é	00E9	ó	00F3
í	00ED	ú	00FA
ü	00FC		

Aserbaidshanischer Zeichensatz

Für ein aserbaidshanisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
ä	00E4	ğ	011F

Zeichen	Code	Zeichen	Code
ç	00E7	ı	0131
ö	00F6	ş	015F
ü	00FC	ə	0259
.	0307		

Armenischer Zeichensatz

Für ein armenisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
մ	0561	ւ	0574
բ	0562	յ	0575
գ	0563	և	0576
դ	0564	ռ	0577
ե	0565	ն	0578
զ	0566	ժ	0579
է	0567	ղ	057A
ը	0568	շ	057B

Zeichen	Code	Zeichen	Code
ṗ	0569	ṛ	057C
ḑ	056A	ṛ	057D
ḥ	056B	ṛ	057E
ḷ	056C	ṛ	057F
ḽ	056D	ṛ	0580
ḏ	056E	ḡ	0581
ḽ	056F	ṛ	0582
ḥ	0570	ṛ	0583
ḏ	0571	f	0584
ṛ	0572	o	0585
ḥ	0573	ṛ	0586

Baschkirischer Zeichensatz

Für ein baschkirisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
а	0430	ӑ	0459

Zeichen	Code	Zeichen	Code
б	0431	һ	045A
в	0432	ћ	045B
г	0433	ќ	045C
д	0434	#	045D
е	0435	ӱ	045E
ж	0436	џ	045F
з	0437	ѓ	0491
и	0438	ѣ	0493
й	0439	жҥ	0497
к	043A	ҙ	0499
л	043B	қ	049B
м	043C	к	049F
н	043D	т	04A1
о	043E	ң	04A3
п	043F	Һ	04A5
р	0440	Ә	04A9
с	0441	Ҙ	04AB
т	0442	ҫ	04AD
у	0443	Ү	04AF
ф	0444	Ү	04B1

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

Baskischer Zeichensatz

Für ein baskisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
á	00E1	ñ	00F1
é	00E9	ó	00F3
í	00ED	ú	00FA
ü	00FC		

Belarussischer Zeichensatz

Für ein belarussisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
a	0430	c	0441

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

Bengalischer Zeichensatz

Für ein bengalisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
৐	0981	দ	09A6
ং	0982	ধ	09A7
ঃ	0983	ন	09A8
অ	0985	প	09AA
আ	0986	ফ	09AB
ই	0987	ব	09AC
ঈ	0988	ভ	09AD
উ	0989	ম	09AE
ঊ	098A	য	09AF
ঋ	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

Zeichen	Code	Zeichen	Code
চ	099A	ঢ়	09C0
ছ	099B	ঝ	09C1
জ	099C	ঞ	09C2
ঝ	099D	ঞ	09C3
ঞ	099E	ট	09C4
ট	099F	ঠ	09C7
ঠ	09A0	ড	09C8
ড	09A1	ঢ	09CB
ঢ	09A2	ঢ়	09CC
ণ	09A3	,	09CD
ত	09A4	#	09CE
থ	09A5	ঢ়	09D7

Bosnischer Zeichensatz

Für ein bosnisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
ć	0107	đ	0111
č	010D	š	0161
ž	017E		

Bulgarischer Zeichensatz

Für ein bulgarisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
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

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

Katalanischer Zeichensatz

Für ein katalanisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
à	00E0	ï	00EF
ç	00E7	ò	00F2
è	00E8	ó	00F3
é	00E9	ú	00FA
í	00ED	ü	00FC
l·	0140		

Zentralkurdischer Zeichensatz

Für ein zentralkurdisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
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	ڤ	06A4
ر	0631	ک	06A9
ز	0632	گ	06AF
س	0633	ڭ	06B5
ش	0634	ه	06BE
ع	0639	ۆ	06C6
غ	063A	ۇ	06C7

Zeichen	Code	Zeichen	Code
ف	0641	س	06CC
ق	0642	ئ	06CE
ج	0644	ه	06D5

Chinesisch, Mandarin (Festlandchina), vereinfachter Zeichensatz

Für chinesische (vereinfacht) benutzerdefinierte Vokabulare kann das Feld `Phrase` jedes der in der folgenden Datei aufgeführten Zeichen verwenden:

- [zh-cn-Zeichensatz](#)

Das Feld `SoundsLike` kann die in der folgenden Datei aufgeführten Pinyin-Silben enthalten:

- [pinyin-Zeichensatz](#)

Wenn Sie im Feld `SoundsLike` Pinyin-Silben verwenden, trennen Sie die Silben durch einen Bindestrich (-).

Amazon Transcribe steht für die vier Töne auf Chinesisch (vereinfacht) unter Verwendung von Zahlen. Die folgende Tabelle zeigt, wie Tonzeichen für das Wort „ma“ abgebildet werden.

Ton	Tonmarkierung	Tonnummer
Ton 1	mā	ma1
Ton 2	má	ma2
Ton 3	mǎ	ma3
Ton 4	mà	ma4

Note

Für den 5. (neutralen) Ton können Sie Ton 1 verwenden, mit Ausnahme von „er“, das auf Ton 2 abgebildet werden muss. Zum Beispiel würde 打转儿 als „da3-zhuan4-er2“ dargestellt werden.

Chinesische (vereinfacht) benutzerdefinierte Vokabulare verwenden das Feld IPA nicht, aber Sie müssen trotzdem den Header IPA in die benutzerdefinierte Vokabulartabelle aufnehmen.

Es folgt ein Beispiel für eine Eingabedatei im Textformat. Im Beispiel werden zum Ausrichten der Spalten Leerzeichen verwendet. Ihre Eingabedateien sollten die Spalten durch TAB-Zeichen voneinander trennen. Fügen Sie Leerzeichen nur in die Spalte DisplayAs ein.

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

Chinesisch, Mandarin (Taiwan), traditioneller Zeichensatz

Für chinesische (traditionell) benutzerdefinierte Vokabulare kann das Feld Phrase jedes der in der folgenden Datei aufgeführten Zeichen verwenden:

- [zh-tw-Zeichensatz](#)

Das Feld SoundsLike kann die in der folgenden Datei aufgeführten Zhuyin-Silben enthalten:

- [zhuyin-Zeichensatz](#)

Wenn Sie zhuyin-Silben im Feld SoundsLike verwenden, trennen Sie die Silben mit einem Bindestrich (-).

Amazon Transcribe steht für die vier Töne auf Chinesisch (traditionell) unter Verwendung von Zahlen. Die folgende Tabelle zeigt, wie die Tonzeichen für das Wort ㄇㄧ ㄩˊ zugeordnet werden.

Ton	Tonmarkierung	
Ton 1	□Υ	
Ton 2	□Υ´	
Ton 3	□Υ˘	
Ton 4	□Υ`	

Chinesische (traditionell) benutzerdefinierte Vokabulare verwenden das Feld IPA nicht, aber Sie müssen trotzdem die Kopfzeile IPA in die benutzerdefinierte Vokabulartabelle aufnehmen.

Es folgt ein Beispiel für eine Eingabedatei im Textformat. Im Beispiel werden zum Ausrichten der Spalten Leerzeichen verwendet. Ihre Eingabedateien sollten die Spalten durch TAB-Zeichen voneinander trennen. Fügen Sie Leerzeichen nur in die Spalte DisplayAs ein.

Phrase	SoundsLike	IPA	DisplayAs
##	###`-##		
##	###~-##´		
####	###´-##´-##`-##˘		
#####	#`-###`-##´-##˘-###`	###	

Chinesisch, Kantonesisch (Hongkong), Traditioneller Zeichensatz

Für benutzerdefinierte Vokabulare für Chinesisch (Kantonesisch) Hongkong kann das Phrase Feld jedes der in der folgenden Datei aufgeführten Zeichen verwenden:

- [zh-hk-Zeichensatz](#)

Kroatischer Zeichensatz

Für ein kroatisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld Phrase verwenden:

- a-z
- - (Bindestrich)

- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
ć	0107	ď	0111
č	010D	š	0161
ž	017E		

Tschechischer Zeichensatz

Für ein tschechisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
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

Zeichen	Code	Zeichen	Code
ž	017E		

Dänischer Zeichensatz

Für ein dänisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- A-Z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
Å	00C5	æ	00E6
Æ	00C6	é	00E9
Ø	00D8	ø	00F8
å	00E5		

Niederländischer Zeichensatz

Für ein niederländisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- A-Z
- ' (Apostroph)
- - (Bindestrich)

- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

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

Englischer Zeichensatz

Für ein englisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- A-Z
- ' (Apostroph)
- - (Bindestrich)
- . (Punkt)

Estnischer Zeichensatz

Für ein estnisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
ä	00E4	ü	00FC
õ	00F5	š	0161
ö	00F6	ž	017E

Farsi-Zeichensatz

Für ein benutzerdefiniertes Farsi-Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden.

Zeichen	Code	Zeichen	Code
ء	0621	ط	0638
آ	0622	ع	0639
إ	0623	غ	063A
ؤ	0624	ف	0641
ئ	0626	ق	0642
ل	0627	ج	0644

Zeichen	Code	Zeichen	Code
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		

Finnischer Zeichensatz

Für ein finnisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld Phrase verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
ä	00E4	ö	00F6
å	00E5	š	0161
ž	017E		

Französischer Zeichensatz

Für ein französisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- A-Z
- ' (Apostroph)
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
À	00C0	à	00E0
Â	00C2	â	00E2
Ç	00C7	ç	00E7
È	00C8	è	00E8
É	00C9	é	00E9
Ê	00CA	ê	00EA
Ë	00CB	ë	00EB

Zeichen	Code	Zeichen	Code
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

Galizischer Zeichensatz

Für ein galizisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
á	00E1	ñ	00F1
é	00E9	ó	00F3
í	00ED	ú	00FA
ü	00FC		

Georgischer Zeichensatz

Für ein georgisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
ა	10D0	რ	10E0
ბ	10D1	ს	10E1
გ	10D2	ჭ	10E2
დ	10D3	ჟ	10E3
ე	10D4	ვ	10E4
ზ	10D5	ქ	10E5
წ	10D6	ყ	10E6
თ	10D7	შ	10E7
ი	10D8	ჩ	10E8
კ	10D9	ც	10E9
ლ	10DA	ც	10EA
მ	10DB	ძ	10EB
ნ	10DC	წ	10EC
ო	10DD	ჭ	10ED

Zeichen	Code	Zeichen	Code
ð	10DE	þ	10EE
ſ	10DF	Œ	10EF
ƒ	10F0		

Deutscher Zeichensatz

Für ein deutsches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- A-Z
- ' (Apostroph)
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
ä	00E4	Ä	00C4
ö	00F6	Ö	00D6
ü	00FC	Ü	00DC
ß	00DF		

Griechischer Zeichensatz

Für ein griechisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
á	03AC	ν	03BD
έ	03AD	ξ	03BE
ή	03AE	ο	03BF
ί	03AF	π	03C0
ü	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
κ	03BA	ϋ	03CB
λ	03BB	ό	03CC

Zeichen	Code	Zeichen	Code
μ	03BC	ω	03CE
ĩ	0390		

Gujarati-Zeichensatz

Für ein benutzerdefiniertes Gujarati-Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
ઁ	0A81	ઃ	0AA6
ઃ	0A82	ઘ	0AA7
:	0A83	ઙ	0AA8
અ	0A85	પ	0AAA
અલ	0A86	ફ	0AAB
ઈ	0A87	બ	0AAC
ઈ	0A88	ભ	0AAD
ઉ	0A89	મ	0AAE
ઉલ	0A8A	ય	0AAF
જ	0A8B	ર	0AB0

Zeichen	Code	Zeichen	Code
એ	0A8D	લ	0AB2
એ	0A8F	ળ	0AB3
એ	0A90	વ	0AB5
ઓ	0A91	શ	0AB6
ઓ	0A93	ખ	0AB7
ઔ	0A94	સ	0AB8
ક	0A95	હ	0AB9
ખ	0A96	.	0ABC
ગ	0A97	ી	0ABE
ઘ	0A98	િ	0ABF
ક	0A99	ૃ	0AC0
ચ	0A9A	ુ	0AC1
છ	0A9B	ૅ	0AC2
જ	0A9C	૆	0AC3
ઝ	0A9D	ે	0AC5
ઞ	0A9E	ૈ	0AC7
ટ	0A9F	ૉ	0AC8
ઠ	0AA0	ૌ	0AC9
ડ	0AA1	્	0ACB
ઢ	0AA2	૎	0ACC

Zeichen	Code	Zeichen	Code
ɓ	0AA3	ɗ	0ACD
ɔ	0AA4	ƙ	0AD0
ʉ	0AA5	ɓ	0AE0

Hausa-Zeichensatz

Für ein benutzerdefiniertes Hausa-Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
k	0199	ɓ	0253
y	01B4	ɗ	0257
~	0303		

Hebräischer Zeichensatz

Für ein hebräisches benutzerdefiniertes Vokabular können Sie die folgenden Unicode-Zeichen im Feld `Phrase` verwenden.

Zeichen	Code	Zeichen	Code
-	002D	◊	05DD

Zeichen	Code	Zeichen	Code
א	05D0	ח	05DE
ב	05D1	ץ	05DF
ג	05D2	נ	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

Hindi-Zeichensatz

Für ein benutzerdefiniertes Hindi-Vokabular können Sie die folgenden Unicode-Zeichen im Feld Phrase verwenden.

Zeichen	Code	Zeichen	Code
-	002D	थ	0925
.	002E	द	0926
ॐ	0901	ध	0927

Zeichen	Code	Zeichen	Code
.	0902	न	0928
:	0903	प	092A
अ	0905	फ	092B
आ	0906	ब	092C
इ	0907	भ	092D
ई	0908	म	092E
उ	0909	य	092F
ऊ	090A	र	0930
ऋ	090B	ल	0932
ए	090F	व	0935
ऐ	0910	श	0936
ऑ	0911	ष	0937
ओ	0913	स	0938
औ	0914	ह	0939
क	0915	ट	093E
ख	0916	डि	093F
ग	0917	ी	0940
घ	0918	उ	0941
ङ	0919	ॆ	0942
च	091A	ॆ	0943

Zeichen	Code	Zeichen	Code
छ	091B	ॠ	0945
ज	091C	ॡ	0947
झ	091D	ॢ	0948
ञ	091E	ॣ	0949
ट	091F	।	094B
ठ	0920	॥	094C
ड	0921	०	094D
ढ	0922	ॠ	095B
ण	0923	ॡ	095C
त	0924	ॢ	095D

Amazon Transcribe ordnet die folgenden Zeichen zu:

Zeichen	Zuordnung zu
त (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)

Ungarischer Zeichensatz

Für ein ungarisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

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

Isländischer Zeichensatz

Für ein isländisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
á	00E1	ú	00FA
é	00E9	ý	00FD
ð	00F0	þ	00FE
í	00ED	æ	00E6
ó	00F3	ö	00F6

Indonesischer Zeichensatz

Für ein indonesisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- A-Z
- ' (Apostroph)
- - (Bindestrich)
- . (Punkt)

Italienischer Zeichensatz

Für ein italienisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- A-Z
- ' (Apostroph)
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
À	00C0	à	00E0
Ä	00C4	ä	00E4
Ç	00C7	ç	00E7
È	00C8	è	00E8
É	00C9	é	00E9
Ê	00CA	ê	00EA
Ë	00CB	ë	00EB
Ì	00CC	ì	00EC
Ò	00D2	ò	00F2
Ù	00D9	ù	00F9
Ü	00DC	ü	00FC

Japanischer Zeichensatz

Bei benutzerdefinierten japanischen Vokabularen unterstützt das Feld `DisplayAs` alle Hiragana-, Katakana- und Kanji-Zeichen sowie Romaji-Großbuchstaben in voller Breite.

Das Feld `Phrase` unterstützt die in der folgenden Datei aufgeführten Zeichen:

- [ja-jp-Zeichensatz](#)

Kabyllischer Zeichensatz

Für ein kabyllisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z

- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

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

Kannada-Zeichensatz

Für ein benutzerdefiniertes Kannada-Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
ೠ	0C82	ೡ	0CA7
ೢ	0C83	ೣ	0CA8
೥	0C85	೦	0CAA

Zeichen	Code	Zeichen	Code
ಆ	0C86	ಛ	0CAB
ಇ	0C87	ಞ	0CAC
ಈ	0C88	ಝ	0CAD
ಉ	0C89	ಞ್	0CAE
ಊ	0C8A	ಯ	0CAF
ಋ	0C8B	ರ	0CB0
ಎ	0C8E	ಲಿ	0CB2
ಏ	0C8F	ಳ	0CB3
ಐ	0C90	ಲೆ	0CB5
ಒ	0C92	ಲ	0CB6
ಓ	0C93	ಲೆ	0CB7
ಔ	0C94	ನ	0CB8
ಕ	0C95	ನಿ	0CB9
ಖ	0C96	#	0CBC
ಗ	0C97	#	0CBD
ಘ	0C98	ವ	0CBE
ಙ	0C99	ವಿ	0CBF
ಚ	0C9A	ವೀ	0CC0
ಛ	0C9B	ವಿ	0CC1
ಜ	0C9C	ವಿ	0CC2

Zeichen	Code	Zeichen	Code
ꠘ	0C9D	ꠘ	0CC3
ꠙ	0C9E	ꠙ	0CC6
ꠚ	0C9F	ꠚ	0CC7
ꠛ	0CA0	ꠛ	0CC8
ꠜ	0CA1	ꠜ	0CCA
ꠝ	0CA2	ꠝ	0CCB
ꠞ	0CA3	ꠞ	0CCC
ꠟ	0CA4	ꠟ	0CCD
ꠠ	0CA5	ꠠ	0CD5
ꠡ	0CA6	ꠡ	0CD6
ꠢ	0CE0		

Kasachischer Zeichensatz

Für ein kasachisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
Ғ	0442	Ы	044B

Zeichen	Code	Zeichen	Code
б	0431	я	044F
о	043E	с	0441
п	043F	h	04BB
ш	0448	д	0434
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

Kinyarwanda-Zeichensatz

Für ein benutzerdefiniertes Kinyarwanda-Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
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

Koreanischer Zeichensatz

Für benutzerdefinierte koreanische Vokabeln können Sie in dem `Phrase` Feld Folgendes verwenden:

- a-z
- A-Z

- - (Bindestrich)
- . (Punkt)

Das `Phrase` Feld kann jede der im folgenden Link aufgeführten Hangul-Silben verwenden:

- [Hangul-Silben](#) auf Wikipedia.

Kirgisischer Zeichensatz

Für ein kirgisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
а	0430	љ	0459
б	0431	њ	045A
в	0432	ћ	045B
г	0433	ќ	045C
д	0434	#	045D
е	0435	ђ	045E
ж	0436	џ	045F
з	0437	ѓ	0491
и	0438	ѣ	0493

Zeichen	Code	Zeichen	Code
й	0439	ж	0497
к	043A	з	0499
л	043B	к	049B
м	043C	к	049F
н	043D	к	04A1
о	043E	ң	04A3
п	043F	п	04A5
р	0440	р	04A9
с	0441	с	04AB
т	0442	т	04AD
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

Zeichen	Code	Zeichen	Code
э	044D	ä	04D3
ю	044E	ě	04D7
я	044F	ə	04D9
#	0450	з	04E1
ë	0451	й	04E3
ђ	0452	ö	04E7
í	0453	ө	04E9
є	0454	ȳ	04EF
s	0455	ÿ	04F1
i	0456	ý	04F3
ï	0457	#	04F7
j	0458	Ы	04F9

Lettischer Zeichensatz

Für ein lettisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

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

Litauischer Zeichensatz

Für ein litauisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld **Phrase** verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld **Phrase** verwenden:

Zeichen	Code	Zeichen	Code
ą	0105	į	012F
č	010D	š	0161
ę	0119	ų	0173
ė	0117	ū	016B
ž	017E		

Luganda-Zeichensatz

Für ein benutzerdefiniertes Luganda-Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
ÿ	00FF	ŋ	014B

Mazedonischer Zeichensatz

Für ein mazedonisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
а	0430	љ	0459
б	0431	њ	045A
в	0432	ћ	045B
г	0433	ќ	045C

Zeichen	Code	Zeichen	Code
д	0434	#	045D
е	0435	ŷ	045E
ж	0436	ц	045F
з	0437	г	0491
и	0438	ѣ	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	џ	04A9
с	0441	џ	04AB
т	0442	џ	04AD
у	0443	џ	04AF
ф	0444	џ	04B1
х	0445	х	04B3
ц	0446	ц	04B5
ч	0447	ч	04B7

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

Malaiischer Zeichensatz

Für ein malaiisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld **Phrase** verwenden:

- a-z
- A-Z
- ' (Apostroph)
- - (Bindestrich)
- . (Punkt)

Malayalam-Zeichensatz

Für ein benutzerdefiniertes Malayalam-Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
ഠ	0D02	ന	0D28
ഃ	0D03	പ	0D2A
അ	0D05	ഫ	0D2B
ആ	0D06	ബ	0D2C
ഇ	0D07	ഭ	0D2D
ഈ	0D08	മ	0D2E
ഉ	0D09	യ	0D2F
ഊ	0D0A	ര	0D30
ഋ	0D0B	റ	0D31
എ	0D0E	ല	0D32

Zeichen	Code	Zeichen	Code
ഏ	0D0F	ഉ	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	ഓ	0D3E
ഘ	0D18	റ	0D3F
ങ	0D19	ീ	0D40
ച	0D1A	ു	0D41
ഛ	0D1B	ൂ	0D42
ജ	0D1C	്യ	0D43
ട	0D1D	െ	0D46
ണ	0D1E	േ	0D47
ട	0D1F	ൈ	0D48
റ	0D20	ൊ	0D4A
ഡ	0D21	ോ	0D4B
ഢ	0D22	ൌ	0D4C
ണ	0D23	ൿ	0D4D

Zeichen	Code	Zeichen	Code
Ⓞ	0D24	#	0D7A
↳	0D25	#	0D7B
β	0D26	#	0D7C
ω	0D27	#	0D7D

Maltesischer Zeichensatz

Für ein maltesisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
à	00E0	ù	00F9
è	00E8	ć	010B
ì	00EC	ġ	0121
ò	00F2	ħ	0127
ż	017C		

Marathi-Zeichensatz

Für ein benutzerdefiniertes Marathi-Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
ॐ	0901	थ	0925
.	0902	द	0926
:	0903	ध	0927
अ	0905	न	0928
आ	0906	प	092A
इ	0907	फ	092B
ई	0908	ब	092C
उ	0909	भ	092D
ऊ	090A	म	092E
ऋ	090B	य	092F
ॠ	090D	र	0930
ए	090F	ल	0932
ऐ	0910	ळ	0933
ऑ	0911	व	0935
ओ	0913	श	0936
औ	0914	ष	0937

Zeichen	Code	Zeichen	Code
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
त	0924	॑	094D
ॐ	0950		

Meadow Mari-Zeichensatz

Für ein benutzerdefiniertes Meadow Mari-Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
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	жџ	0497
к	043A	џ	0499
л	043B	ќ	049B
м	043C	к	049F
н	043D	т	04A1
о	043E	ћ	04A3
п	043F	н	04A5

Zeichen	Code	Zeichen	Code
p	0440	q	04A9
c	0441	ç	04AB
т	0442	ṭ	04AD
y	0443	ŷ	04AF
ϕ	0444	ϣ	04B1
x	0445	χ	04B3
ц	0446	ƚ	04B5
ч	0447	ƚ	04B7
ш	0448	h	04BB
щ	0449	ƚ	04BD
ъ	044A	ƚ	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

Zeichen	Code	Zeichen	Code
€	0454	ȳ	04EF
s	0455	ÿ	04F1
i	0456	ŷ	04F3
ï	0457	#	04F7
j	0458	Ы	04F9

Mongolischer Zeichensatz

Für ein mongolisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
а	0430	лъ	0459
б	0431	нь	045A
в	0432	һ	045B
г	0433	ќ	045C
д	0434	#	045D
е	0435	ÿ	045E
ж	0436	ұ	045F

Zeichen	Code	Zeichen	Code
з	0437	г	0491
и	0438	ф	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	р	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
ч	0447	ч	04B7
ш	0448	h	04BB
щ	0449	е	04BD
ъ	044A	ё	04BF

Zeichen	Code	Zeichen	Code
Ы	044B	#	04CA
Ь	044C	ă	04D1
Э	044D	ä	04D3
Ю	044E	ě	04D7
Я	044F	ə	04D9
#	0450	з	04E1
ë	0451	й	04E3
ђ	0452	ö	04E7
í	0453	ө	04E9
є	0454	ȳ	04EF
s	0455	ÿ	04F1
i	0456	ý	04F3
ï	0457	#	04F7
j	0458	Ы	04F9

Norwegischer Bokmål-Zeichensatz

Für ein benutzerdefiniertes norwegisches Bokmål-Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
å	00E5	æ	00E6
ø	00F8		

Odia/Oriya Zeichensatz

Für Odia/Oriya benutzerdefinierte Vokabeln können Sie die folgenden Zeichen in dem `Phrase` Feld verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
ୠ	0B01	ୡ	0B26
ୢ	0B02	ୣ	0B27
୦	0B03	ୠ	0B28
ୡ	0B05	ୢ	0B2A
ୣ	0B06	୥	0B2B
୦	0B07	୧	0B2C
ୡ	0B08	ୢ	0B2D
ୣ	0B09	୥	0B2E
୦	0B0A	୧	0B2F

Zeichen	Code	Zeichen	Code
ର	0B0B	ର	0B30
ଏ	0B0F	ଲ	0B32
ଏଁ	0B10	ଳ	0B33
ଓ	0B13	ଶ	0B36
ଓଁ	0B14	ଷ	0B37
କ	0B15	ସ	0B38
ଖ	0B16	ହ	0B39
ଗ	0B17	.	0B3C
ଘ	0B18		0B3E
ଙ	0B19	~	0B3F
ଚ	0B1A	୧	0B40
ଛ	0B1B	୨	0B41
ଜ	0B1C	୩	0B42
ଝ	0B1D	୪	0B43
ଞ	0B1E	୫	0B47
ଟ	0B1F	୬	0B48
ଠ	0B20	୬#	0B4B
ଡ	0B21	୬ୱ	0B4C
ଢ	0B22	୬	0B4D
ଣ	0B23	୬	0B56

Zeichen	Code	Zeichen	Code
ᱠ	0B24	ᱡ	0B5F
ᱢ	0B25	ᱣ	0B60
#	0B71		

Paschtu-Zeichensatz

Für ein benutzerdefiniertes Paschtu-Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
Ā	0622	و	0648
Ā̄	0623	ښ	064A
ؤ	0624	ښ	064B
ئ	0626	ښ	064C
ل	0627	ښ	064D
پ	0628	ښ	064E
ت	062A	ښ	064F
ث	062B	ښ	0650
ج	062C	ښ	0651

Zeichen	Code	Zeichen	Code
ح	062D	°	0652
خ	062E	#	0654
د	062F	'	0670
ذ	0630	ټ	067C
ر	0631	پ	067E
ز	0632	خ	0681
س	0633	څ	0685
ش	0634	چ	0686
ص	0635	ډ	0689
ض	0636	ړ	0693
ط	0637	ښ	0696
ظ	0638	ژ	0698
ع	0639	ښ	069A
غ	063A	ک	06A9
ف	0641	ګ	06AB
ق	0642	ګ	06AF
ل	0644	ښ	06BC
م	0645	ی	06CC
ن	0646	ی	06CD
ه	0647	ې	06D0

Polnischer Zeichensatz

Für ein polnisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

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

Portugiesischer Zeichensatz

Für ein portugiesisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- A-Z
- ' (Apostroph)
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

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

Pandschabi-Zeichensatz

Für ein benutzerdefiniertes Pandschabi-Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
ਅ	0A05	ਧ	0A27
ਆ	0A06	ਨ	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

Zeichen	Code	Zeichen	Code
ਙ	0A19	ਟ	0A3E
ਚ	0A1A	ਫ	0A3F
ਛ	0A1B	ਭ	0A40
ਜ	0A1C	ਲ	0A41
ਝ	0A1D	ੲ	0A42
ਞ	0A1E	ਯ	0A47
ਟ	0A1F	ਰ	0A48
ਠ	0A20	ਕ	0A4B
ਡ	0A21	ਖ	0A4C
ਢ	0A22	ਗ	0A4D
ਣ	0A23	ਘ	0A5C
ਤ	0A24	ੰ	0A70
ਥ	0A25	ੰ	0A71
ਦ	0A26	ੲ	0A72
ਉ	0A73		

Rumänischer Zeichensatz

Für ein rumänisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld Phrase verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

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

Russischer Zeichensatz

Für ein russisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
'	0027	п	043F
-	002D	р	0440
.	002E	с	0441
а	0430	т	0442
б	0431	у	0443
в	0432	ф	0444
г	0433	х	0445
д	0434	ц	0446
е	0435	ч	0447
ж	0436	ш	0448
з	0437	щ	0449

Zeichen	Code	Zeichen	Code
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

Serbischer Zeichensatz

Für ein serbisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
ć	0107	ı	0456
č	010D	ï	0457
đ	0111	j	0458
š	0161	љ	0459
ž	017E	њ	045A

Zeichen	Code	Zeichen	Code
а	0430	ћ	045B
б	0431	ќ	045C
в	0432	#	045D
г	0433	ђ	045E
д	0434	џ	045F
е	0435	ѓ	0491
ж	0436	ѣ	0493
з	0437	жѝ	0497
и	0438	џ	0499
й	0439	ќ	049B
к	043A	к	049F
л	043B	ќ	04A1
м	043C	њ	04A3
н	043D	н	04A5
о	043E	џ	04A9
п	043F	џ	04AB
р	0440	џ	04AD
с	0441	џ	04AF
т	0442	џ	04B1
у	0443	х	04B3

Zeichen	Code	Zeichen	Code
ф	0444	ц	04B5
х	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	ȳ	04EF
í	0453	ÿ	04F1
є	0454	ŷ	04F3
s	0455	#	04F7
ß	04F9		

Singhalesischer Zeichensatz

Für ein singhalesisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
#	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

Zeichen	Code	Zeichen	Code
#	0D94	#	0DC1
#	0D95	#	0DC2
#	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

Zeichen	Code	Zeichen	Code
#	0DAC	##	0DDE
#	0DAD	#	0DDF
#	0DAE	#	0DF2

Slowakischer Zeichensatz

Für ein slowakisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
á	00E1	ň	0148
ä	00E4	ó	00F3
č	010D	ô	00F4
ď	010F	í	0155
é	00E9	š	0161
í	00ED	ť	0165
í	013A	ú	00FA
ĭ	013E	ý	00FD
ž	017E		

Slowenischer Zeichensatz

Für ein slowenisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
č	010D	š	0161
ž	017E		

Somalischer Zeichensatz

Für ein somalisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
S	0073	d	0064
t	0074	a	0061
a	0061	r	0072

Zeichen	Code	Zeichen	Code
n	006E	d	0064

Spanischer Zeichensatz

Für ein spanisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden.

- a-z
- A-Z
- ' (Apostroph)
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
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		

Sundanesischer Zeichensatz

Für ein sundanesisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
S	0073	d	0064
t	0074	a	0061
a	0061	r	0072
n	006E	d	0064

Swahili-Zeichensatz

Für ein benutzerdefiniertes Swahili-Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
S	0073	d	0064

Zeichen	Code	Zeichen	Code
t	0074	a	0061
a	0061	r	0072
n	006E	d	0064

Schwedischer Zeichensatz

Für ein schwedisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- A-Z
- ' (Apostroph)
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
Ä	00C4	ä	00E4
Å	00C5	å	00E5
Ö	00D6	ö	00F6

Tagalog/Filipino Zeichensatz

Für Tagalog/Filipino benutzerdefinierte Vokabeln können Sie die folgenden Zeichen in dem `Phrase` Feld verwenden:

- a-z
- - (Bindestrich)

- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code		
ñ	00F1		

Tamilischer Zeichensatz

Für ein tamilisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

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

Zeichen	Code	Zeichen	Code
◌̣	0B83	◌̣	0BCD
Ҁ	0B95	Ҁ	0BBE
ҁ	0B99	ҁ	0BBF
҂	0B9A	҂	0BC0
҃	0B9E	҃	0BC1
҄	0B9F	҄	0BC2
҅	0BA3	҅	0BC6
҆	0BA4	҆	0BC7
҇	0BA8	҇	0BC8
҈	0BAA	҈	0BCA
҉	0BAE	҉	0BCB
Ҋ	0BAF	Ҋ	0BCC

Tatarischer Zeichensatz

Für ein tatarisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
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	жџ	0497
к	043A	џ	0499
л	043B	ќ	049B
м	043C	к	049F
н	043D	т	04A1
о	043E	ћ	04A3
п	043F	н	04A5
р	0440	ѣ	04A9
с	0441	џ	04AB
т	0442	т	04AD
у	0443	у	04AF

Zeichen	Code	Zeichen	Code
ф	0444	ү	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
я	044F	ə	04D9
#	0450	з	04E1
ë	0451	й	04E3
ђ	0452	ö	04E7
í	0453	ө	04E9
є	0454	ȳ	04EF
s	0455	ÿ	04F1
i	0456	ÿ	04F3
ï	0457	#	04F7

Zeichen	Code	Zeichen	Code
j	0458	ఱ	04F9

Telugu-Zeichensatz

Für ein benutzerdefiniertes Telugu-Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

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

Zeichen	Code	Zeichen	Code
ప	0C35	ఓ	0C13
శ	0C36	ఔ	0C14
ష	0C37	క	0C15
స	0C38	ఖ	0C16
హ	0C39	గ	0C17
ఱ	0C3E	ఘ	0C18
ళ	0C3F	ఙ	0C19
ఠ	0C40	చ	0C1A
డ	0C41	ఛ	0C1B
ణ	0C42	జ	0C1C
ట	0C43	ఝ	0C1D
ఢ	0C44	భ	0C1E
న	0C47	ట	0C1F
ని	0C48	థ	0C20
నీ	0C4A	డ	0C21
నె	0C4B	ధ	0C22
నో	0C4C	ణ	0C23
న్	0C4D		

Thailändischer Zeichensatz

Für ein benutzerdefiniertes Thai-Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
ก	0E01	ล	0E25
ข	0E02	ฬ	0E26
ฃ	0E03	ว	0E27
ค	0E04	ศ	0E28
ฅ	0E05	ษ	0E29
ฆ	0E06	ส	0E2A
ง	0E07	ห	0E2B
จ	0E08	ฬ	0E2C
ฉ	0E09	อ	0E2D
ช	0E0A	ฮ	0E2E
ฌ	0E0B	า	0E2F
ฎ	0E0C	ะ	0E30
ญ	0E0D	ั	0E31
ฎ	0E0E	า	0E32

Zeichen	Code	Zeichen	Code
ฏ	0E0F	ด	0E34
ฐ	0E10	ด	0E35
ท	0E11	ด	0E36
ฒ	0E12	ด	0E37
ณ	0E13	,	0E38
ด	0E14	น	0E39
ต	0E15	.	0E3A
ถ	0E16	เ	0E40
ท	0E17	แ	0E41
ธ	0E18	โ	0E42
น	0E19	ใ	0E43
บ	0E1A	ใ	0E44
ป	0E1B	จ	0E45
ผ	0E1C	จ	0E46
ฝ	0E1D	ช	0E47
พ	0E1E	'	0E48
ฟ	0E1F	ช	0E49
ภ	0E20	ช	0E4A
ม	0E21	+	0E4B
ย	0E22	ย	0E4C

Zeichen	Code	Zeichen	Code
ı	0E23	•	0E4D
î	0E24		

Türkischer Zeichensatz

Für ein türkisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- A-Z
- ' (Apostroph)
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
Ç	00C7	ö	00F6
Ö	00D6	û	00FB
Ü	00DC	ü	00FC
â	00E2	Ğ	011E
ä	00E4	ğ	011F
ç	00E7	İ	0130
è	00E8	ı	0131
é	00E9	Ş	015E
ê	00EA	ş	015F

Zeichen	Code	Zeichen	Code
í	00ED	š	0161
î	00EE	ž	017E
ó	00F3		

Ukrainischer Zeichensatz

Für ein ukrainisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
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

Zeichen	Code	Zeichen	Code
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
ѓ	0491		

Uigurischer Zeichensatz

Für ein uigurisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld Phrase verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld Phrase verwenden:

Zeichen	Code	Zeichen	Code
#	0611	و	0648
#	0613	س	0649
#	0614	ش	064A
ع	0621	=	064B

Zeichen	Code	Zeichen	Code
Ā	0622	𐰀	064C
ǣ	0623	𐰁	064D
ǿ	0624	𐰂	064E
ǣ	0625	𐰃	064F
ǿ	0626	𐰄	0650
ǣ	0627	𐰅	0651
ǿ	0628	𐰆	0652
ǿ	0629	#	0653
𐰇	062A	#	0654
𐰈	062B	#	0657
𐰉	062C	𐰇	0670
𐰊	062D	𐰈	0679
𐰋	062E	𐰉	067A
𐰌	062F	𐰊	067B
𐰍	0630	𐰋	067C
𐰎	0631	𐰌	067D
𐰏	0632	𐰍	067E
𐰐	0633	𐰎	067F
𐰑	0634	𐰏	0680
𐰒	0635	𐰐	0681

Zeichen	Code	Zeichen	Code
ض	0636	چ	0683
ط	0637	چ	0684
ظ	0638	خ	0685
ع	0639	چ	0686
غ	063A	چ	0687
-	0640	ڭ	0688
ف	0641	ڊ	0689
ق	0642	ڊ	068A
ك	0643	ڌ	068C
ل	0644	ڍ	068D
م	0645	ڏ	068F
ن	0646	ڙ	0691
ه	0647	ر	0693
ر	0695		

Usbekischer Zeichensatz

Für ein usbekisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
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
х	0445	ф	0444
ц	0446	к	043A
м	043C	у	0443
ь	044C	ж	0436
ъ	044A	ғ	0493
а	0430		

Vietnamesischer Zeichensatz

Amazon Transcribe steht für die sechs Töne auf Vietnamesisch mit Zahlen. Die folgende Tabelle zeigt, wie Tonzeichen für das Wort „ma“ abgebildet werden.

Name des Klangs	Tonmarkierung	Tonnummer
ngang	MA	ma1
sắc	má	ma2
huyền	mà	ma3
hỏi	mả	ma4
ngã	mã	ma5
nặng	mạ	ma6

Für ein vietnamesisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- A-Z
- ' (Apostroph)
- - (Bindestrich)
- . (Punkt)
- & (kaufmännisches Und)
- ; (Semikolon)
- _ (Unterstrich)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
à	00E0	À	00C0

Zeichen	Code	Zeichen	Code
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
ă	0103	Ă	0102
đ	0111	Đ	0110
ĩ	0129	Ĩ	0128
ũ	0169	Ũ	0168
ơ	01A1	Ơ	01A0

Zeichen	Code	Zeichen	Code
ư	01B0	Ư	01AF
ạ	1EA1	Ạ	1EA0
ả	1EA3	Ả	1EA2
ấ	1EA5	Ấ	1EA4
ầ	1EA7	Ầ	1EA6
ẩ	1EA9	Ổ	1EA8
ẫ	1EAB	Ẫ	1EAA
ậ	1EAD	Ậ	1EAC
ắ	1EAF	Ắ	1EAE
ằ	1EB1	Ằ	1EB0
ẳ	1EB3	Ẳ	1EB2
ẵ	1EB5	Ẵ	1EB4
ặ	1EB7	Ặ	1EB6
ẹ	1EB9	Ẹ	1EB8
ẻ	1EBB	Ẻ	1EBA
ẽ	1EBD	Ẽ	1EBC
ể	1EBF	Ể	1EBE
ề	1EC1	Ề	1EC0
ễ	1EC3	Ễ	1EC2
ễ	1EC5	Ễ	1EC4

Zeichen	Code	Zeichen	Code
ê	1EC7	Ê	1EC6
ỉ	1EC9	Ỉ	1EC8
ị	1ECB	Ị	1ECA
ơ	1ECD	Ơ	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
ụ	1EE5	Ụ	1EE4
ủ	1EE7	Ủ	1EE6
ứ	1EE9	Ứ	1EE8
ừ	1EEB	Ừ	1EEA
ử	1EED	Ử	1EEC

Zeichen	Code	Zeichen	Code
ŭ	1EEF	Ů	1EEE
ų	1EF1	Ū	1EF0
ỳ	1EF3	Ỡ	1EF2
ყ	1EF5	ყ	1EF4
Ỳ	1EF7	Ỳ	1EF6
ỹ	1EF9	Ỹ	1EF8

Walisischer Zeichensatz

Für ein walisches benutzerdefiniertes Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
à	00E0	ò	00F2
á	00E1	ó	00F3
â	00E2	ô	00F4
ä	00E4	ö	00F6
è	00E8	ù	00F9
é	00E9	ú	00FA

Zeichen	Code	Zeichen	Code
ê	00EA	û	00FB
ë	00EB	ü	00FC
ì	00EC	ý	00FD
í	00ED	ÿ	00FF
î	00EE	ŵ	0175
ï	00EF	ÿ	0177
ÿ	1EF3		

Wolof-Zeichensatz

Für ein benutzerdefiniertes Wolof-Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
à	00E0	ê	00EA
ã	00E3	ë	00EB
ç	00E7	ñ	00F1
è	00E8	ó	00F3
é	00E9	ô	00F4

Zeichen	Code	Zeichen	Code
ŋ	014B		

Zulu-Zeichensatz

Für ein benutzerdefiniertes Zulu-Vokabular können Sie die folgenden Zeichen im Feld `Phrase` verwenden:

- a-z
- - (Bindestrich)
- . (Punkt)

Sie können auch die folgenden Unicode-Zeichen im Feld `Phrase` verwenden:

Zeichen	Code	Zeichen	Code
S	0073	d	0064
t	0074	a	0061
a	0061	r	0072
n	006E	d	0064

Wie Amazon Transcribe funktioniert

Amazon Transcribe verwendet Modelle für maschinelles Lernen, um Sprache in Text umzuwandeln.

Zusätzlich zum transkribierten Text enthalten die Transkripte Daten über den transkribierten Inhalt, einschließlich Konfidenzwerte und Zeitstempel für jedes Wort oder Satzzeichen. Ein Beispiel für die Ausgabe finden Sie im Abschnitt [Dateneingabe und -ausgabe](#). Eine vollständige Liste der Features, die Sie auf Ihre Transkription anwenden können, finden Sie in der [Featureübersicht](#).

Die Transkriptionsmethoden lassen sich in zwei Hauptkategorien einteilen:

- Batch-Transkriptionen: Transcribe Sie Mediendateien, die in einen Bucket hochgeladen wurden. Amazon S3 Sie können die [AWS CLI](#), [AWS-Managementkonsole](#) und verschiedene [AWS -SDKs](#) für Batch-Transkriptionen verwenden.
- Streaming-Transkriptionen: Medien-Streams in Echtzeit transkribieren. Sie können die [AWS-Managementkonsole](#), und verschiedene [AWS SDKs](#) zum [HTTP/2WebSockets](#) Streamen von Transkriptionen verwenden.

Beachten Sie, dass sich die Unterstützung von Features und Sprachen bei Batch- und Streaming-Transkriptionen unterscheidet. Weitere Informationen finden Sie unter [Amazon Transcribe features](#) und [Unterstützte Sprachen](#).

Topics

- [Dateneingabe und -ausgabe](#)
- [Transkribieren von Zahlen und Interpunktion](#)

API-Vorgänge für den Einstieg

Batch: [StartTranscriptionJob](#)

Streamen: [StartStreamTranscription](#) StartStreamTranscriptionWebSocket

Dateneingabe und -ausgabe

Amazon Transcribe nimmt Audiodaten als Mediendatei in einem Amazon S3 Bucket oder einem Medienstream auf und konvertiert sie in Textdaten.

Wenn Sie Mediendateien transkribieren, die in einem Amazon S3 Bucket gespeichert sind, führen Sie Batch-Transkriptionen durch. Wenn Sie Medien-Streams transkribieren, führen Sie Streaming-Transkriptionen durch. Für diese beiden Verfahren gelten unterschiedliche Regeln und Anforderungen.

Bei Batch-Transkriptionen können Sie [Auftragswarteschlangen](#) verwenden, wenn Sie nicht alle Ihre Transkriptionsaufträge gleichzeitig bearbeiten müssen. Auf diese Weise können Amazon Transcribe Sie Ihre Transkriptionsaufträge verfolgen und sie bearbeiten, wenn Slots verfügbar sind.

Note

Amazon Transcribe kann Ihre Inhalte vorübergehend speichern, um die Qualität seiner Analysemodelle kontinuierlich zu verbessern. Weitere Informationen finden Sie in den [Häufig gestellten Fragen zu Amazon Transcribe](#). Um die Löschung von Inhalten zu beantragen, die möglicherweise von gespeichert wurden Amazon Transcribe, eröffnen Sie einen Fall bei [Support](#).

Topics

- [Medienformate](#)
- [Audiokanäle](#)
- [Samplerate](#)
- [Ausgabe](#)

Medienformate

Die unterstützten Medientypen unterscheiden sich zwischen Batch-Transkriptionen und Streaming-Transkriptionen, wobei verlustfreie Formate für beide empfohlen werden. Einzelheiten sind der folgenden Tabelle zu entnehmen:

	Batch	Streaming
Unterstützte Formate	• AMR • FLAC • M4A	• FLAC • Ogg Opus • PCM-Kodierung

	Batch	Streaming
	• MP3 • MP4 • ogg • WebM • WAV	
Empfohlene Formate	• FLAC • WAV mit PCM 16-Bit-Kodierung	• FLAC • PCM signiertes 16-Bit Little-Endian-Audio (beachten Sie, dass dies nicht WAV beinhaltet)

Die besten Ergebnisse erzielen Sie, wenn Sie ein verlustfreies Format wie FLAC oder WAV mit PCM 16-Bit-Kodierung verwenden.

Note

Streaming-Transkriptionen werden nicht in allen Sprachen unterstützt. Einzelheiten finden Sie in der Spalte „Dateneingabe“ in der [Tabelle „Unterstützte Sprachen“](#).

Audiokanäle

Amazon Transcribe unterstützt Einkanal- und Zweikanal-Medien. Medien mit mehr als zwei Kanälen werden derzeit nicht unterstützt.

Wenn Ihr Audiomaterial mehrere Sprecher auf einem Kanal enthält und Sie die einzelnen Sprecher in Ihrer Transkriptionsausgabe partitionieren und beschriften möchten, können Sie die [Sprecherpartitionierung \(Diarisierung\)](#) verwenden.

Wenn Ihr Audiomaterial Sprache auf zwei separaten Kanälen enthält, können Sie [die Kanalidentifikation](#) verwenden, um jeden Kanal separat in Ihrem Transkript zu transkribieren.

Beide Optionen erzeugen eine einzige Transkriptionsdatei.

 Note

Wenn Sie die [Sprecherpartitionierung](#) oder die [Kanalidentifizierung](#) nicht aktivieren, wird Ihr Transkripttext als ein durchgehender Abschnitt bereitgestellt.

Samplerate

Bei Batch-Transkriptionsaufträgen können Sie eine Samplerate angeben, dieser Parameter ist jedoch optional. Wenn Sie dies in Ihrer Anfrage angeben, vergewissern Sie sich, dass der von Ihnen angegebene Wert mit der tatsächlichen Samplerate in Ihrem Audiomaterial übereinstimmt. Wenn Sie eine Samplerate angeben, die nicht zu Ihrem Audiomaterial passt, kann Ihr Auftrag fehlschlagen.

Bei Streaming-Transkriptionen müssen Sie in Ihrem Antrag eine Samplerate angeben. Wie bei Batch-Transkriptionsaufträgen müssen Sie sicherstellen, dass der von Ihnen angegebene Wert mit der tatsächlichen Samplerate in Ihrem Audio übereinstimmt.

Sampleraten für Low-Fidelity-Audio, wie z. B. Telefonaufnahmen, verwenden normalerweise 8.000 Hz. Amazon Transcribe unterstützt Werte zwischen 16.000 Hz und 48.000 Hz für Hi-Fidelity-Audio.

Ausgabe

Die Transkriptionsausgabe erfolgt im JSON-Format. Der erste Teil Ihrer Abschrift enthält das eigentliche Transkript in Absatzform, gefolgt von zusätzlichen Daten für jedes Wort und Satzzeichen. Die bereitgestellten Daten hängen von den Features ab, die Sie in Ihrer Anfrage angeben. Ihr Transkript enthält mindestens die Startzeit, die Endzeit und den Konfidenzwert für jedes Wort. Der [folgende Abschnitt](#) zeigt ein Beispiel für die Ausgabe einer einfachen Transkriptionsanfrage, die keine zusätzlichen Optionen oder Features enthielt.

Alle Batch-Transkripte werden in Buckets gespeichert. Amazon S3 Sie können wählen, ob Sie Ihr Transkript in Ihrem eigenen Amazon S3 Bucket speichern oder einen sicheren Standard-Bucket Amazon Transcribe verwenden möchten. Weitere Informationen zum Erstellen und Verwenden von Amazon S3 -Buckets finden Sie unter [Arbeiten mit Buckets](#).

Wenn Sie möchten, dass Ihr Transkript in einem Amazon S3 Bucket gespeichert wird, den Sie besitzen, geben Sie die URI des Buckets in Ihrer Transkriptionsanfrage an. Stellen Sie sicher, dass Sie Amazon Transcribe Schreibberechtigungen für diesen Bucket erteilen, bevor Sie mit der Batch-Transkription beginnen. Wenn Sie Ihren eigenen Eimer angeben, verbleibt Ihr Transkript in diesem Bucket, bis Sie es entfernen.

Wenn Sie keinen Amazon S3 Bucket angeben, Amazon Transcribe verwendet es einen sicheren, vom Service verwalteten Bucket und stellt Ihnen eine temporäre URI zur Verfügung, die Sie zum Herunterladen Ihres Transkripts verwenden können. Beachten Sie, dass temporäre URIs nur 15 Minuten lang gültig sind. Wenn Sie bei der Verwendung der angegebenen URI eine `AccessDenied`-Fehlermeldung erhalten, stellen Sie eine `GetTranscriptionJob`-Anfrage, um eine neue temporäre URI für Ihr Transkript zu erhalten.

Wenn Sie sich für einen Standard-Bucket entscheiden, wird Ihr Transkript gelöscht, wenn Ihr Auftrag abläuft (90 Tage). Wenn Sie Ihr Transkript nach diesem Ablaufdatum aufbewahren möchten, müssen Sie es herunterladen.

Streaming-Transkripte werden über dieselbe Methode zurückgegeben, die Sie für Ihren Stream verwenden.

Tip

[Wenn Sie Ihre JSON-Ausgabe in ein Turn-by-Turn-Transkript im Word-Format konvertieren möchten, sehen Sie sich dieses GitHub Beispiel an \(für Python3\).](#) Dieses Skript funktioniert mit Protokollen von Analyse nach Anrufen und Standard-Batch-Protokollen mit aktivierter Diarisierung.

Beispielausgabe

Transkripte bieten eine vollständige Transkription in Absatzform, gefolgt von einer Wort-für-Wort-Aufschlüsselung, die Daten für jedes Wort und jedes Satzzeichen enthält. Dazu gehören die Startzeit, die Endzeit, ein Vertrauenswert und ein Typ (`pronunciation` oder `punctuation`).

Das folgende Beispiel stammt von einem einfachen Batch-Transkriptionsauftrag, der keine [zusätzlichen Features](#) enthielt. Mit jedem zusätzlichen Feature, die Sie auf Ihre Transkriptionsanfrage anwenden, erhalten Sie zusätzliche Daten in Ihrer Transkriptionsausgabedatei.

Einfache Batch-Transkripte enthalten zwei Hauptabschnitte:

1. `transcripts`: enthält das gesamte Transkript in einem Textblock.
2. `items`: Enthält Informationen zu jedem Wort und Satzzeichen aus dem `transcripts`-Abschnitt.
3. `audio_segments`: Ein Audiosegment ist ein bestimmter Teil einer Audioaufnahme, der ununterbrochen gesprochene Sprache mit minimalen Pausen oder Pausen enthält. Dieses

Segment erfasst einen natürlichen Sprachfluss und wird `audio_segments` mit einer Start- und Endzeit aufgenommen. Das `items` Element innerhalb eines Audiosegments ist eine Folge von Kennungen, die jedem Element innerhalb des Segments entsprechen.

Jedes zusätzliche Feature, das Sie in Ihre Transkriptionsanfrage aufnehmen, führt zu zusätzlichen Informationen in Ihrem Transkript.

```
{
  "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"  
}
```

Transkribieren von Zahlen und Interpunktion

Amazon Transcribe fügt allen unterstützten Sprachen automatisch Satzzeichen hinzu und schreibt Wörter entsprechend groß für Sprachen, die in ihren Schriftsystemen die Unterscheidung zwischen Groß- und Kleinschreibung verwenden.

In den meisten Sprachen werden die Zahlen in ihre Wortformen transkribiert. Bei Sprachen, die das Transkribieren von Zahlen unterstützen, werden Zahlen jedoch je nach Kontext, in dem sie verwendet werden, unterschiedlich Amazon Transcribe behandelt.

Wenn ein Sprecher zum Beispiel „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“ sagt, wird dies transkribiert als:

- Sprachen mit Unterstützung für die Nummertranskription: Meet me at 8:30 a.m. on June 1st at 100 Main Street with \$3.50 and 1.5 chocolate bars
- Alle anderen Sprachen: 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

Eine Liste der Sprachen, die das Transkribieren von Zahlen unterstützen, finden Sie unter.

[Unterstützte Sprachen und sprachspezifische Funktionen](#)

Erste Schritte mit Amazon Transcribe

Bevor Sie Transkriptionen erstellen können, müssen Sie einige Voraussetzungen erfüllen:

- [Registrieren Sie sich für ein AWS-Konto](#)
- [Installieren Sie die SDKs AWS CLI und](#) (wenn Sie die AWS-Managementkonsole für Ihre Transkriptionen verwenden, können Sie diesen Schritt überspringen)
- [Konfigurieren von IAM -Anmeldeinformationen](#)
- [Einen Amazon S3 -Bucket einrichten](#)
- [Eine IAM -Richtlinie erstellen](#)

Sobald Sie diese Voraussetzungen erfüllt haben, sind Sie bereit für die Transkription. Wählen Sie Ihre bevorzugte Transkriptionsmethode aus der folgenden Liste aus, um zu beginnen.

- [AWS CLI](#)
- [AWS-Managementkonsole](#)
- [AWS SDK](#)
- [HTTP](#)
- [WebSockets](#)

Tip

Wenn Sie mit unseren Funktionen noch nicht vertraut sind Amazon Transcribe oder diese erkunden möchten, empfehlen wir die Verwendung von [AWS-Managementkonsole](#). Dies ist auch die einfachste Option, wenn Sie einen Stream über das Mikrofon Ihres Computers starten möchten.

Da das Streamen mit HTTP/2 und komplizierter WebSockets ist als die anderen Transkriptionsmethoden, empfehlen wir, den [Einrichten einer Streaming-Transkription](#) Abschnitt zu lesen, bevor Sie mit diesen Methoden beginnen. Beachten Sie, dass wir dringend empfehlen, ein SDK für Streaming-Transkriptionen zu verwenden.

Melden Sie sich an für ein AWS-Konto

Sie können sich für ein [kostenloses](#) Konto oder ein [kostenpflichtiges Konto](#) registrieren. Mit beiden Optionen haben Sie Zugriff auf alle AWS-Services. Das kostenlose Kontingent umfasst eine Testphase, in der Sie Ihre Nutzung erkunden AWS-Services und einschätzen können. Nach Ablauf des Testzeitraums können Sie auf ein kostenpflichtiges Konto umsteigen. Die Gebühren werden auf einer Pay-as-you-use-Basis erhoben; Einzelheiten finden Sie unter [Amazon Transcribe -Preise](#).

Tip

Notieren Sie sich bei der Einrichtung Ihres Kontos Ihre AWS-Konto ID, da Sie sie zum Erstellen von IAM Entitäten benötigen.

Installieren von AWS CLI und SDKs

Um die Amazon Transcribe API verwenden zu können, müssen Sie zuerst die AWS CLI installieren. Die aktuelle AWS CLI ist Version 2. Installationsanweisungen für [Linux](#), [Mac](#), [Windows](#) und [Docker](#) finden Sie im [AWS Command Line Interface -Benutzerhandbuch](#).

Sobald Sie das AWS CLI installiert haben, müssen Sie es für Ihre Sicherheitsanmeldedaten [konfigurieren](#) und AWS-Region.

Wenn Sie es Amazon Transcribe mit einem SDK verwenden möchten, wählen Sie Ihre bevorzugte Sprache für die Installationsanweisungen aus:

- [.NET](#)
- [C++](#)
- [Go](#)
- [Java V2](#)
- [JavaScript](#)
- [PHP V3](#)
- [AWS SDK für Python \(Boto3\)](#) (Batch-Transkriptionen)
- [Python](#) (Streaming-Transkriptionen)
- [Rubin V3](#)
- [Rust](#) (Batch-Transkriptionen)

- [Rust](#) (Streaming-Transkriptionen)

Konfiguration IAM Anmeldedaten

Wenn Sie ein neues AWS-Konto erstellen, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle AWS Dienste und Ressourcen in Ihrem Konto hat. Diese Identität wird als AWS-Konto Root-Benutzer bezeichnet. Sie können darauf zugreifen, indem Sie sich mit der E-Mail-Adresse und dem Passwort anmelden, mit denen Sie das Konto erstellt haben.

Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden, auch nicht für administrative Aufgaben. Schützen Sie Ihre Root-Benutzer-Anmeldeinformationen und verwenden Sie diese, um die Aufgaben auszuführen, die nur der Root-Benutzer ausführen kann.

Es hat sich bewährt, dass Benutzer — auch solche, die Administratorzugriff benötigen — den Verbund mit einem Identitätsanbieter verwenden müssen, um mithilfe temporärer Anmeldeinformationen auf AWS Dienste zuzugreifen.

Eine föderierte Identität ist jeder Benutzer, der mithilfe von Anmeldeinformationen, die über eine Identitätsquelle bereitgestellt wurden, auf AWS Dienste zugreift. Wenn Verbundidentitäten auf AWS-Konten zugreifen, übernehmen sie Rollen und die Rollen stellen temporäre Anmeldeinformationen bereit.

Für die zentrale Zugriffsverwaltung empfehlen wir Ihnen, [AWS IAM Identity Center](#) zu verwenden. Sie können Benutzer und Gruppen in IAM Identity Center erstellen. Oder Sie können eine Verbindung zu einer Gruppe von Benutzern und Gruppen in Ihrer eigenen Identitätsquelle herstellen und diese synchronisieren, um sie in all Ihren Anwendungen zu AWS-Konten verwenden. Weitere Informationen finden Sie unter [Identity and Access Management für Amazon Transcribe](#).

Weitere Informationen zu IAM bewährten Methoden finden Sie unter [Bewährte Sicherheitsmethoden unter IAM](#).

Erstellen eines Amazon S3 bucket

Amazon S3 ist ein sicherer Objektspeicherdienst. Amazon S3 speichert Ihre Dateien (Objekte genannt) in Containern (sogenannten Buckets).

Um eine Batch-Transkription auszuführen, müssen Sie zuerst Ihre Mediendateien in einen Amazon S3 Bucket hochladen. Wenn Sie keinen Amazon S3 Bucket für Ihre Transkriptionsausgabe angeben,

wird Ihr Transkript in einem temporären AWS, Amazon Transcribe verwalteten Bucket gespeichert. Amazon S3 Die Transkriptionsausgabe in AWS-verwalteten Buckets wird nach 90 Tagen automatisch gelöscht.

Erfahren Sie, wie Sie [Ihren ersten S3-Bucket erstellen](#) und [Ein Objekt in Ihren Bucket hochladen](#).

Erstellen eines IAM policy

Um den Zugriff zu verwalten AWS, müssen Sie Richtlinien erstellen und diese an IAM Identitäten (Benutzer, Gruppen oder Rollen) oder AWS Ressourcen anhängen. Eine Richtlinie definiert die Berechtigungen der Entität, der sie zugeordnet ist. Beispielsweise kann eine Rolle nur dann auf eine Mediendatei zugreifen, die sich in Ihrem Amazon S3 Bucket befindet, wenn Sie dieser Rolle eine Richtlinie angehängt haben, die ihr Zugriff gewährt. Wenn Sie diese Rolle weiter einschränken möchten, können Sie stattdessen ihren Zugriff auf eine bestimmte Datei innerhalb eines Amazon S3 Buckets einschränken.

Weitere Informationen zur Verwendung von AWS Richtlinien finden Sie unter:

- [Richtlinien und Berechtigungen in IAM](#)
- [Erstellen von IAM -Richtlinien](#)
- [Wie Amazon Transcribe funktioniert mit IAM](#)

Richtlinien, die Sie zusammen verwenden können Amazon Transcribe, finden Sie beispielsweise unter [Amazon Transcribe Beispiele für identitätsbasierte Politik](#). Wenn Sie benutzerdefinierte Richtlinien erstellen möchten, sollten Sie den [AWS -Richtlinien-Generator](#) verwenden.

Sie können eine Richtlinie mithilfe des AWS SDK AWS-Managementkonsole AWS CLI, oder hinzufügen. Anweisungen finden Sie unter [Hinzufügen und Entfernen von IAM Identitätsberechtigungen](#).

Richtlinien haben das Format:

JSON

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

```

    {
      "Sid": "my-policy-name",
      "Effect": "Allow",
      "Action": [
        "service:action"
      ],
      "Resource": [
        "amazon-resource-name"
      ]
    }
  ]
}

```

Amazon Resource Names (ARNs) identifizieren alle AWS Ressourcen eindeutig, z. B. einen Amazon S3 Bucket. Sie können ARNs in Ihrer Richtlinie verwenden, um bestimmten Aktionen die Berechtigung zur Nutzung bestimmter Ressourcen zu erteilen. Wenn Sie beispielsweise Lesezugriff auf einen Amazon S3 Bucket und seine Unterordner gewähren möchten, können Sie dem Abschnitt Ihrer Vertrauensrichtlinie den folgenden Code hinzufügen: Statement

```

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

```

Im Folgenden finden Sie eine Beispielrichtlinie, die einem Amazon S3 Bucket und seinen Unterordnern Amazon Transcribe Lese- (PutObject) und Schreibrechte () gewährt: GetObject ListBucket amzn-s3-demo-bucket

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/*"
      ]
    }
  ]
}

```

Transkribieren mit dem AWS-Managementkonsole

Sie können die AWS Konsole für Batch- und Streaming-Transkriptionen verwenden. Wenn Sie eine Mediendatei transkribieren, die sich in einem Amazon S3 Bucket befindet, führen Sie eine Batch-Transkription durch. Wenn Sie einen Echtzeitstream von Audiodaten transkribieren, führen Sie eine Streaming-Transkription durch.

Bevor Sie eine Batch-Transkription starten, müssen Sie zunächst Ihre Mediendatei in einen Bucket hochladen. Amazon S3 Für Streaming-Transkriptionen mit dem AWS-Managementkonsole müssen Sie Ihr Computermikrofon verwenden.

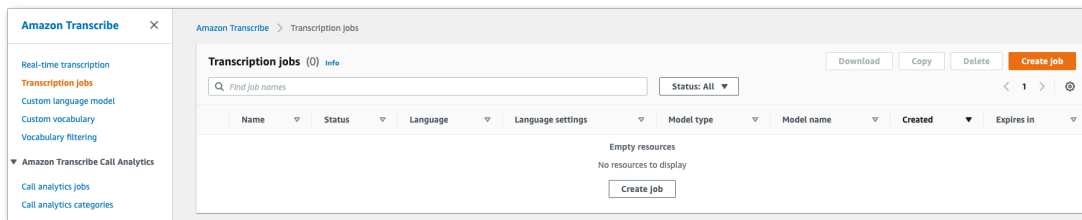
Informationen zu unterstützten Medienformaten und anderen Medienanforderungen und -beschränkungen finden Sie unter [Dateneingabe und -ausgabe](#).

In den folgenden Abschnitten finden Sie kurze Erläuterungen zu den einzelnen Transkriptionsmethoden.

Batch-Transkriptionen

Stellen Sie zunächst sicher, dass Sie die Mediendatei, die Sie transkribieren möchten, in einen Amazon S3 -Bucket hochgeladen haben. Wenn Sie sich nicht sicher sind, wie das geht, lesen Sie das Amazon S3 -Benutzerhandbuch: [Hochladen eines Objekts in Ihren Bucket](#).

1. Wählen Sie im [AWS-Managementkonsole](#) im linken Navigationsbereich Transkriptionsaufträge. Dies führt Sie zu einer Liste Ihrer Transkriptionsaufträge.



Wählen Sie Auftrag erstellen.

2. Füllen Sie die Felder auf der Seite Auftragsdetails aus.

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**

Der Eingabeort muss ein Objekt in einem Amazon S3 Bucket sein. Als Ausgabespeicherort können Sie einen sicheren, vom Amazon S3 Service verwalteten Bucket wählen oder Sie können Ihren eigenen Amazon S3 Bucket angeben.

Wenn Sie sich für einen serviceverwalteten Bucket entscheiden, können Sie eine Transkriptvorschau in der AWS-Managementkonsole anzeigen und Ihre Transkription von der Auftragsdetailseite herunterladen (siehe unten).

Wenn Sie Ihren eigenen Amazon S3 Bucket wählen, wird im Bucket keine Vorschau angezeigt AWS-Managementkonsole und Sie müssen zum Amazon S3 Bucket gehen, um Ihr Transkript herunterzuladen.

Input data [Info](#)

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

Browse 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.

Add new tag

You can add up to 50 more tags.

Cancel **Next**

Klicken Sie auf Weiter.

3. Wählen Sie auf der Seite Auftrag konfigurieren die gewünschten Optionen aus. Wenn Sie [Benutzerdefinierte Vokabulare](#) oder [Benutzerdefinierte Sprachmodelle](#) für Ihre Transkription verwenden möchten, müssen Sie diese vor Beginn Ihres Transkriptionsauftrags erstellen.

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](#)

Wählen Sie Auftrag erstellen.

- Sie befinden sich jetzt auf der Seite für Transkriptionsaufträge. Hier können Sie den Status des Transkriptionsauftrags sehen. Wählen Sie anschließend Ihre Transkription aus.

Transcription jobs (5) Info

Find job names

Status: All

Download

Copy

Delete

Create Job

<

1

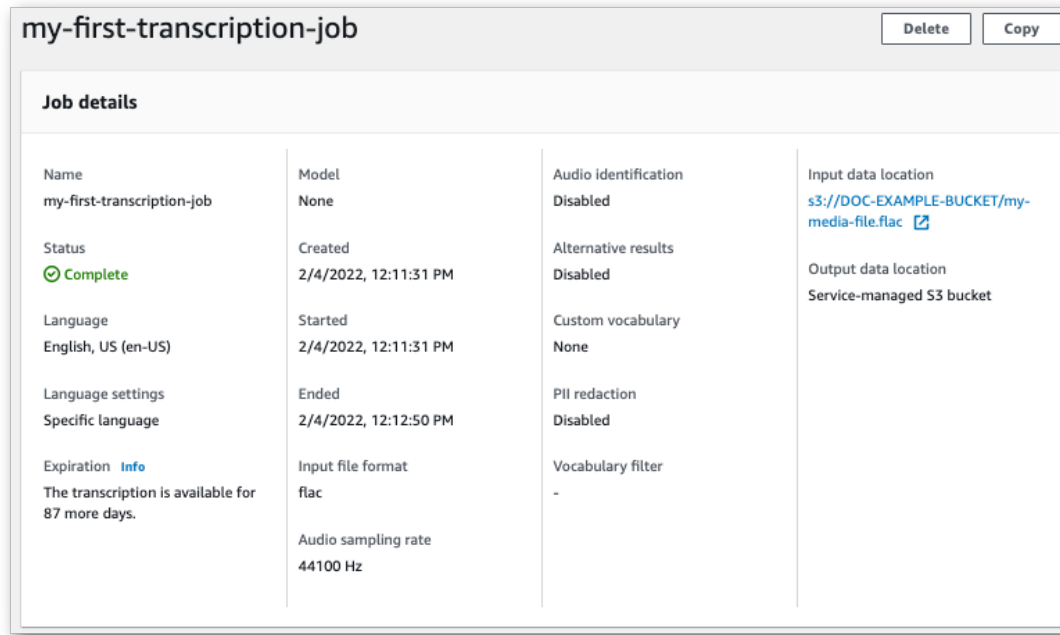
>

⌕

	Name	Status	Language	Language settings	Model type	Model name	Created	Expires in
☐	my-first-transcription	Complete	English, US (en-US)	Specific language	General	-	October 18 2021, 16:48 (UTC-07:00)	89 days

- Sie befinden sich jetzt auf der Seite mit den Auftragsdetails für Ihre Transkription. Hier können Sie alle Optionen einsehen, die Sie bei der Einrichtung Ihres Transkriptionsauftrags angegeben haben.

Um Ihr Transkript anzuzeigen, wählen Sie den verlinkten Dateipfad in der rechten Spalte unter Speicherort der Ausgabedaten. Dadurch gelangen Sie zu dem von Ihnen angegebenen Amazon S3 Ausgabeordner. Wählen Sie Ihre Ausgabedatei aus, die jetzt die Erweiterung .json hat.



6. Wie Sie Ihr Transkript herunterladen, hängt davon ab, ob Sie einen vom Service verwalteten Amazon S3 Bucket oder Ihren eigenen Bucket ausgewählt haben. Amazon S3
 - a. Wenn Sie sich für einen serviceverwalteten Bucket entschieden haben, sehen Sie auf der Informationsseite Ihres Transkriptionsauftrags ein Transkriptionsvorschaufenster und eine Schaltfläche zum Herunterladen.

my-first-transcription-job

DeleteCopy

Job details

Name my-first-transcription-job	Model None	Audio identification Disabled	Input data location s3://DOC-EXAMPLE-BUCKET/my-media-file.flac
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 Info 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.

Download

TextAudio identificationSubtitles

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.

Wählen Sie Herunterladen und wählen Sie Transkript herunterladen.

- b. Wenn Sie Ihren eigenen Amazon S3 Bucket ausgewählt haben, wird im Bereich der Transkriptionsvorschau auf der Informationsseite Ihres Transkriptionsauftrags kein Text angezeigt. Stattdessen sehen Sie ein blaues Informationsfeld mit einem Link zu dem von Ihnen ausgewählten Amazon S3 Bucket.

my-first-transcription-job

DeleteCopy

Job details

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

Transcription preview

Select download to save a local copy of the transcription.

Download ▼

Text

Audio identification

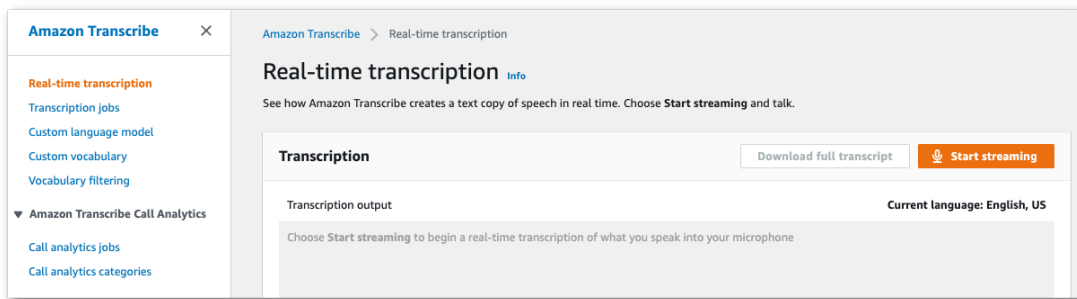
Subtitles

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

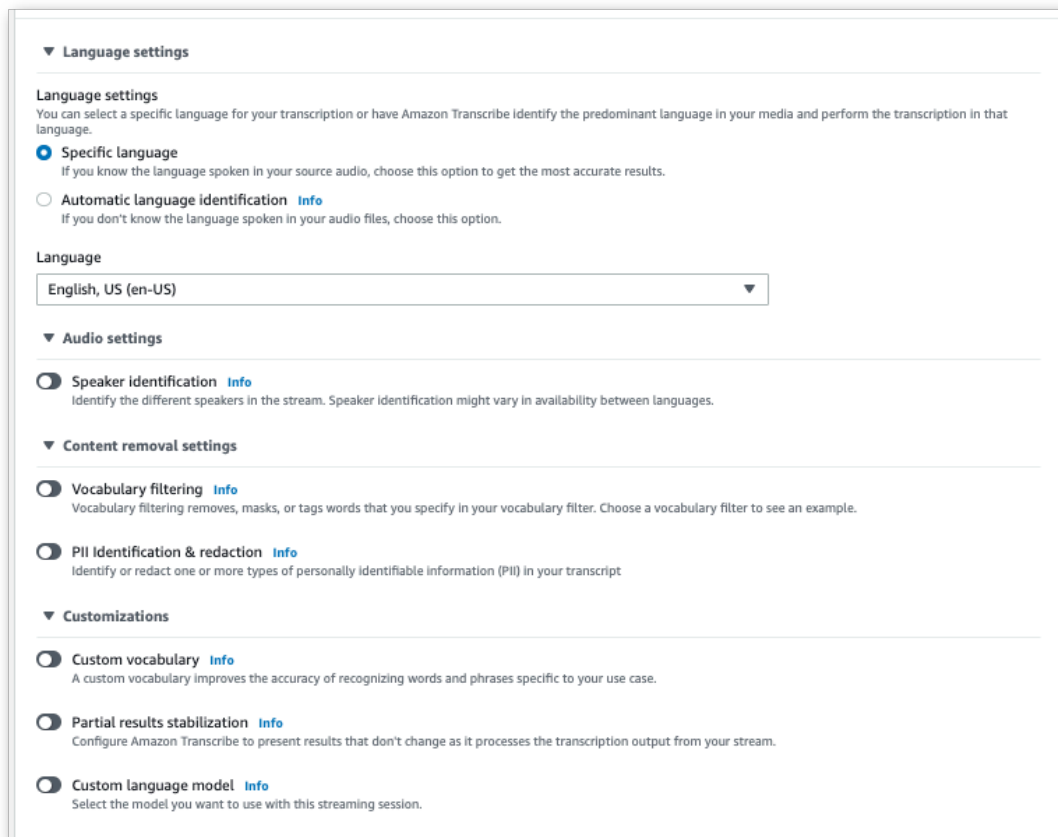
Um auf Ihr Transkript zuzugreifen, gehen Sie über den Link unter Speicherort der Ausgabedaten im Bereich Jobdetails oder über den Link S3-Bucket im blauen Informationsfeld im Bereich Transkriptionsvorschau zum angegebenen Amazon S3 Bucket.

Streaming-Transkriptionen

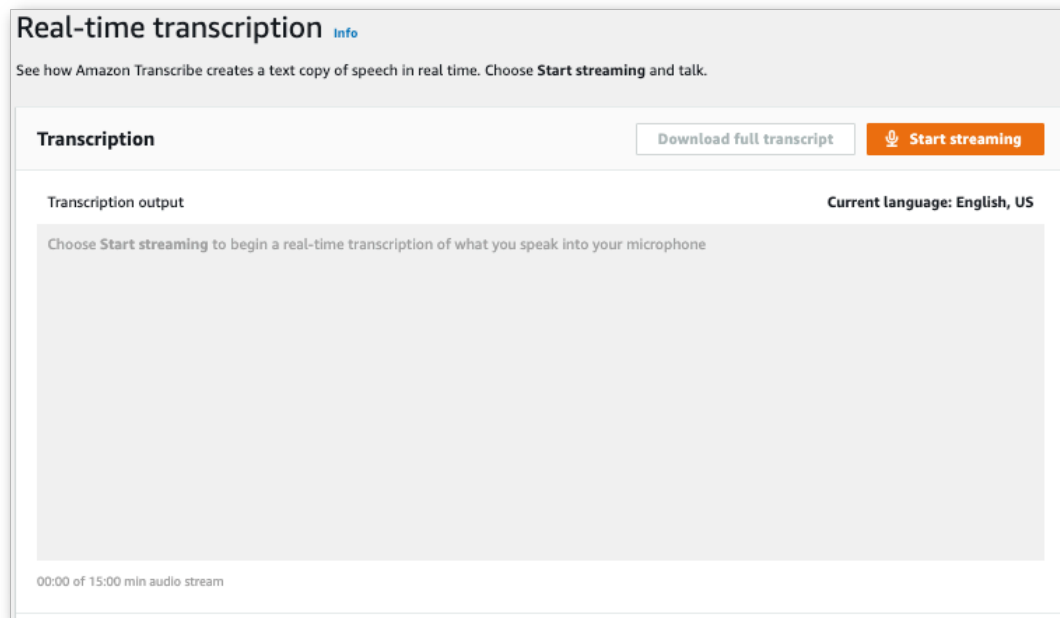
- Wählen Sie im [AWS-Managementkonsole](#) linken Navigationsbereich die Option Real-time Transkription aus. Dadurch gelangen Sie auf die Hauptseite für das Streaming, auf der Sie Optionen auswählen können, bevor Sie Ihren Stream starten.



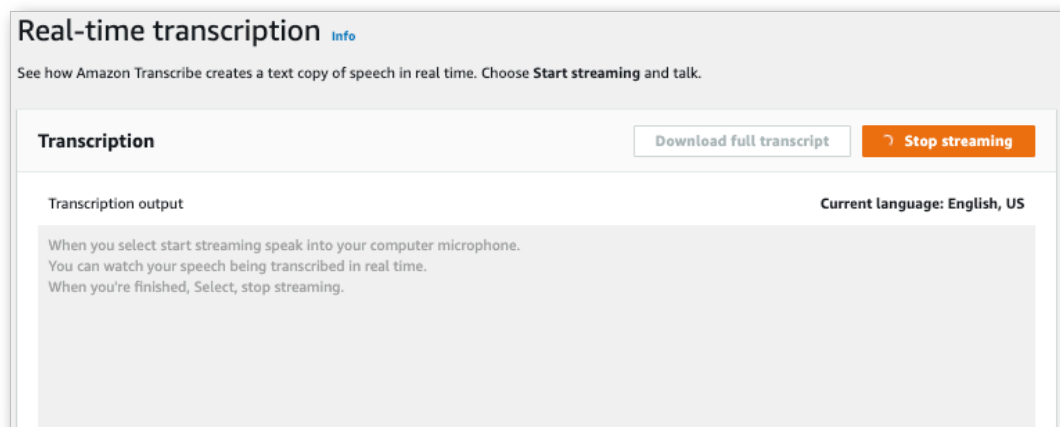
2. Unterhalb des Feldes Transkriptionsausgabe haben Sie die Möglichkeit, verschiedene Sprach- und Audioeinstellungen zu wählen.



3. Nachdem Sie die entsprechenden Einstellungen ausgewählt haben, blättern Sie zum oberen Rand der Seite, wählen Sie Streaming starten und beginnen Sie, in das Mikrofon Ihres Computers zu sprechen. Sie können sehen, wie Ihre Rede in Echtzeit transkribiert wird.



4. Wenn Sie fertig sind, wählen Sie Streaming beenden.



Sie können nun Ihr Zeugnis herunterladen, indem Sie Vollständige Transkription herunterladenwählen.

Transkribieren mit dem AWS CLI

Wenn Sie den verwenden AWS CLI , um eine Transkription zu starten, können Sie alle Befehle auf CLI-Ebene ausführen. Oder Sie führen den gewünschten Befehl aus, gefolgt von AWS-Region und dem Speicherort einer JSON-Datei, die einen Anforderungstext enthält. In diesem Leitfaden werden beide Methoden anhand von Beispielen dargestellt; dieser Abschnitt konzentriert sich jedoch auf die erste Methode.

Das unterstützt AWS CLI keine Streaming-Transkriptionen.

Bevor Sie fortfahren, vergewissern Sie sich, dass Sie Folgendes getan haben:

- Sie haben Ihre Mediendatei in einen Amazon S3 Bucket hochgeladen. Wenn Sie sich nicht sicher sind, wie Sie einen Amazon S3 Bucket erstellen oder Ihre Datei hochladen, finden Sie weitere Informationen unter [Erstellen Sie Ihren ersten Amazon S3 Bucket](#) und [Laden Sie ein Objekt in Ihren Bucket](#) hoch.
- Installiert die [AWS CLI](#).

Sie finden alle AWS CLI Befehle für Amazon Transcribe in der [AWS CLI Befehlsreferenz](#).

Aufnahme einer neuen Tätigkeit im Bereich der Transkription

Um eine neue Transkription zu starten, verwenden Sie den Befehl `start-transcription-job`.

1. Geben Sie in einem Terminalfenster Folgendes ein:

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

In der nächsten Zeile erscheint ein „>“, und Sie können nun, wie im nächsten Schritt beschrieben, die erforderlichen Parameter hinzufügen.

Sie können „\“ auch weglassen und alle Parameter mit einem Leerzeichen getrennt anhängen.

2. Mit dem Befehl `start-transcription-job` müssen Sie `region`, `transcription-job-name`, `media` und entweder `language-code` oder `identify-language` einschließen.

Wenn Sie einen Ausgabespeicherort angeben möchten, fügen Sie `output-bucket-name` in Ihre Anfrage ein; wenn Sie einen Unterordner des angegebenen Ausgabebereichs angeben möchten, fügen Sie auch `output-key` ein.

```
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
```

Wenn alle Parameter angehängt werden, sieht diese Anfrage wie folgt aus:

```
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
```

Wenn Sie mit `output-bucket-name` keinen Ausgabebereich angeben, legt Amazon Transcribe Ihre Transkriptionsausgabe in einem vom Dienst verwalteten Bereich ab. Die in einem serviceverwalteten Bucket gespeicherten Transkripte laufen nach 90 Tagen ab.

Amazon Transcribe antwortet mit:

```
{
  "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"
  }
}
```

Ihr Transkriptionsauftrag ist erfolgreich, wenn [TranscriptionJobStatus](#) von IN_PROGRESS auf COMPLETED wechselt. Um die aktualisierte [TranscriptionJobStatus](#) zu sehen, verwenden Sie den Befehl `get-transcription-job` oder `list-transcription-job`, wie im folgenden Abschnitt gezeigt.

Abrufen des Status eines Transkriptionsauftrags

Um Informationen über Ihren Transkriptionsauftrag zu erhalten, verwenden Sie den Befehl `get-transcription-job`.

Die einzigen erforderlichen Parameter für diesen Befehl sind der AWS-Region Ort, an dem sich der Job befindet, und der Name des Jobs.

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

```
--transcription-job-name my-first-transcription-job
```

Amazon Transcribe antwortet mit:

```
{
  "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
    }
  }
}
```

Wenn Sie Ihren eigenen Amazon S3 Bucket für Ihre Transkriptionsausgabe ausgewählt haben, ist dieser Bucket mit `TranscriptFileUri` aufgeführt. Wenn Sie einen serviceverwalteten Bucket ausgewählt haben, wird ein temporärer URI bereitgestellt. Verwenden Sie diesen URI, um Ihr Transkript herunterzuladen.

Note

Temporäre URIs für vom Service verwaltete Amazon S3 Buckets sind nur 15 Minuten gültig. Wenn Sie bei der Verwendung der URI eine `AccessDenied` Fehlermeldung erhalten, führen Sie die Anforderung `get-transcription-job` erneut aus, um eine neue temporäre URI zu erhalten.

Auflistung Ihrer Transkriptionsaufträge

Verwenden Sie den Befehl, um alle Ihre Transkriptionsaufträge in einer bestimmten AWS-Region Liste aufzulisten. `list-transcription-jobs`

Der einzige erforderliche Parameter für diesen Befehl ist der, AWS-Region in dem sich Ihre Transkriptionsaufträge befinden.

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

Amazon Transcribe antwortet mit:

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

Löschen Ihres Transkriptionsauftrags

Zum Löschen Ihres Transkriptionsauftrags verwenden Sie den Befehl `delete-transcription-job`.

Die einzigen erforderlichen Parameter für diesen Befehl sind der AWS-Region Ort, an dem sich der Job befindet, und der Name des Jobs.

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

Um zu bestätigen, dass Ihre Löschanfrage erfolgreich war, können Sie den Befehl `list-transcription-jobs` ausführen. Ihr Auftrag sollte nicht mehr in der Liste erscheinen.

Transkribieren mit dem AWS SDKs

Sie können SDKs sowohl für Batch- als auch für Streaming-Transkriptionen verwenden. Wenn Sie eine Mediendatei transkribieren, die sich in einem Amazon S3 Bucket befindet, führen Sie eine Batch-Transkription durch. Wenn Sie einen Echtzeitstream von Audiodaten transkribieren, führen Sie eine Streaming-Transkription durch.

Eine Liste der Programmiersprachen, die Sie mit Amazon Transcribe verwenden können, finden Sie unter [Unterstützte Programmiersprachen](#). Beachten Sie, dass Streaming-Transkriptionen nicht von allen AWS SDKs unterstützt werden. Informationen zu unterstützten Medienformaten und anderen Medienanforderungen und -beschränkungen finden Sie unter [Dateneingabe und -ausgabe](#).

Weitere Informationen zu allen verfügbaren AWS SDKs und Builder-Tools finden Sie unter [Tools](#) to Build On. AWS

Tip

Weitere Beispiele für die Verwendung der AWS SDKs, einschließlich funktionspezifischer, szenarienspezifischer und serviceübergreifender Beispiele, finden Sie im Kapitel.

[Codebeispiele für Amazon Transcribe mit AWS SDKs](#)

Sie finden SDK-Codebeispiele auch in diesen Repositories: GitHub

- [AWS Codebeispiele](#)
- [Amazon Transcribe Beispiele](#)

Batch-Transkriptionen

Sie können Batch-Transkriptionen mit dem URI einer Mediendatei erstellen, die sich in einem Amazon S3 -Bucket befindet. Wenn Sie sich nicht sicher sind, wie Sie einen Amazon S3 Bucket erstellen oder Ihre Datei hochladen, finden Sie weitere Informationen unter [Erstellen Sie Ihren ersten S3-Bucket](#) und [Laden Sie ein Objekt in Ihren Bucket](#) hoch.

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()
```

Streaming-Transkriptionen

Sie können Streaming-Transkriptionen mit einer gestreamten Mediendatei oder einem Live-Medienstream erstellen.

Beachten Sie, dass der Standard AWS SDK für Python (Boto3) für Amazon Transcribe Streaming nicht unterstützt wird. Verwenden Sie dieses [asynchrone Python-SDK für, um eine Streaming-Transkription mit Python](#) zu starten. Amazon Transcribe

Java

Das folgende Beispiel ist ein Java-Programm, das gestreamte Audiodaten transkribiert.

Um dieses Beispiel auszuführen, beachten Sie Folgendes:

- Sie müssen das [AWS -SDK für Java 2.x](#) verwenden.
- Clients müssen Java 1.8 verwenden, um mit dem [AWS -SDK für Java 2.x](#) kompatibel zu sein.
- Die von Ihnen angegebene Samplerate muss mit der tatsächlichen Samplerate Ihres Audiostreams übereinstimmen.

Siehe auch: [Client für Amazon Transcribe Streaming erneut versuchen \(Java SDK\)](#). Dieser Code verwaltet die Verbindung zu Amazon Transcribe und versucht erneut, Daten zu senden, wenn bei der Verbindung Fehler auftreten. Beispiel: Wenn ein vorübergehender Fehler im Netzwerk vorliegt, sendet dieser Client die fehlgeschlagene Anforderung erneut.

```
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

Das folgende Beispiel ist ein Python-Programm, das Streaming-Audio transkribiert.

Um dieses Beispiel auszuführen, beachten Sie Folgendes:

- Sie müssen dieses [SDK für Python](#) verwenden.
- Die von Ihnen angegebene Samplerate muss mit der tatsächlichen Samplerate Ihres Audiostreams übereinstimmen.

```

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++

Ein [Streaming C++ SDK-Beispiel](#) finden Sie im Kapitel Codebeispiele.

Verwenden Sie diesen Dienst mit einem AWS SDK

AWS Software Development Kits (SDKs) sind für viele beliebte Programmiersprachen verfügbar. Jedes SDK bietet eine API, Codebeispiele und Dokumentation, die es Entwicklern erleichtern, Anwendungen in ihrer bevorzugten Sprache zu erstellen.

SDK-Dokumentation	Codebeispiele
AWS SDK für C++	AWS SDK für C++ Code-Beispiele
AWS CLI	AWS CLI Code-Beispiele
AWS SDK für Go	AWS SDK für Go Code-Beispiele
AWS SDK für Java	AWS SDK für Java Code-Beispiele
AWS SDK für JavaScript	AWS SDK für JavaScript Code-Beispiele
AWS SDK für Kotlin	AWS SDK für Kotlin Code-Beispiele
AWS SDK für .NET	AWS SDK für .NET Code-Beispiele
AWS SDK für PHP	AWS SDK für PHP Code-Beispiele
AWS -Tools für PowerShell	AWS -Tools für PowerShell Code-Beispiele
AWS SDK für Python (Boto3)	AWS SDK für Python (Boto3) Code-Beispiele
AWS SDK für Ruby	AWS SDK für Ruby Code-Beispiele
AWS SDK für Rust	AWS SDK für Rust Code-Beispiele
AWS SDK für SAP ABAP	AWS SDK für SAP ABAP Code-Beispiele
AWS SDK für Swift	AWS SDK für Swift Code-Beispiele

Weitere Beispiele speziell für diesen Service finden Sie unter [Codebeispiele für Amazon Transcribe mit AWS SDKs](#).

Beispiel für die Verfügbarkeit

Sie können nicht finden, was Sie brauchen? Fordern Sie ein Codebeispiel an, indem Sie unten den Link [Provide feedback \(Feedback geben\)](#) auswählen.

Transkribieren mit HTTP oder WebSockets

Amazon Transcribe unterstützt HTTP sowohl für Batch (HTTP/1.1) als auch für Streaming (HTTP/2) - Transkriptionen. WebSockets werden für Streaming-Transkriptionen unterstützt.

Wenn Sie eine Mediendatei transkribieren, die sich in einem Amazon S3 Bucket befindet, führen Sie eine Batch-Transkription durch. Wenn Sie einen Echtzeitstream von Audiodaten transkribieren, führen Sie eine Streaming-Transkription durch.

Sowohl HTTP als auch WebSockets erfordern, dass Sie Ihre Anfrage mithilfe von AWS Signature Version 4-Headern authentifizieren. Weitere Informationen finden Sie unter [AWS API-Anfragen signieren](#).

Batch-Transkriptionen

Sie können eine Batch-HTTP-Anfrage mit den folgenden Headern stellen:

- Host
- x-amz-ziel
- Inhalts-Typ
- x-amz-content-sha256
- x-amz-Datum
- Autorisierung

Hier ist ein Beispiel für eine `StartTranscriptionJob` Anfrage:

```
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/"
}
```

Zusätzliche Operationen und Parameter sind in der [API-Referenz](#) aufgeführt. Parameter, die allen AWS API-Vorgängen gemeinsam sind, sind im Abschnitt [Allgemeine Parameter](#) aufgeführt. Weitere Signaturelemente werden unter [Elemente einer AWS Signaturversion 4-Anfrage](#) detailliert beschrieben.

Streaming-Transkriptionen

Streaming-Transkriptionen, die SDKs verwenden HTTP/2, WebSockets sind aufwändiger als die Verwendung von SDKs. Wir empfehlen Ihnen, den Abschnitt [Einrichten einer Streaming-Transkription](#) durchzulesen, bevor Sie Ihren ersten Stream einrichten.

Weitere Informationen zu diesen Methoden finden Sie unter [Einen HTTP/2 Stream einrichten](#) oder [Einen WebSocket Stream einrichten](#).

Note

Wir empfehlen dringend die Verwendung eines SDK für Streaming-Transkriptionen. Eine Liste der unterstützten SDKs finden Sie unter [Unterstützte Programmiersprachen](#).

Transkribieren gestreamter Audiodaten

Mithilfe von Amazon Transcribe Streaming können Sie in Echtzeit Transkriptionen für Ihre Medieninhalte erstellen. Im Gegensatz zu Batch-Transkriptionen, bei denen Mediendateien hochgeladen werden, werden Streaming-Medien in Echtzeit übertragen. Amazon Transcribe Amazon Transcribe gibt dann ein Transkript zurück, ebenfalls in Echtzeit.

Streaming kann sowohl aufgezeichnete Medien (Filme, Musik und Podcasts) als auch Echtzeitmedien (Live-Nachrichtensendungen) umfassen. Zu den häufigsten Streaming-Anwendungsfällen Amazon Transcribe gehören Live-Untertitel für Sportveranstaltungen und die Echtzeitüberwachung von Callcenter-Audio.

Streaming-Inhalte werden in Form von aufeinanderfolgenden Datenpaketen oder „Chunks“ geliefert, die Amazon Transcribe sofort transkribiert. Zu den Vorteilen von Streaming gegenüber Batchverarbeitung gehören Echtzeit-Sprache-zu-Text-Funktionen in Ihren Anwendungen und schnellere Transkriptionszeiten. Diese höhere Geschwindigkeit kann jedoch in einigen Fällen zu Einschränkungen bei der Genauigkeit führen.

Amazon Transcribe bietet die folgenden Optionen für das Streaming:

- [SDKs](#) (bevorzugt)
- [HTTP/2](#)
- [WebSockets](#)
- [AWS-Managementkonsole](#)

Um Streaming-Audio im zu transkribieren AWS-Managementkonsole, sprechen Sie in das Mikrofon Ihres Computers.



Tip

SDK-Codebeispiele finden Sie im [AWS Samples-Repository](#) unter. GitHub

Folgende Audioformate werden für Streaming-Transkriptionen unterstützt:

- FLAC
- OPUS-encoded Audio in einem Ogg-Container

- PCM (nur signierte 16-Bit Little-Endian-Audioformate, die WAV nicht enthalten)

Verlustfreie Formate (FLAC oder PCM) werden empfohlen.

Note

Streaming-Transkriptionen werden nicht in allen Sprachen unterstützt. Einzelheiten finden Sie in der Spalte „Dateneingabe“ in der [Tabelle „Unterstützte Sprachen“](#).

Informationen zur Verfügbarkeit von Streaming-Transkriptionen in der Amazon Transcribe Region finden Sie unter: [Amazon Transcribe Endpunkte](#) und Kontingente.

Best Practices

Die folgenden Empfehlungen verbessern die Effizienz der Streaming-Transkription:

- Verwenden Sie nach Möglichkeit Audio. PCM-encoded
- Stellen Sie sicher, dass Ihr Stream so nah wie möglich an Echtzeit ist.
- Die Latenzzeit hängt von der Größe der Audio-Blöcke ab. Wenn Sie bei Ihrem Audiotyp (z. B. PCM) die Möglichkeit haben, die Chunk-Größe anzugeben, stellen Sie jeden Chunk auf einen Wert zwischen 50 ms und 200 ms ein. Sie können die Größe des Audio-Chunks mit der folgenden Formel berechnen:

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

- Verwenden Sie eine einheitliche Chunk-Größe.
- Stellen Sie sicher, dass Sie die Anzahl der Audiokanäle korrekt angeben.
- Bei einkanaligem PCM-Audio besteht jedes Sample aus zwei Bytes, sodass jeder Chunk aus einer geraden Anzahl von Bytes bestehen sollte.
- Bei Zweikanal-PCM-Audio besteht jedes Sample aus vier Bytes, sodass jeder Chunk ein Vielfaches von 4 Bytes sein sollte.
- Wenn Ihr Audiostream keine Sprache enthält, kodieren und senden Sie die gleiche Menge an Stille. So ist beispielsweise Stille bei PCM ein Stream von null Bytes.
- Vergewissern Sie sich, dass Sie die richtige Abtastrate für Ihr Audiomaterial angeben. Nehmen Sie nach Möglichkeit mit einer Abtastrate von 16.000 Hz auf; dies ist der beste Kompromiss zwischen

Qualität und Datenmenge, die über das Netzwerk übertragen wird. Beachten Sie, dass die meisten High-End-Mikrofone mit 44.100 Hz oder 48.000 Hz aufnehmen.

Umgang mit LimitExceededException Fehlern

Verfügt wie jedes verteilte System Amazon Transcribe über Schutzmechanismen, die einen übermäßigen Ressourcenverbrauch erkennen und entsprechend reagieren.

`LimitExceededException`-Fehler können auftreten, wenn einer dieser Mechanismen ausgelöst wird. Für diesen Fehler gibt es drei unterschiedliche Ursachen:

Das Kontingent für den gleichzeitigen Stream-Service wurde überschritten

Dies ist die häufigste Ursache. Sie tritt auf, wenn Sie Ihr Kontingent für den gleichzeitigen Stream-Service überschreiten. Um diesen Fehler zu beheben, versuchen Sie es erneut mit exponentiellem Backoff. Wenn Sie dieses Limit ständig erreichen, fordern Sie über die Service [Quota-Konsole eine Erhöhung der Service Quotas](#) an. Sie können sich auch an das [AWS Support Center](#) wenden, um Unterstützung zu erhalten. Weitere Informationen zu Wiederholungsstrategien finden Sie unter [Verhalten bei Wiederholungsversuchen im Referenzhandbuch](#) für AWS SDKs und Tools.

Die maximale Sitzungsdauer wurde überschritten

Dieser Fehler tritt auf, wenn ein Stream die maximal zulässige Sitzungsdauer überschreitet. Dies ist ein fester Grenzwert, der nicht erhöht werden kann. Um mit dem Transkribieren fortzufahren, starten Sie eine neue Streaming-Sitzung.

Die Anzahl der gleichzeitigen Streams ist zu schnell gestiegen

Dies ist eine seltene Ursache. Dies kann auftreten, wenn Sie die Anzahl der gleichzeitigen Streams zu schnell erhöhen, z. B. bei Auslastungstests. Dabei handelt es sich um einen Schutzmechanismus auf Systemebene ohne einstellbares Kontingent. Um diesen Fehler zu beheben, versuchen Sie es erneut mit exponentiellem Backoff und erhöhen Sie schrittweise die Anzahl gleichzeitiger Streams. Weitere Informationen zu Wiederholungsstrategien finden Sie unter [Wiederholungsverhalten](#) im Referenzhandbuch für SDKs und Tools. Sie können auch [AWS re:POST](#) besuchen oder den [AWS Premium-Support](#) kontaktieren.

Streaming und Teilergebnisse

Da das Streaming in Echtzeit funktioniert, werden Transkripte in Teilergebnissen erstellt. Amazon Transcribe unterbricht den eingehenden Audiostream auf der Grundlage natürlicher Sprachsegmente, z. B. eines Sprecherwechsels oder einer Audiopause. Die Transkription wird in einem Stream von Transkriptionsereignissen an Ihre Anwendung zurückgegeben, wobei jede Antwort mehr transkribierte Sprache enthält, bis ein ganzes Segment transkribiert ist.

Eine Annäherung an diesen Vorgang zeigt der folgende Codeblock. Sie können diesen Vorgang in Aktion verfolgen, indem Sie sich bei anmelden [AWS-Managementkonsole](#), Real-time Transkription auswählen und in Ihr Mikrofon sprechen. Beobachten Sie das Transkriptionsfenster, während Sie sprechen.

In diesem Beispiel ist jede Zeile das Teilergebnis eines Audiosegments.

```
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.
```

Diese Teilergebnisse sind in Ihrer Transkriptionsausgabe innerhalb der [Results](#)-Objekte. In diesem Objektblock befindet sich auch ein `IsPartial`-Feld. Wenn dieses Feld wahr ist, ist Ihr Transkriptionssegment noch nicht vollständig. Den Unterschied zwischen einem unvollständigen und einem vollständigen Segment können Sie unten sehen:

```
"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
```

Jedem Wort innerhalb eines vollständigen Segments ist ein Konfidenzwert zugeordnet, der zwischen 0 und 1 liegt. Ein größerer Wert bedeutet, dass das Wort mit größerer Wahrscheinlichkeit richtig transkribiert wird.

Tip

Die `StartTime` und `EndTime` eines Audiosegments können verwendet werden, um die Transkriptionsausgabe mit dem Videodialog zu synchronisieren.

Wenn Sie eine Anwendung ausführen, die eine geringe Latenzzeit erfordert, sollten Sie die [Stabilisierung von Teilergebnissen](#) verwenden.

Partial-result Stabilisierung

Amazon Transcribe beginnt mit der Rückgabe von Transkriptionsergebnissen, sobald Sie mit dem Streamen Ihres Audios beginnen. Es gibt diese Teilergebnisse schrittweise zurück, bis es ein fertiges Ergebnis auf der Ebene eines natürlichen Sprachsegments erzeugt. Ein natürliches Sprachsegment ist eine kontinuierliche Rede, die eine Pause oder einen Sprecherwechsel enthält.

Amazon Transcribe setzt die Ausgabe von Teilergebnissen fort, bis das endgültige Transkriptionsergebnis für ein Sprachsegment generiert wird. Da die Spracherkennung Wörter überarbeiten kann, wenn sie mehr Kontext erhält, können sich die Streaming-Transkriptionen mit jeder neuen Teilergebnisausgabe leicht verändern.

Bei diesem Verfahren stehen Ihnen für jedes Sprachsegment zwei Optionen zur Verfügung:

- Warten auf das fertige Segment
- Verwenden Sie die Teilergebnisse des Segments

Die Stabilisierung der partiellen Ergebnisse verändert die Art und Weise, wie das endgültige Transkriptionsergebnis für jedes vollständige Segment Amazon Transcribe erzeugt wird. Wenn diese Funktion aktiviert ist, können nur die letzten Wörter der Teilergebnisse geändert werden. Aus diesem Grund kann die Genauigkeit der Transkription beeinträchtigt werden. Ihr Transkript wird jedoch schneller zurückgegeben als ohne Stabilisierung von Teilergebnissen. Diese Verringerung der Latenzzeit kann bei der Untertitelung von Videos oder der Erstellung von Untertiteln für Live-Streams von Vorteil sein.

Die folgenden Beispiele zeigen, wie derselbe Audiostrom behandelt wird, wenn die Stabilisierung von Teilergebnissen nicht aktiviert ist und wenn sie aktiviert ist. Beachten Sie, dass Sie die Stabilitätsstufe auf niedrig, mittel oder hoch einstellen können. Geringe Stabilität bietet höchste Genauigkeit. Hohe Stabilität bedeutet eine schnellere Transkription, allerdings mit etwas geringerer Genauigkeit.

„Transkript“:	"EndTime":	"IsPartial":
Partial-result Stabilisierung nicht aktiviert		
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
The Amazon is the		
largest rainforest on		
the		
The Amazon is the		
largest rainforest on		
the planet.		
The Amazon is the		
largest rainforest on		
the planet.		

„Transkript“:	"EndTime":	"IsPartial":
The Amazon is the largest rainforest on the planet.		

Partial-result Stabilisierung aktiviert (hohe Stabilität)

The	0.515	true
The	1.015	true
The Amazon.	1.515	true
The Amazon is	2.015	true
The Amazon is the large	2.515	true
The Amazon is the	3.015	true
largest	3.515	true
The Amazon is the	4.015	true
largest rainfall.	4.515	true
The Amazon is the	5.015	true
largest rain forest.	5.515	true
The Amazon is the	6.015	true
largest rain forest on	6.335	true
The Amazon is the	6.335	false
largest rain forest on		
the planet.		
The Amazon is the		
largest rain forest on		
the planet.		
The Amazon is the		
largest rain forest on		
the planet.		
The Amazon is the		
largest rain forest on		
the planet.		

Wenn Sie die Stabilisierung von Teilergebnissen aktivieren, Amazon Transcribe verwendet ein Stable Feld, um anzugeben, ob ein Element stabil ist, wobei sich „Element“ auf ein transkribiertes Wort oder ein Interpunktionszeichen bezieht. Die Werte für Stable sind true oder false.

Elemente, die als `false` (nicht stabil) gekennzeichnet sind, werden sich während der Transkription Ihres Segments mit größerer Wahrscheinlichkeit ändern. Umgekehrt ändern sich die als `true` (stabil) gekennzeichneten Elemente nicht.

Sie können festlegen, dass nicht stabile Wörter wiedergegeben werden, damit Ihre Beschriftungen mit der Sprache übereinstimmen. Selbst wenn sich die Beschriftungen mit dem hinzugefügten Kontext leicht verändern, ist dies eine bessere Benutzererfahrung als periodische Textbursts, die mit der Sprache übereinstimmen können oder auch nicht.

Sie können auch nicht stabile Wörter in einem anderen Format anzeigen, z. B. kursiv, um den Betrachter darauf hinzuweisen, dass sich diese Wörter ändern können. Die Anzeige von Teilergebnissen schränkt die Menge des angezeigten Textes ein. Dies kann wichtig sein, wenn der Platz begrenzt ist, wie bei Videountertiteln.

 Tauchen Sie tiefer ein mit dem AWS Blog zum Machine Learning

Weitere Informationen zur Verbesserung der Genauigkeit von Echtzeit-Transkriptionen finden Sie hier:

- [Verbessern Sie das Streaming-Transkriptionserlebnis durch die Stabilisierung Amazon Transcribe teilweiser Ergebnisse](#)
- [„Was war das?“ Erhöhung der Untertitelgenauigkeit bei Live-Übertragungen mit Amazon Transcribe](#)

Partial-result Ausgabe eines Beispiels für eine Stabilisierung

Die folgende Beispielausgabe zeigt `Stable`-Markierungen für ein unvollständiges Segment (`"IsPartial": true`). Sie können sehen, dass die Wörter „to“ und „Amazon“ nicht stabil sind und sich daher ändern können, bevor das Segment abgeschlossen ist.

```
"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
}
]
}

```

Einrichten einer Streaming-Transkription

Dieser Abschnitt erweitert den Hauptabschnitt über [Streaming](#). Es soll Informationen für Benutzer bereitstellen, die ihren Stream mit HTTP/2 oder WebSockets direkt und nicht mit einem AWS SDK einrichten möchten. Die Informationen in diesem Abschnitt können auch zur Erstellung eines eigenen SDKs verwendet werden.

Important

Wir empfehlen dringend, SDKs zu verwenden, anstatt HTTP/2 und WebSockets direkt zu verwenden. SDKs sind die einfachste und zuverlässigste Methode zur Transkription von Datenströmen. Informationen zum Starten des Streamings mithilfe eines AWS SDK finden Sie unter [Transkribieren mit dem AWS SDKs](#).

Einen HTTP/2 Stream einrichten

Die wichtigsten Komponenten für ein [HTTP/2 Protokoll](#) zum Streamen von Transkriptionsanfragen mit Amazon Transcribe sind:

- Ein Header-Frame. Es enthält die HTTP/2 Header für Ihre Anfrage und eine Signatur im Autorisierungsheader, die als Startsignatur zum Signieren der Datenrahmen Amazon Transcribe verwendet wird.
- Ein oder mehrere Nachrichtenframe(s) in [Ereignisstromkodierung](#), der/die Metadaten und Roh-Audio-Bytes enthält/enthalten.
- Ein Endframe. Hierbei handelt es sich um eine signierte Nachricht in der [Ereignis-Stream-Kodierung](#) mit einem leeren Text.

Note

Amazon Transcribe unterstützt nur einen Stream pro HTTP/2 Sitzung. Wenn Sie versuchen, mehrere Streams zu verwenden, schlägt Ihre Transkriptionsanfrage fehl.

1. Fügen Sie der IAM Rolle, die die Anfrage stellt, die folgende Richtlinie hinzu. Weitere Informationen finden Sie unter [IAM Richtlinien hinzufügen](#).

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "myTranscribeHttp2Policy",
      "Effect": "Allow",
      "Action": "transcribe:StartStreamTranscription",
      "Resource": "*"
    }
  ]
}
```

2. Um die Sitzung zu starten, senden Sie eine HTTP/2 Anfrage an 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
```

Zusätzliche Operationen und Parameter sind in der [API-Referenz](#) aufgeführt. Parameter, die allen AWS -API-Operationen gemeinsam sind, werden im Abschnitt [Allgemeine Parameter](#) aufgeführt.

Amazon Transcribe sendet die folgende Antwort:

```
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. Erstellen Sie ein Audio-Ereignis, das Ihre Audiodaten enthält. Kombinieren Sie die in der folgenden Tabelle beschriebenen Headern mit einem Chunk von Audio-Bytes in einer ereigniskodierten Nachricht. Nutzen Sie einen Puffer im Rohbyteformat, um die Nutzlast für die Ereignisnachricht zu erstellen.

Bytelänge des Header-Namens	Header-Name (Zeichenfolge)	Werttyp des Header	Bytelänge der Wertzeichenfolge	Wert string (UTF-8)
13	:content-type	7	24	application/octet-stream
11	:event-type	7	10	AudioEvent
13	:message-type	7	5	event

Die binären Daten in dieser Beispielanfrage sind base64-kodiert. Bei einer tatsächlichen Anfrage sind die Daten Rohbytes.

```
:content-type: "application/vnd.amazon.eventstream"
:event-type: "AudioEvent"
:message-type: "event"
UklGRjzxPQBxQVZFZm10IBAAAAABAAEAgD4AAAB9AAACABAAZGF0YVtwPQAAAAAAAAAAAAAAAAAAD//wIA/f8EAA==
```

4. Erstellen Sie eine Audionachricht, die Ihre Audiodaten enthält.
- Ihr Audionachrichten-Datenrahmen enthält Ereigniscodierungs-Header, die das aktuelle Datum und eine Signatur für den Audiochunk und das Audioereignis enthalten.

Bytelänge des Header-Namens	Header-Name (Zeichenfolge)	Werttyp des Header	Bytelänge der Wertzeichenfolge	Wert
16	:chunk-signature	6	variiert	generierte Signatur
5	:date	8	8	Zeitstempel

Die binären Daten in dieser Anfrage sind base64-kodiert. Bei einer tatsächlichen Anfrage sind die Daten Rohbytes.

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

```
:chunk-signature: signature
```

```
AAAA0gAAAIKVoRFcTTcjb250ZW50LXR5cGUHABhhcHBsaWNhdGlvbi9vY3RldC1zdHJlYW0L0mV2ZW50LXR5cGUHAAPBdWRpb0V2ZW50DTptZXNzYWdl1LXR5cGUHAAPVdmVudAxDb256ZW50LVR5cGUHABphcHBsaWNhdGlvbi94LWFtei1qc29uLTEuMVJJRkY88T0AV0FWRWZtdCAQAAAAAQAABAIA
+AAAAfQAAAgAQAGRhdGFU8D0AAAAA
AAAAAAAAAAAA//8CAP3/BAC7QLFf
```

- b. Konstruieren Sie eine zu signierende Zeichenfolge, wie unter [Erstellen einer zu signierenden Zeichenfolge für Signature Version 4](#) beschrieben. Ihre Zeichenkette hat dieses Format:

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

- *DateTime*: Datum und Uhrzeit der Erstellung der Signatur. Das Format ist JJJJMMTTTHHMMSSZ, wobei JJJJ=Jahr, MM=Monat, TT=Tag, HH=Stunde,

MM=Minute, SS= Sekunden und „T“ und „Z“ feste Zeichen sind. Weitere Informationen finden Sie unter [Handhabung von Daten in Signature Version 4](#).

- Keypath: Der Signaturbereich im Format date/region/service/aws4_request. Beispiel, 20220127/us-west-2/transcribe/aws4_request.
 - Hex: Eine Funktion, die Eingaben in eine hexadezimale Darstellung kodiert.
 - priorSignature: Die Signatur für den vorherigen Frame. Verwenden Sie die Signatur des Header-Frames für den ersten Datenframe.
 - HexHash: Eine Funktion, die zuerst einen SHA-256 Hash ihrer Eingabe erstellt und dann die Hex-Funktion verwendet, um den Hash zu kodieren.
 - non SignatureHeaders: Der DateTime Header ist als Zeichenfolge kodiert.
 - payload: Der Byte-Puffer, der die Audio-Ereignisdaten enthält.
- c. Leiten Sie aus Ihrem AWS geheimen Zugriffsschlüssel einen Signaturschlüssel ab und signieren Sie ihn damit. `stringToSign` Für ein höheres Maß an Schutz ist der abgeleitete Schlüssel spezifisch für das Datum, den Service und AWS-Region. Weitere Informationen finden Sie unter [Berechnen der Signatur für AWS Signature Version 4](#).

Stellen Sie sicher, dass Sie die Funktion `GetSignatureKey` implementieren, um Ihren Signaturschlüssel abzuleiten. Wenn Sie noch keinen Signaturschlüssel abgeleitet haben, lesen Sie den Abschnitt [Beispiele für die Ableitung eines Signaturschlüssels für die Signature Version 4](#).

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

- HMACSHA256: Eine Funktion, die mithilfe der Hash-Funktion eine Signatur erstellt. SHA-256
- abgeleitet SigningKey: Der Signaturschlüssel von Signature Version 4.
- string ToSign: Die Zeichenfolge, die Sie für den Datenrahmen berechnet haben.

Nachdem Sie die Signatur für den Data Frame berechnet haben, erstellen Sie einen Bytepuffer, der das Datum, die Signatur und die Nutzlast des Audioereignisses enthält. Senden Sie das Byte-Array zur Transkription an Amazon Transcribe .

5. Um anzuzeigen, dass der Audiostream vollständig ist, senden Sie einen End-Frame (einen leeren Datenrahmen), der nur das Datum und die Signatur enthält. Sie konstruieren diesen Endframe auf die gleiche Weise wie einen Data Frame.

Amazon Transcribe antwortet mit einem Stream von Transkriptionseignissen, die an Ihre Anwendung gesendet werden. Diese Antwort ist Ereignisstreamkodiert. Sie enthält die Standardeinleitung und die folgenden Header:

Bytelänge des Header-Namens	Header-Name (Zeichenfolge)	Werttyp des Header	Bytelänge der Wertzeichenfolge	Wertzeichenfolge (UTF-8)
13	:content-type	7	16	application/json
11	:event-type	7	15	TranscriptEvent
13	:message-type	7	5	event

Die Ereignisse werden im Rohbyteformat gesendet. In diesem Beispiel sind die Bytes base64-kodiert.

```
AAAAUwAAAEp1RHpYBTpkYXR1CAAAAWiXUkMLEDpjaHVuay1zaWduYXR1cmUGACCT6Zy+uymwEK2Srlp/
zVBI
5eGn83jdBwCaRUBJA+eaDafqjqI=
```

Um die Transkriptionsergebnisse zu sehen, dekodieren Sie die rohen Bytes mit der Ereignisstreamkodierung.

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

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

6. Um Ihren Stream zu beenden, senden Sie ein leeres Audioereignis an Amazon Transcribe. Erstellen Sie das Audioereignis genau wie jedes andere (Ausnahme: leere Nutzlast). Signieren Sie das Ereignis und fügen Sie die Signatur wie folgt in den Header `:chunk-signature` ein:

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

Umgang mit HTTP/2 Streaming-Fehlern

Wenn bei der Verarbeitung Ihres Medienstreams ein Fehler auftritt, Amazon Transcribe sendet eine Ausnahmeantwort. Die Antwort ist Ereignis-Stream-kodiert.

Die Antwort enthält die Standardeinleitung und die folgenden Header:

Bytelänge des Header-Namens	Header-Name (Zeichenfolge)	Werttyp des Header	Bytelänge der Wertzeichenfolge	Wertzeichenfolge (UTF-8)
13	:content-type	7	16	application/json
11	:event-type	7	19	BadRequestException
13	:message-type	7	9	Ausnahme

Wenn die Ausnahmeantwort kodiert ist, enthält sie die folgenden Informationen:

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

Exception message

Einen WebSocket Stream einrichten

Die wichtigsten Komponenten eines [WebSocketProtokolls](#) zum Streamen von Transkriptionsanfragen mit Amazon Transcribe sind:

- Die Upgrade-Anforderung. Es enthält die Abfrageparameter für Ihre Anfrage und eine Signatur, die als Startsignatur zum Signieren der Datenrahmen Amazon Transcribe verwendet wird.
- Ein oder mehrere Nachrichtenframe(s) in [Ereignisstromkodierung](#), der/die Metadaten und Roh-Audio-Bytes enthält/enthalten.
- Ein Endframe. Hierbei handelt es sich um eine signierte Nachricht in der [Ereignis-Stream-Kodierung](#) mit einem leeren Text.

Note

Amazon Transcribe unterstützt nur einen Stream pro WebSocket Sitzung. Wenn Sie versuchen, mehrere Streams zu verwenden, schlägt Ihre Transkriptionsanfrage fehl.

1. Fügen Sie der IAM Rolle, die die Anfrage stellt, die folgende Richtlinie hinzu. Weitere Informationen finden Sie unter [IAM Richtlinien hinzufügen](#).

JSON

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

2. Um die Sitzung zu starten, erstellen Sie eine vordefinierte URL in folgendem Format. Für eine bessere Lesbarkeit werden Zeilenumbrüche hinzugefügt.

```
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

Der Höchstwert für X-Amz-Expires ist 300 (5 Minuten).

Zusätzliche Operationen und Parameter sind in der [API-Referenz](#) aufgeführt. Parameter, die allen AWS -API-Operationen gemeinsam sind, werden im Abschnitt [Allgemeine Parameter](#) aufgeführt.

Um die URL für Ihre Anfrage zu konstruieren und die [Signature Version 4](#) zu erstellen, führen Sie die folgenden Schritte aus. Die Beispiele sind in Pseudocode.

- a. Erstellen Sie eine kanonische Anforderung. Eine kanonische Anfrage ist eine Zeichenfolge, die Informationen aus Ihrer Anfrage in einem standardisierten Format enthält. Dadurch wird sichergestellt, dass beim AWS Empfang der Anfrage dieselbe Signatur berechnet werden kann, die Sie für Ihre URL erstellt haben. Weitere Informationen finden Sie unter [Erstellen eines kanonischen Antrags für Signature Version 4](#).

```
# 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. Erstellen Sie einen kanonischen URI, d. h. den Teil des URI, der zwischen der Domain und dem Abfrage-String liegt.

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

- c. Erstellen Sie die kanonischen Header und signierten Header. Beachten Sie das abschließende `\n` in den kanonischen Headern.
- Fügen Sie den Namen des Headers in Kleinbuchstaben gefolgt von einem Doppelpunkt (`:`) an.
 - Fügen Sie eine durch Komma getrennte Liste der Werte für diesen Header an. Sortieren Sie keine Werte in Headern, die mehrere Werte enthalten.
 - Eine neue Zeile anfügen (`\n`).

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

- d. Gleichen Sie den Algorithmus mit dem Hash-Algorithmus ab. Verwenden Sie SHA-256.

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

- e. Erstellen Sie den Berechtigungsbereich, der den abgeleiteten Schlüssel auf das Datum, AWS-Region und den Service ausdehnt. Beispiel, `20220127/us-west-2/transcribe/aws4_request`.

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

- f. Erstellen Sie die kanonische Abfragezeichenfolge. Die Werte der Abfragezeichenfolge müssen URI-encoded nach Namen sortiert sein.
- Sortieren Sie die Parameternamen nach Zeichencodepunkt in aufsteigender Reihenfolge. Parameter mit doppelten Namen sollten nach Wert sortiert werden. So wird beispielsweise ein Parameternamen, der mit dem Großbuchstaben F beginnt, einem Parameternamen, der mit dem Kleinbuchstaben b beginnt, vorangestellt.
 - Verwenden Sie URI-encode keines der nicht reservierten Zeichen, die in RFC 3986 definiert sind: A-Z, a-z, 0-9, Bindestrich (`-`), Unterstrich (`_`), Punkt (`.`) und Tilde (`~`).

- Percent-encode alle anderen Zeichen mit %XY, wobei X und Y Hexadezimalzeichen (0-9 und Großbuchstaben) sind. A-F Beispielsweise muss das Leerzeichen als %20 codiert sein (kein '+' enthalten, wie es bei einigen Kodierungsschemata der Fall ist); erweiterte UTF-8 Zeichen müssen die Form %XY%ZA%BC haben.
- Double-encode alle Gleichheitszeichen (=) in Parameterwerten.

```
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. Erstellen Sie ein Hash der Nutzlast. Bei einer Anfrage an GET ist die Nutzlast eine leere Zeichenkette.

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

- h. Kombinieren Sie die folgenden Elemente, um die kanonische Anfrage zu erstellen.

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

3. Erstellen Sie die zu signierende Zeichenkette, die Metainformationen über Ihre Anfrage enthält. Sie verwenden die zu signierende Zeichenfolge im nächsten Schritt, wenn Sie die Anforderungssignatur berechnen. Weitere Informationen finden Sie unter [Erstellen einer zu signierenden Zeichenfolge für Signature Version 4](#).

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

4. Berechnen Sie die Signatur. Um dies zu tun, leiten Sie einen Signaturschlüssel aus Ihrem AWS geheimen Zugriffsschlüssel ab. Für ein höheres Maß an Schutz ist der abgeleitete Schlüssel spezifisch für das Datum, den Service und AWS-Region. Verwenden Sie diesen abgeleiteten Schlüssel, um die Anfrage zu signieren. Weitere Informationen finden Sie unter [AWS Signatur für Signaturversion 4 berechnen](#).

Stellen Sie sicher, dass Sie die Funktion `GetSignatureKey` implementieren, um Ihren Signaturschlüssel abzuleiten. Wenn Sie noch keinen Signaturschlüssel abgeleitet haben, lesen Sie den Abschnitt [Beispiele für die Ableitung eines Signaturschlüssels für die Signature Version 4](#).

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

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

Die Funktion `HMAC(key, data)` stellt eine HMAC-SHA256 Funktion dar, die Ergebnisse im Binärformat zurückgibt.

5. Fügen Sie der Anfrage Signierinformationen hinzu und erstellen Sie die Anfrage-URL.

Nachdem Sie die Signatur berechnet haben, fügen Sie sie zur Abfragezeichenfolge hinzu. Weitere Informationen finden Sie unter [Hinzufügen der Signatur zur Anfrage](#).

Fügen Sie zunächst die Authentifizierungsinformationen in die Abfragezeichenfolge ein.

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

Zweitens: Erstellen Sie die URL für die Anfrage.

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

Verwenden Sie die Anforderungs-URL zusammen mit Ihrer WebSocket Bibliothek, an die Sie die Anfrage richten möchten Amazon Transcribe.

6. Die Anfrage an Amazon Transcribe muss die folgenden Header enthalten. In der Regel werden diese Header von Ihrer WebSocket Client-Bibliothek verwaltet.

```
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. Wenn Ihre Amazon Transcribe WebSocket Anfrage eingeht, antwortet sie mit einer WebSocket Upgrade-Antwort. In der Regel verwaltet Ihre WebSocket Bibliothek diese Antwort und richtet einen Socket für die Kommunikation mit ein Amazon Transcribe.

Das Folgende ist die Antwort von Amazon Transcribe. Für eine bessere Lesbarkeit werden Zeilenumbrüche hinzugefügt.

```
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. Stellen Sie Ihre WebSocket Streaming-Anfrage.

Nachdem die WebSocket Verbindung hergestellt wurde, kann der Client mit dem Senden einer Sequenz von Audioframes beginnen, die jeweils mithilfe der [Event-Stream-Kodierung](#) codiert sind.

Jeder Datenrahmen enthält drei Header, die mit einem Teil der Rohdaten kombiniert werden; die folgende Tabelle beschreibt diese Header.

Bytelänge des Header-Namens	Header-Name (Zeichenfolge)	Werttyp des Header	Bytelänge der Wertzeichenfolge	Wert: Zeichenfolge () UTF-8
13	:content-type	7	24	application/octet-stream
11	:event-type	7	10	AudioEvent
13	:message-type	7	5	event

- Um den Datenströmen zu beenden, senden Sie einen leeren Audiochunk in einer ereignisstromkodierten Nachricht.

Die Antwort enthält Ereignis-Stream-kodierte Rohbytes in der Nutzlast. Sie enthält die Standardeinleitung und die folgenden Header:

Bytelänge des Header-Namens	Header-Name (Zeichenfolge)	Werttyp des Header	Bytelänge der Wertzeichenfolge	Wert: Zeichenfolge () UTF-8
13	:content-type	7	16	application/json
11	:event-type	7	15	TranscriptEvent
13	:message-type	7	5	event

Wenn Sie die binäre Antwort dekodieren, erhalten Sie eine JSON-Struktur, die die Transkriptionsergebnisse enthält.

Umgang mit WebSocket Streaming-Fehlern

Wenn bei der Bearbeitung Ihrer Anfrage eine Ausnahme auftritt, Amazon Transcribe antwortet die Antwort mit einem WebSocket Terminalframe, der eine in einem Eventstream kodierte Antwort

enthält. Diese Antwort enthält die in der folgenden Tabelle beschriebenen Header. Der Hauptteil der Antwort enthält eine beschreibende Fehlermeldung. Amazon Transcribe Sendet nach dem Senden der Ausnahmeantwort einen geschlossenen Frame.

Bytelänge des Header-Namens	Header-Name (Zeichenfolge)	Werttyp des Header	Bytelänge der Wertzeichenfolge	Wertzeichenfolge (UTF-8)
13	:content-type	7	16	application/json
15	:exception-type	7	variiert	variiert, siehe unten
13	:message-type	7	9	Ausnahme

Der `exception-type`-Header enthält einen der folgenden Werte.

- **BadRequestException:** Bei der Erstellung des Streams ist ein Client-Fehler aufgetreten, oder es ist ein Fehler beim Streaming von Daten aufgetreten. Stellen Sie sicher, dass Ihr Client bereit ist, Daten zu akzeptieren, und versuchen Sie es erneut.
- **InternalFailureException:** Amazon Transcribe hatte ein Problem beim Handshake mit dem Client. Versuchen Sie es erneut.
- **LimitExceededException:** Der Client hat das Limit für gleichzeitige Streams überschritten. Weitere Informationen finden Sie unter [Amazon Transcribe Limits](#). Reduzieren Sie die Anzahl der Streams, die Sie transkribieren.
- **UnrecognizedClientException:** Die WebSocket Upgrade-Anfrage wurde mit einem falschen Zugriffs- oder Geheimschlüssel signiert. Vergewissern Sie sich, dass Sie den Zugriffsschlüssel korrekt erstellt haben, und versuchen Sie Ihre Anfrage erneut.

Amazon Transcribe kann auch alle gängigen Servicefehler zurückgeben. Eine Liste finden Sie unter [Häufige Fehler](#).

Ereignis-Stream-Kodierung

Amazon Transcribe verwendet ein Format namens Event Stream Encoding für Streaming-Transkriptionen.

Die Ereignis-Stream-Kodierung stellt eine bidirektionale Kommunikation zwischen einem Client und einem Server bereit. Datenframes, die an den Amazon Transcribe Streaming-Dienst gesendet werden, werden in diesem Format codiert. Die Antwort von verwendet Amazon Transcribe ebenfalls diese Kodierung.

Jede Nachricht besteht aus zwei Abschnitten: der Einleitung und den Daten. Die Einleitung setzt sich folgendermaßen zusammen:

1. Die Byte-Länge der Nachricht
2. Die kombinierte Bytelänge aller Header

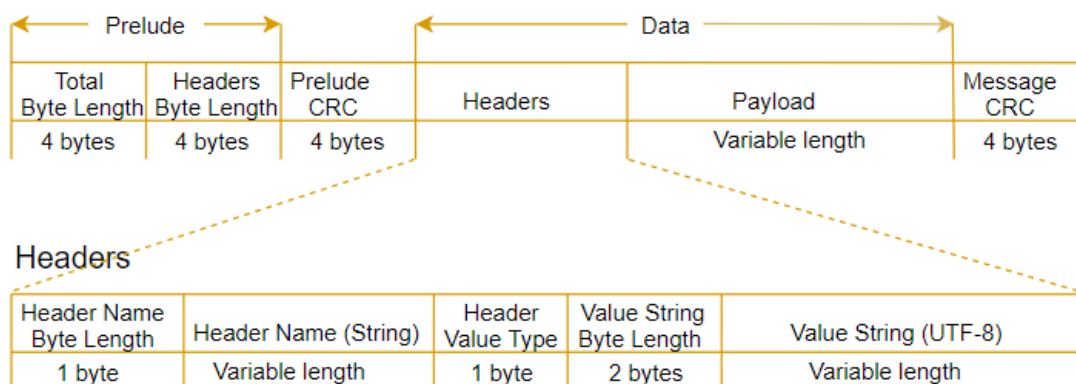
Der Datenabschnitt setzt sich folgendermaßen zusammen:

1. Kopfzeilen
2. Nutzlast

Jeder Abschnitt endet mit einer 4-Byte-Big-Endian-Integer-CRC-Prüfsumme (Cyclic Redundancy Check). Die Meldung CRC-Prüfsumme gilt sowohl für den Einleitungsabschnitt als auch für den Datenabschnitt. Amazon Transcribe verwendet CRC32 (oft als GZIP CRC32 bezeichnet), um beide CRCs zu berechnen. Weitere Informationen zu CRC32 finden Sie unter [GZIP-Dateiformatspezifikation Version 4.3](#).

Der gesamte Nachrichten-Overhead, einschließlich der Einleitung und beider Prüfsummen, beträgt 16 Bytes.

Das folgende Diagramm zeigt die Komponenten, aus denen eine Nachricht und ein Header bestehen. Pro Nachricht gibt es mehrere Header.



Jede Nachricht enthält die folgenden Komponenten:

- **Prelude:** Besteht aus zwei 4-Byte-Feldern, also insgesamt 8 Byte.
 - Die ersten 4 Bytes: Die Big-Endian-Integer-Byte-Länge der gesamten Nachricht, einschließlich dieses 4-Byte-Längenfeldes.
 - Zweite 4 Bytes: Die Big-Endian-Integer-Byte-Länge des „Header“-Teils der Nachricht, mit Ausnahme des „Header“-Längenfeldes selbst.
- **Einleitungs-CRC:** die 4-Byte-CRC-Prüfsumme des Einleitungsteils der Nachricht, mit Ausnahme der CRC selbst. Das Prelude hat einen von der Nachricht getrennten CRC. Dadurch wird sichergestellt, dass beschädigte Informationen mit Bytelänge sofort erkannt werden Amazon Transcribe können, ohne dass es zu Fehlern wie Pufferüberläufen kommt.
- **Header:** Metadaten, die die Nachricht annotieren, z. B. Nachrichtentyp und Inhaltstyp. Nachrichten haben mehrere Header, bei denen es sich um Schlüssel/Wert-Paare handelt, wobei der Schlüssel eine Zeichenfolge ist. UTF-8 Header können in beliebiger Reihenfolge im „Header“-Teil der Nachricht erscheinen, und jeder Header darf nur einmal vorkommen.
- **Nutzlast:** die zu transkribierenden Audioinhalte.
- **Nachrichten-CRC:** die 4-Byte-CRC-Prüfsumme vom Beginn der Nachricht bis zum Beginn der Prüfsumme. Also alles in der Nachricht bis auf die CRC selbst.

Der Header-Frame ist der Autorisierungs-Frame für die Streaming-Transkription. Amazon Transcribe verwendet den Wert des Autorisierungsheaders als Ausgangswert für die Generierung einer Kette von Autorisierungsheadern für die Datenframes in der Anfrage.

Jeder Header enthält die folgenden Komponenten; es gibt mehrere Header pro Frame.

- **Byte-Länge des Header-Namens:** die Byte-Länge des Header-Namens.
- **Header-Name:** Der Name des Headers, der den Header-Typ angibt. Gültige Werte finden Sie in den folgenden Frame-Beschreibungen.
- **Typ des Header-Werts:** Eine Zahl, die den Header-Wert angibt. Die folgende Liste zeigt die möglichen Werte für den Header und was sie bedeuten.
 - 0 – WAHR
 - 1 – FALSCH
 - 2 – BYTE
 - 3 – KURZ
 - 4 – INTEGER
 - 5 – LANG

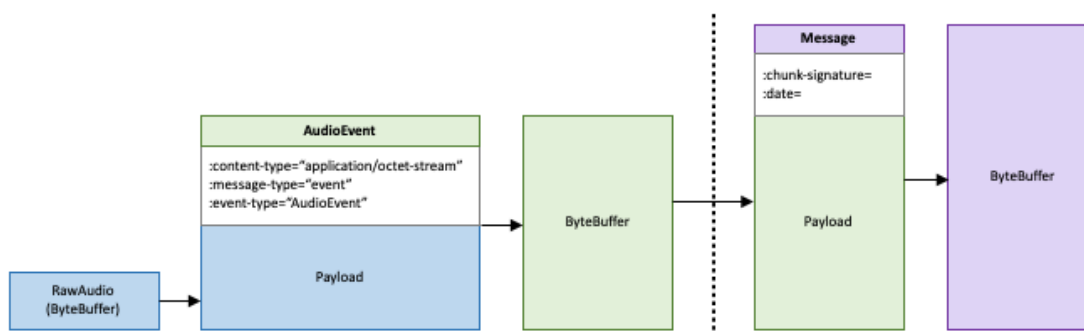
- 6 – BYTE ARRAY
 - 7 – Zeichenfolge
 - 8 – ZEITSTEMPEL
 - 9 – UUID
- Byte-Länge der Wertzeichenfolge: die Byte-Länge der Header-Wertzeichenfolge.
 - Header-Wert: der Wert der Header-Zeichenfolge. Gültige Werte für dieses Feld sind vom Header-Typ abhängig. Weitere Informationen finden Sie unter [Einen HTTP/2 Stream einrichten](#) oder [Einen WebSocket Stream einrichten](#).

Datenframes

Jede Streaming-Anfrage enthält einen oder mehrere Data Frames. Zum Erstellen eines Datenframes sind die folgenden zwei Schritte erforderlich:

1. Kombinieren Sie Audio-Rohdaten mit Metadaten, um die Nutzdaten Ihrer Anfrage zu erstellen.
2. Kombinieren Sie die Nutzlast mit einer Signatur, um die Ereignisnachricht zu erstellen, die an Amazon Transcribe gesendet wird.

In der folgenden Abbildung ist diese Funktionsweise dargestellt.



Auftragswarteschlangen

Mit der Warteschlangenfunktion können Sie mehr Transkriptionsaufträge übermitteln, als gleichzeitig bearbeitet werden können. Ohne Auftragswarteschlange müssen Sie, sobald Sie das Quota der zulässigen gleichzeitigen Anforderungen erreicht haben, warten, bis eine oder mehrere Anforderungen abgeschlossen sind, bevor Sie eine neue Anforderung senden.

Die Warteschleife von Job ist sowohl für Transkriptionsaufträge als auch für Analytics-Jobanfragen nach dem Anruf optional.

Wenn Sie die Auftragswarteschlange aktivieren, Amazon Transcribe wird eine Warteschlange erstellt, die alle Anfragen enthält, die Ihr Limit überschreiten. Sobald eine Anfrage abgeschlossen ist, wird eine neue Anfrage aus Ihrer Warteschlange gezogen und bearbeitet. Anfragen in der Warteschlange werden in einer FIFO-Reihenfolge (first in, first out) bearbeitet.

Sie können bis zu 10.000 Aufträge zu Ihrer Warteschlange hinzufügen. Wenn Sie diese Grenze überschreiten, erhalten Sie eine Fehlermeldung `LimitExceededConcurrentJobException`. Um eine optimale Leistung aufrechtzuerhalten, werden Amazon Transcribe nur bis zu 90 Prozent Ihres Kontingents (ein Bandbreitenverhältnis von 0,9) für die Verarbeitung von Aufträgen in der Warteschlange verwendet. Beachten Sie, dass dies Standardwerte sind, die auf Anfrage erhöht werden können.

Tip

Eine Liste der Standardlimits und Kontingente für Amazon Transcribe Ressourcen finden Sie in der [AWS Allgemeinen Referenz](#). Einige dieser Standardwerte können auf Anfrage erhöht werden.

Wenn Sie die Auftragswarteschlange aktivieren, aber das Quota für gleichzeitige Anforderungen nicht überschreiten, werden alle Anforderungen gleichzeitig verarbeitet.

Aktivieren der Auftragswarteschlange

Sie können die Auftragswarteschlange mit den SDKs AWS-Managementkonsole, AWS CLI oder AWS aktivieren. Beispiele finden Sie im Folgenden. Beispiele finden Sie im Folgenden:

AWS-Managementkonsole

1. Melden Sie sich an der [AWS-Managementkonsole](#) an.
2. Wählen Sie im Navigationsbereich Transkriptionsaufträge und dann Auftrag erstellen (oben rechts). Dies öffnet die Seite Auftragsdetails angeben.
3. Im Feld Auftragseinstellungen gibt es ein Feld Zusätzliche Einstellungen. Wenn Sie diesen Bereich erweitern, können Sie das Kästchen Zur Auftragswarteschlange hinzufügen aktivieren, um die Auftragswarteschlange zu aktivieren.

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**

4. Füllen Sie alle anderen Felder aus, die Sie auf der Seite Auftragsdetails angeben möchten, und wählen Sie dann Weiter. Hier gelangen Sie zur Seite Auftrag konfigurieren – optional .
5. Wählen Sie Auftrag erstellen, um Ihren Transkriptionsauftrag auszuführen.

AWS CLI

In diesem Beispiel werden der Befehl [start-Transkription-job](#) und der Parameter `job-execution-settings` mit dem Unterparameter `AllowDeferredExecution` verwendet. Beachten Sie, dass Sie, wenn Sie `AllowDeferredExecution` in Ihrer Anfrage angeben, auch `DataAccessRoleArn` angeben müssen.

Weitere Informationen erhalten Sie unter [StartTranscriptionJob](#) und [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
```

Hier ein weiteres Beispiel mit dem Befehl [start-Transkription-job](#) und einem Anforderungstext, der die Warteschlangenbildung aktiviert.

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

Die Datei `my-first-queueing-request.json` enthält den folgenden Anforderungstext.

```
{  
  "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"  
  }  
}
```

```
}
}
```

AWS SDK für Python (Boto3)

In diesem Beispiel wird das AWS SDK für Python (Boto3) zur Aktivierung der Job-Warteschleife verwendet, wobei das `AllowDeferredExecution` Argument für die Methode `start_transcription_job` verwendet wird. Beachten Sie, dass Sie, wenn Sie `AllowDeferredExecution` in Ihrer Anfrage angeben, auch `DataAccessRoleArn` angeben müssen. Weitere Informationen erhalten Sie unter [StartTranscriptionJob](#) und [JobExecutionSettings](#).

Weitere Beispiele für die Verwendung der AWS SDKs, einschließlich funktionspezifischer, szenarienspezifischer und serviceübergreifender Beispiele, finden Sie im Kapitel. [Codebeispiele für Amazon Transcribe mit AWS SDKs](#)

```
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)
```

Sie können den Status eines Auftrags in der Warteschlange über AWS-Managementkonsole oder überprüfen, indem Sie eine Anfrage einreichen. [GetTranscriptionJob](#) Wenn ein Auftrag in die Warteschlange gestellt wird, lautet der Status QUEUED. Der Status ändert sich zu IN_PROGRESS, sobald die Bearbeitung Ihres Auftrags beginnt, und wechselt dann zu COMPLETED oder FAILED, wenn die Bearbeitung abgeschlossen ist.

Taggen von Ressourcen

Ein Tag ist eine benutzerdefinierte Metadatenbeschriftung, die Sie einer Ressource hinzufügen können, um sie leichter identifizieren, organisieren und in einer Suche finden zu können. Tags bestehen aus zwei einzelnen Teilen: Einem Tag-Schlüssel und einem Tag-Wert. Dies wird als Schlüssel:Wert-Paar bezeichnet.

Ein Tag-Schlüssel steht in der Regel für eine größere Kategorie, während ein Tag-Wert eine Teilmenge dieser Kategorie darstellt. Sie könnten zum Beispiel `tag key=Color` und `tag value=Blue` haben, was das Schlüssel:Wert-Paar `Color:Blue` ergeben würde. Beachten Sie, dass Sie den Wert eines Tags auf eine leere Zeichenkette setzen können, nicht aber den Wert eines Tags auf null. Ein nicht angegebener Tag-Wert entspricht einer leeren Zeichenfolge.

Tip

AWS Fakturierung und Kostenmanagement kann Tags verwenden, um Ihre Rechnungen in dynamische Kategorien zu unterteilen. Wenn Sie zum Beispiel Tags hinzufügen, die verschiedene Abteilungen in Ihrem Unternehmen repräsentieren, wie `Department:Sales` oder `Department:Legal`, kann AWS Ihnen Ihre Kostenverteilung pro Abteilung anzeigen.

In Amazon Transcribe können Sie die folgenden Ressourcen taggen:

- Transkriptionsaufträge
- Medical-Transkriptionsaufträge
- Rufen Sie Analyseaufträge nach dem Anruf zur Transkription auf
- Benutzerdefinierte Vokabulare
- Kundenspezifische medizinische Vokabulare
- Benutzerdefinierte Vokalbularfilter
- Rufen Sie Analyse-Kategorien auf
- Benutzerdefinierte Sprachmodelle

Tag-Schlüssel können bis zu 128 Zeichen lang sein und Tag-Werte können bis zu 256 Zeichen lang sein. In beiden Fällen wird zwischen Groß- und Kleinschreibung unterschieden. Amazon Transcribe

unterstützt bis zu 50 Tags pro Ressource. Für eine bestimmte Ressource muss jeder Tag-Schlüssel eindeutig sein und darf nur einen Wert haben. Beachten Sie, dass Ihre Tags nicht mit `aws:` beginnen dürfen, da dieses Präfix für vom System generierte Tags AWS reserviert ist. Sie können keine `aws:*`-Tags hinzufügen, ändern oder löschen, und sie werden nicht auf die Anzahl der Tags pro Ressource angerechnet.

 Spezifische API-Vorgänge für die Kennzeichnung von Ressourcen

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

Um die Tagging-APIs zu nutzen, müssen Sie Ihrer Anfrage einen Amazon Resource Name (ARN) beifügen. ARNs haben das Format `arn:partition:service:region:account-id:resource-type/resource-id`. Der ARN für einen Transkriptionsauftrag kann zum Beispiel so aussehen: `arn:aws:transcribe:us-west-2:111122223333:transcription-job/my-transcription-job-name`.

[Weitere Informationen zum Tagging, einschließlich bewährter Methoden, finden Sie unter Tagging-Ressourcen. AWS](#)

Tag-based Zugriffskontrolle

Sie können Tags verwenden, um den Zugriff in Ihrem zu kontrollieren AWS-Konten. Bei der tagbasierten Zugriffskontrolle geben Sie Taginformationen im Bedingungelement einer IAM Richtlinie an. Sie können dann Tags und den zugehörigen Tag-Bedingungsschlüssel verwenden, um den Zugriff auf Folgendes zu steuern:

- **Ressourcen:** Steuern Sie den Zugriff auf Ihre Amazon Transcribe Ressourcen anhand der Tags, die Sie diesen Ressourcen zugewiesen haben.
 - Verwenden Sie den `aws:ResourceTag/key-name`-Bedingungsschlüssel, um anzugeben, welches Tag-Schlüssel:Wert-Paar an die Ressource angehängt werden muss.
- **Abfragen:** Steuern Sie, welche Tags in einer Anfrage übergeben werden können.
 - Verwenden Sie den `aws:RequestTag/key-name` Bedingungsschlüssel, um anzugeben, welche Tags einem IAM Benutzer oder einer Rolle hinzugefügt, geändert oder daraus entfernt werden können.
- **Autorisierungsprozesse:** Steuern Sie den Tag-basierten Zugriff für jeden Teil Ihres Autorisierungsprozesses.

- Verwenden Sie den `aws:TagKeys/-`Bedingungsschlüssel, um zu steuern, ob bestimmte Tag-Schlüssel auf einer Ressource, in einer Anfrage oder von einem Auftraggeber verwendet werden können. In diesem Fall spielt der Schlüsselwert keine Rolle.

Ein Beispiel für eine Tag-basierte Zugriffskontrollrichtlinie finden Sie unter [Anzeigen von Transkriptionsaufträgen basierend auf Tags](#).

Ausführlichere Informationen zur Tag-basierten Zugriffskontrolle finden Sie unter [Steuerung des Zugriffs auf AWS -Ressourcen mithilfe von Tags](#).

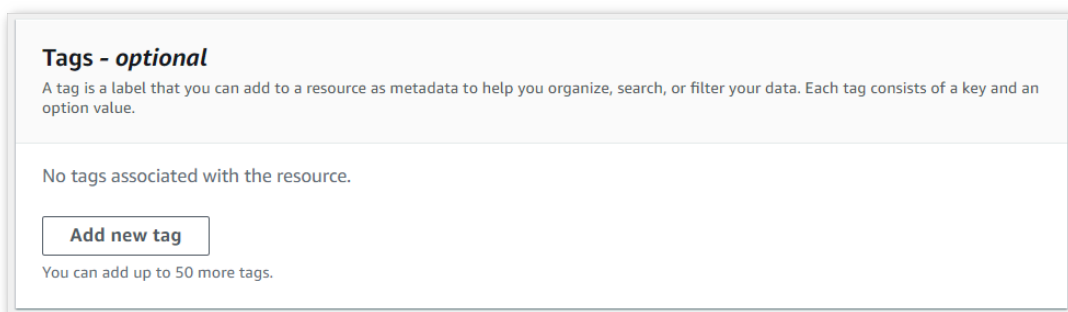
Hinzufügen von Tags zu Ihrem Amazon Transcribe Ressourcen

Sie können Tags vor oder nach der Ausführung Ihres Amazon Transcribe Jobs hinzufügen. Mit den bestehenden APIs `Create*` und `Start*` können Sie mit Ihrer Transkriptionsanfrage Tags hinzufügen.

Sie können Tags mit Hilfe der SDKs AWS-Managementkonsole, AWS CLI, oder AWS hinzufügen, ändern oder löschen. Beispiele finden Sie im Folgenden:

AWS-Managementkonsole

1. Melden Sie sich an der [AWS-Managementkonsole](#) an.
2. Wählen Sie im Navigationsbereich Transkriptionsaufträge und dann Auftrag erstellen (oben rechts). Dies öffnet die Seite Auftragsdetails angeben.
3. Scrollen Sie auf der Seite Auftragsdetails angeben bis zum Ende, um das Feld Tags – optional zu finden, und wählen Sie Neues Tag hinzufügen.



4. Geben Sie Informationen für das Feld Schlüssel und, optional, für das Feld Wert ein.

Tags - optional

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

Key

Q color

X

Value - optional

Q blue

X

Remove

Add new tag

You can add up to 49 more tags.

5. Füllen Sie alle anderen Felder aus, die Sie auf der Seite Auftragsdetails angeben möchten, und wählen Sie dann Weiter. Hier gelangen Sie zur Seite Auftrag konfigurieren – optional .

Wählen Sie Auftrag erstellen, um Ihren Transkriptionsauftrag auszuführen.

6. Sie können die mit einem Transkriptionsauftrag verbundenen Tags anzeigen, indem Sie zur Seite Transkriptionsaufträge navigieren, einen Transkriptionsauftrag auswählen und auf der Informationsseite dieses Auftrags bis zum Ende scrollen. Wenn Sie Ihre Tags bearbeiten möchten, können Sie dies tun, indem Sie Tags verwalten wählen.

Tags (2)

Manage Tags

Key	Value
color	blue

AWS CLI

In diesem Beispiel werden der Befehl [start-Transkription-job](#) und der Parameter Tags verwendet. Weitere Informationen erhalten Sie unter [StartTranscriptionJob](#) und [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
```

Hier ein weiteres Beispiel mit dem Befehl [start-transcription-job](#) und einem Anforderungstext, der diesem Auftrag Tags hinzufügt.

Hinzufügen von Tags zu Ihrem Amazon Transcribe Ressourcen

202

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

Die Datei `my-first-tagging-job.json` enthält den folgenden Anforderungstext.

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

AWS SDK für Python (Boto3)

Im folgenden Beispiel wird mithilfe des AWS SDK für Python (Boto3) Tags Arguments für die Methode [start_transcription_job](#) ein Tag hinzugefügt. Weitere Informationen erhalten Sie unter [StartTranscriptionJob](#) und [Tag](#).

Weitere Beispiele für die Verwendung der AWS SDKs, einschließlich funktionspezifischer, szenarienspezifischer und serviceübergreifender Beispiele, finden Sie im Kapitel. [Codebeispiele für Amazon Transcribe mit AWS SDKs](#)

```
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)
```

Sprecherpartitionierung (Diarisierung)

Mit der Lautsprecher-Diarisierung können Sie in Ihrer Transkriptionsausgabe zwischen verschiedenen Sprechern unterscheiden. Amazon Transcribe kann zwischen maximal 30 einzelnen Sprechern unterscheiden und kennzeichnet den Text jedes einzelnen Sprechers mit einem eindeutigen Wert (spk_0 durch spk_9).

Zusätzlich zu den [Standard-Transkriptabschnitten](#) (transcripts und items) enthalten Anfragen mit aktivierter Sprecherpartitionierung einen Abschnitt speaker_labels. Dieser Abschnitt ist nach Sprechern gruppiert und enthält Informationen zu jeder Äußerung, einschließlich Sprecherbezeichnung und Zeitstempel.

```
"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"
        }
      ],
    },
  ],
}
```

Ein vollständiges Beispieltranskript mit Sprecherpartitionierung (für zwei Sprecher) finden Sie unter [Beispiel für die Diarisierungsausgabe \(Batch\)](#).

Aufteilung der Sprecher in einer Batch-Transkription

Zur Sprecherpartitionierung in einer Batch-Transkription siehe die folgenden Beispiele:

AWS-Managementkonsole

1. Melden Sie sich an der [AWS-Managementkonsole](#) an.
2. Wählen Sie im Navigationsbereich Transkriptionsaufträge und dann Auftrag erstellen (oben rechts). Dies öffnet die Seite Auftragsdetails angeben.

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. Füllen Sie alle Felder aus, die Sie auf der Seite Auftragsdetails angeben möchten, und wählen Sie dann Weiter. Dadurch gelangen Sie zur Seite Auftrag konfigurieren – optional.

Um die Lautsprecherpartitionierung zu aktivieren, wählen Sie in den Audioeinstellungen die Option Audioidentifikation. Wählen Sie dann Lautsprecherpartitionierung und geben Sie die Anzahl der Lautsprecher an.

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. Wählen Sie Auftrag erstellen, um Ihren Transkriptionsauftrag auszuführen.

AWS CLI

In diesem Beispiel wird [start-Transkription-job](#) verwendet. Weitere Informationen finden Sie unter [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
```

Hier ein weiteres Beispiel mit dem Befehl [start-Transkription-job](#) und einem Anforderungstext, der die Sprecherpartitionierung mit diesem Auftrag aktiviert.

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

Die Datei *my-first-Transkription-job.json* enthält den folgenden Anforderungstext.

```
{
  "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
}
```

AWS SDK für Python (Boto3)

In diesem Beispiel werden Kanäle mithilfe der AWS SDK für Python (Boto3) Methode [start_transcription_job](#) identifiziert. Weitere Informationen finden Sie unter [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)
```

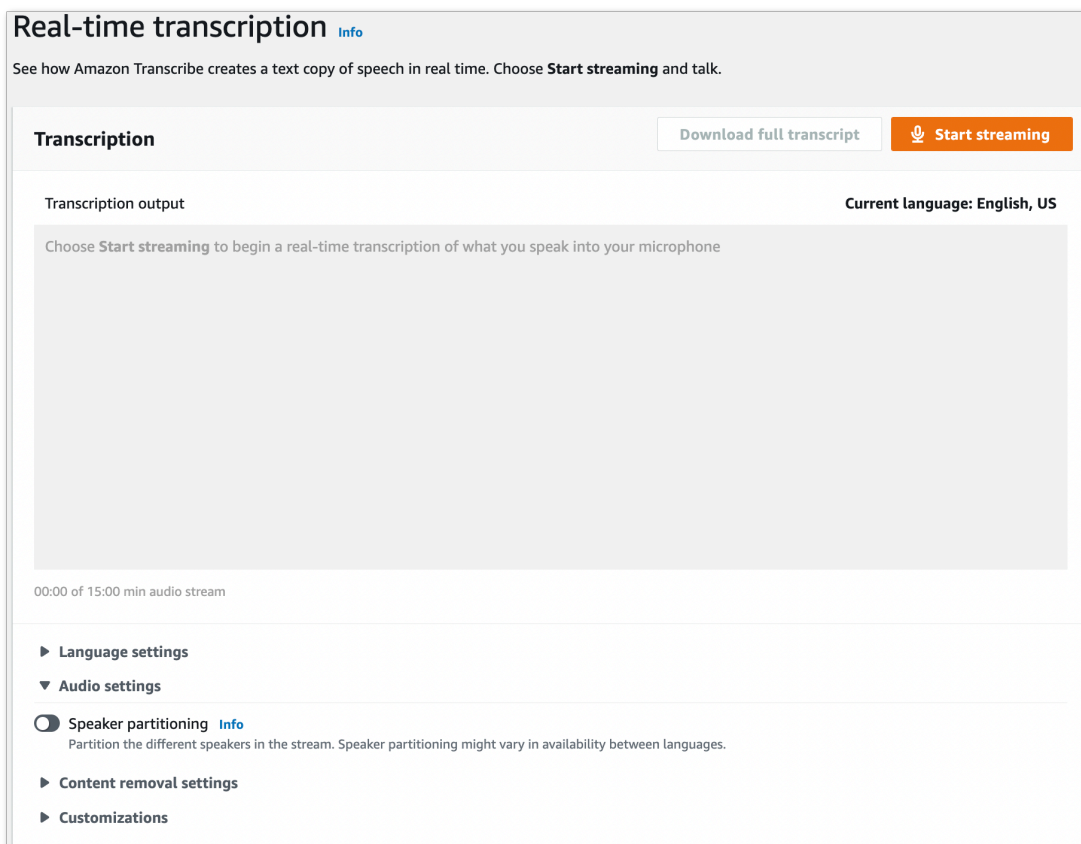
```
print(status)
```

Partitionierung der Sprecher in einer Streaming-Transkription

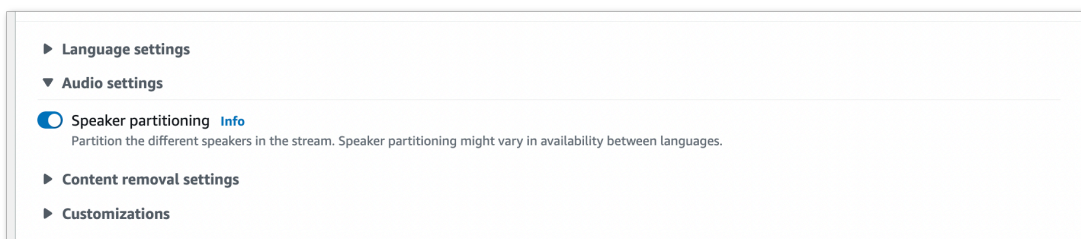
Um die Sprecher in einer Streaming-Transkription zu trennen, sehen Sie sich die folgenden Beispiele an:

Streaming-Transkriptionen

1. Melden Sie sich an der [AWS-Managementkonsole](#) an.
2. Wählen Sie im Navigationsbereich Real-time Transkription aus. Blättern Sie nach unten zu den Audioeinstellungen und erweitern Sie dieses Feld, falls es minimiert ist.



3. Aktivieren Sie die Sprecherpartitionierung.



4. Jetzt können Sie Ihren Stream transkribieren. Wählen Sie Streaming starten und beginnen Sie zu sprechen. Um Ihr Diktat zu beenden, wählen Sie Streaming beenden.

HTTP/2 streamen

In diesem Beispiel wird eine HTTP/2 Anfrage erstellt, die Lautsprecher in Ihrer Transkriptionsausgabe partitioniert. Weitere Informationen zur Verwendung von HTTP/2 Streaming mit finden Sie Amazon Transcribe unter [Einen HTTP/2 Stream einrichten](#). Weitere Einzelheiten zu Parametern und Kopfzeilen, die speziell für Amazon Transcribe gelten, finden Sie unter [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
```

Parameterdefinitionen finden Sie in der [API-Referenz](#). Parameter, die allen AWS API-Vorgängen gemeinsam sind, sind im Abschnitt [Allgemeine Parameter](#) aufgeführt.

WebSocket streamen

In diesem Beispiel wird eine vorsignierte URL erstellt, die die Sprecher in Ihrer Transkriptionsausgabe trennt. Für eine bessere Lesbarkeit werden Zeilenumbrüche hinzugefügt. Weitere Hinweise zur Verwendung von WebSocket Streams mit Amazon Transcribe finden Sie unter [Einen WebSocket Stream einrichten](#). Weitere Einzelheiten zu den Parametern finden Sie unter [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
```

Parameterdefinitionen finden Sie in der [API-Referenz](#). Parameter, die allen AWS API-Vorgängen gemeinsam sind, sind im Abschnitt [Allgemeine Parameter](#) aufgeführt.

Beispiel für die Diarisierungsausgabe (Batch)

Hier ein Beispiel für die Ausgabe einer Batch-Transkription mit aktivierter Diarisierung.

```
{
  "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"  
}
```


Transkribieren von Mehrkanal-Audio

Wenn Ihr Audio über zwei Kanäle verfügt, können Sie die Kanalidentifikation verwenden, um die Sprache von jedem Kanal separat zu transkribieren. Amazon Transcribe unterstützt derzeit kein Audio mit mehr als zwei Kanälen.

In Ihrem Transkript sind die Kanäle mit den Bezeichnungen `ch_0` und `ch_1` versehen.

Zusätzlich zu den [standardmäßigen Transkriptionsabschnitten](#) (transcripts und items) enthalten Anfragen mit aktivierter Kanalidentifizierung einen Abschnitt `channel_labels`. Dieser Abschnitt enthält jede Äußerung oder jedes Satzzeichen, gruppiert nach Kanal, sowie die zugehörigen Kanalbezeichnungen, Zeitstempel und Konfidenzwert.

```
"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
  },
```

Wenn eine Person auf einem Kanal zur gleichen Zeit wie eine Person auf einem anderen Kanal spricht, überschneiden sich die Zeitstempel für jeden Kanal, während die Personen übereinander sprechen.

Ein vollständiges Beispieltranskript mit Kanalidentifikation finden Sie unter [Beispiel Kanalidentifikationsausgabe \(Batch\)](#).

Verwendung der Kanalidentifizierung in einer Batch-Transkription

Um Kanäle in einer Batch-Transkription zu identifizieren, können Sie die SDKs AWS-Managementkonsole, AWS CLI oder AWS verwenden. Beispiele finden Sie im Folgenden:

AWS-Managementkonsole

1. Melden Sie sich an der [AWS-Managementkonsole](#) an.
2. Wählen Sie im Navigationsbereich Transkriptionsaufträge und dann Auftrag erstellen (oben rechts). Dies öffnet die Seite Auftragsdetails angeben.

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. Füllen Sie alle Felder aus, die Sie auf der Seite Auftragsdetails angeben möchten, und wählen Sie dann Weiter. Dadurch gelangen Sie zur Seite Auftrag konfigurieren – optional.

Wählen Sie im Bereich Audioeinstellungen die Option Kanalidentifikation (unter der Überschrift „Art der Audioidentifikation“).

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. Wählen Sie Auftrag erstellen, um Ihren Transkriptionsauftrag auszuführen.

AWS CLI

In diesem Beispiel wird [start-Transkription-job](#) verwendet. Weitere Informationen finden Sie unter [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
```

Hier ein weiteres Beispiel mit dem Befehl [start-Transkription-job](#) und einem Anforderungstext, der die Kanalidentifizierung mit diesem Auftrag ermöglicht.

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

Die Datei *my-first-Transkription-job.json* enthält den folgenden Anforderungstext.

```
{
  "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
    }
}

```

AWS SDK für Python (Boto3)

In diesem Beispiel werden Kanäle mithilfe der AWS SDK für Python (Boto3) Methode [start_transcription_job](#) identifiziert. Weitere Informationen finden Sie unter [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)

```

Verwenden der Kanalidentifizierung in einer Streaming-Transkription

Um Kanäle in einer Streaming-Transkription zu identifizieren, können Sie HTTP/2 oder WebSockets verwenden. Beispiele finden Sie im Folgenden:

HTTP/2 streamen

In diesem Beispiel wird eine HTTP/2 Anfrage erstellt, die Kanäle in Ihrer Transkriptionsausgabe voneinander trennt. Weitere Informationen zur Verwendung von HTTP/2 Streaming mit Amazon Transcribe finden Sie unter [Einen HTTP/2 Stream einrichten](#). Weitere Einzelheiten zu Parametern und Kopfzeilen, die speziell für Amazon Transcribe gelten, finden Sie unter [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
```

Parameterdefinitionen finden Sie in der [API-Referenz](#). Parameter, die allen AWS API-Vorgängen gemeinsam sind, sind im Abschnitt [Allgemeine Parameter](#) aufgeführt.

WebSocket streamen

In diesem Beispiel wird eine vorzeichenbehaftete URL erstellt, die Kanäle in Ihrer Transkriptionsausgabe trennt. Für eine bessere Lesbarkeit werden Zeilenumbrüche hinzugefügt. Weitere Hinweise zur Verwendung von WebSocket Streams mit Amazon Transcribe finden Sie unter [Einen WebSocket Stream einrichten](#). Weitere Einzelheiten zu den Parametern finden Sie unter [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
&channel-identification=TRUE
```

Parameterdefinitionen finden Sie in der [API-Referenz](#). Parameter, die allen AWS API-Vorgängen gemeinsam sind, sind im Abschnitt [Allgemeine Parameter](#) aufgeführt.

Beispiel Kanalidentifikationsausgabe (Batch)

Hier ein Beispiel für die Ausgabe einer Batch-Transkription mit aktivierter Kanalidentifizierung.

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

Identifizierung der vorherrschenden Sprachen in Ihren Medien

Amazon Transcribe ist in der Lage, die in Ihren Medien gesprochenen Sprachen automatisch zu identifizieren, ohne dass Sie einen Sprachcode angeben müssen.

Die [Batch-Sprachidentifizierung](#) kann die in Ihrer Mediendatei vorherrschende Sprache identifizieren, oder, wenn Ihre Medien mehrere Sprachen enthalten, alle gesprochenen Sprachen identifizieren. Um die Genauigkeit der Sprachidentifizierung zu verbessern, können Sie optional eine Liste mit zwei oder mehr Sprachen angeben, von denen Sie annehmen, dass sie in Ihren Medien vorkommen.

Die [Streaming-Sprachidentifizierung](#) kann eine Sprache pro Kanal identifizieren (maximal zwei Kanäle werden unterstützt) oder, falls Ihr Stream mehrere Sprachen enthält, alle gesprochenen Sprachen identifizieren. Bei Streaming-Anfragen müssen mindestens zwei zusätzliche Sprachoptionen angegeben werden. Die Bereitstellung von Sprachoptionen ermöglicht eine schnellere Identifizierung der Sprache. Je schneller Amazon Transcribe die Sprache erkannt werden kann, desto geringer ist die Veränderung des Datenverlusts in den ersten Sekunden Ihres Streams.

Important

Batch- und Streaming-Transkriptionen unterstützen verschiedene Sprachen. Einzelheiten finden Sie in der Spalte Dateneingabe in der Tabelle [Unterstützte Sprachen](#). Beachten Sie, dass Schwedisch und Vietnamesisch derzeit nicht von der Sprachidentifizierung unterstützt werden.

Informationen zur Überwachung und zu Ereignissen mit Sprachidentifizierung finden Sie unter [Sprachenidentifikationsereignisse](#).

Sprachidentifizierung mit Batch-Transkriptionsaufträgen

Verwenden Sie die Batch-Sprachidentifizierung, um die Sprache(n) in Ihrer Mediendatei automatisch zu identifizieren.

Wenn Ihre Medien nur eine Sprache enthalten, können Sie die [einsprachige Identifizierung](#) aktivieren, die die dominante Sprache identifiziert, die in Ihrer Mediendatei gesprochen wird und Ihr Transkript nur mit dieser Sprache erstellt.

Wenn Ihre Medien mehr als eine Sprache enthalten, können Sie die [mehrsprachige Identifizierung](#) aktivieren, die alle in Ihrer Mediendatei gesprochenen Sprachen identifiziert und Ihr Transkript unter Verwendung jeder identifizierten Sprache erstellt. Beachten Sie, dass ein mehrsprachiges Transkript erstellt wird. Sie können andere Dienste nutzen, z. B. Amazon Translate um Ihr Transkript zu übersetzen.

In der Tabelle der [unterstützten Sprachen](#) finden Sie eine vollständige Liste der unterstützten Sprachen und der zugehörigen Sprachcodes.

Um optimale Ergebnisse zu erzielen, sollten Sie sicherstellen, dass Ihre Mediendatei mindestens 30 Sekunden Sprache enthält.

Anwendungsbeispiele mit dem AWS-Managementkonsole, AWS CLI, und dem AWS Python-SDK finden Sie unter [Sprachidentifizierung mit Batch-Transkriptionen](#).

Identifizierung von Sprachen in mehrsprachigen Audiodateien

Multi-language Die Identifizierung ist für mehrsprachige Mediendateien vorgesehen und bietet Ihnen ein Transkript, das alle [unterstützten Sprachen](#) wiedergibt, die in Ihren Medien gesprochen werden. Das bedeutet, dass Ihre Transkriptionsausgabe jede Sprache korrekt erkennt und transkribiert, wenn die Sprecher während des Gesprächs die Sprache wechseln oder wenn jeder Teilnehmer eine andere Sprache spricht. Wenn Ihre Medien beispielsweise einen zweisprachigen Sprecher enthalten, der abwechselnd in US-Englisch (en-US) und Hindi (hi-IN) spricht, kann die Mehrsprachenerkennung gesprochenes US-Englisch als en-US und gesprochenes Hindi als hi-IN identifizieren und transkribieren.

Dies unterscheidet sich von der einsprachigen Identifizierung, bei der nur eine dominante Sprache zur Erstellung eines Transkripts verwendet wird. In diesem Fall wird jede gesprochene Sprache, die nicht die dominante Sprache ist, falsch transkribiert.

Note

Schwärzen und benutzerdefinierte Sprachmodelle werden derzeit bei mehrsprachiger Identifizierung nicht unterstützt.

Note

Die folgenden Sprachen werden derzeit mit mehrsprachiger Identifizierung unterstützt: en-AB, en-AU, en-GB, en-IE, 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, He-IL, Hi-IN, ID-ID, fa-IR, it-IT, jA-JP, ko-KR, MS-MY, nl-NL, ru-RU, Ta-in, Te-in, Th-TR, tr-TR

Multi-language Transkripte bieten eine Zusammenfassung der erkannten Sprachen und die Gesamtzeit, in der jede Sprache in Ihren Medien gesprochen wurde. Hier ein Beispiel:

```
"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
    }
  ]
}
```

Verbesserung der Genauigkeit der Sprachidentifizierung

Bei der Sprachidentifizierung haben Sie die Möglichkeit, eine Liste von Sprachen anzugeben, von denen Sie glauben, dass sie in Ihren Medien vorkommen könnten. Durch das Einbeziehen von Sprachoptionen (LanguageOptions) Amazon Transcribe werden nur die Sprachen verwendet, die Sie bei der Zuordnung Ihres Audiomaterials zur richtigen Sprache angegeben haben. Dadurch kann die Sprachenerkennung beschleunigt und die Genauigkeit bei der Zuweisung des richtigen Sprachdialekts verbessert werden.

Wenn Sie Sprachcodes angeben möchten, müssen Sie mindestens zwei angeben. Es gibt keine Begrenzung für die Anzahl der Sprachcodes, die Sie einfügen können, aber wir empfehlen, zwischen zwei und fünf zu verwenden, um optimale Effizienz und Genauigkeit zu erzielen.

Note

Wenn Sie Ihrer Anfrage Sprachcodes beifügen und keiner der von Ihnen angegebenen Sprachcodes mit der Sprache oder den Sprachen übereinstimmt, die in Ihrem Audio identifiziert wurden, wird aus den von Ihnen angegebenen Sprachcodes die Sprache Amazon Transcribe ausgewählt, die am ehesten entspricht. Es wird dann eine Transkription in dieser Sprache erstellt. Wenn Ihre Medien beispielsweise in US-Englisch (en-US) verfasst sind und Sie die Sprachcodes angeben zh-CN, Amazon Transcribe ist es wahrscheinlich fr-FR, dass Ihre Medien Amazon Transcribe mit Deutsch (de-DE) übereinstimmen und eine German-language Transkription erstellt wird. de-DE Wenn Sprachcodes und gesprochene Sprachen nicht übereinstimmen, kann dies zu einem ungenauen Transkript führen, weshalb wir bei der Angabe von Sprachcodes zur Vorsicht raten.

Kombination der Sprachenidentifikation mit anderen Amazon Transcribe features

Sie können die Batch-Sprachidentifizierung in Kombination mit jedem anderen Amazon Transcribe - Feature verwenden. Wenn Sie die Sprachidentifizierung mit anderen Features kombinieren, sind Sie auf die Sprachen beschränkt, die von diesen Features unterstützt werden. Wenn Sie beispielsweise die Sprachenidentifikation zusammen mit der Inhaltsredaktion verwenden, sind Sie auf US-Englisch (en-US) oder US-Spanisch (es-US) beschränkt, da dies nur die Sprache ist, die für die Schwärzung verfügbar ist. Weitere Informationen finden Sie unter [Unterstützte Sprachen und sprachspezifische Funktionen](#).

Important

Wenn Sie die automatische Sprachenerkennung mit aktivierter Inhaltsredaktion verwenden und Ihr Audio andere Sprachen als US-Englisch (en-US) oder US-Spanisch (es-US) enthält, werden in Ihrem Transkript nur die Inhalte in US-Englisch oder US-Spanisch redigiert. Andere Sprachen können nicht geschwärzt werden, und es gibt keine Warnungen oder Fehlschläge.

Benutzerdefinierte Sprachmodelle, benutzerdefinierte Vokabulare und benutzerdefinierte Wortschatzfilter

Wenn Sie ein oder mehrere benutzerdefinierte Sprachmodelle, benutzerdefinierte Vokabulare oder benutzerdefinierte Wortschatzfilter zu Ihrer Sprachidentifikationsanfrage hinzufügen möchten, müssen Sie den [LanguageIdSettings](#)-Parameter enthalten. Sie können dann einen Sprachcode mit einem entsprechenden benutzerdefinierten Sprachmodell, einem benutzerdefinierten Vokabular und einem benutzerdefinierten Wortschatzfilter angeben. Beachten Sie, dass die mehrsprachige Identifizierung keine benutzerdefinierten Sprachmodelle unterstützt.

Es wird empfohlen, dass Sie `LanguageOptions` einschließen, wenn Sie [LanguageIdSettings](#) verwenden, um sicherzustellen, dass der richtige Sprachdialekt erkannt wird. Wenn Sie beispielsweise ein en-US benutzerdefiniertes Vokabular angeben, aber Amazon Transcribe feststellen, dass es sich bei der in Ihren Medien gesprochenen Sprache um dieses handelten-AU, wird Ihr benutzerdefiniertes Vokabular nicht auf Ihre Transkription angewendet. Wenn Sie `LanguageOptions` einschließen und en-US als einzigen englischen Dialekt angeben, wird Ihr benutzerdefiniertes Vokabular auf Ihre Transkription angewendet.

Für Beispiele von [LanguageIdSettings](#) in einer Anfrage finden Sie unter Option 2 in den SDKs AWS CLI und AWS in den Dropdown-Bereichen im Abschnitt [Sprachidentifizierung mit Batch-Transkriptionen](#).

Sprachidentifizierung mit Batch-Transkriptionen

Sie können die automatische Sprachidentifizierung in einem Batch-Transkriptionsauftrag mithilfe der AWS -, AWS-Managementkonsole- oder AWS CLI-SDKs verwenden. Beispiele finden Sie im Folgenden:

AWS-Managementkonsole

1. Melden Sie sich an der [AWS-Managementkonsole](#) an.
2. Wählen Sie im Navigationsbereich Transkriptionsaufträge und dann Auftrag erstellen (oben rechts). Dies öffnet die Seite Auftragsdetails angeben.
3. Suchen Sie im Bereich Auftragseinstellungen den Abschnitt Spracheinstellungen und wählen Sie Automatische Sprachidentifizierung oder Automatische Mehrsprachenidentifizierung.

Sie haben die Möglichkeit, mehrere Sprachen auszuwählen (aus dem Dropdown-Feld Sprachen auswählen), wenn Sie wissen, welche Sprachen in Ihrer Audiodatei enthalten sind. Die Bereitstellung von Sprachoptionen kann die Genauigkeit verbessern, ist aber nicht erforderlich.

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.

Select languages

☐ English, US (en-US)

☐ English, AU (en-AU)

☐ English, UK (en-GB)

☐ Hindi, IN (hi-IN)

☐ Spanish, US (es-US)

4. Füllen Sie alle anderen Felder aus, die Sie auf der Seite Auftragsdetails angeben möchten, und wählen Sie dann Weiter. Dadurch gelangen Sie zur Seite Auftrag konfigurieren – optional.

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. Wählen Sie Auftrag erstellen, um Ihren Transkriptionsauftrag auszuführen.

AWS CLI

In diesem Beispiel werden der Befehl [start-Transkription-job](#) und der Parameter `IdentifyLanguage` verwendet. Weitere Informationen erhalten Sie unter [StartTranscriptionJob](#) und [LanguageIdSettings](#).

Option 1: Ohne den Parameter `language-id-settings`. Verwenden Sie diese Option, wenn Sie kein benutzerdefiniertes Sprachmodell, kein benutzerdefiniertes Vokabular und keinen benutzerdefinierten Wortschatzfilter in Ihre Anfrage aufnehmen. `language-options` ist optional, wird aber empfohlen.

```
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"
```

Option 2: Mit dem Parameter `language-id-settings`. Verwenden Sie diese Option, wenn Sie ein benutzerdefiniertes Sprachmodell, ein benutzerdefiniertes Vokabular oder einen benutzerdefinierten Wortschatzfilter in Ihre Anfrage aufnehmen.

```
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
```

Hier ein weiteres Beispiel mit dem Befehl [start-Transkription-job](#) und einem Anforderungstext, der die Sprache angibt.

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

Die Datei `my-first-language-id-job.json` enthält den folgenden Anforderungstext.

Option 1: Ohne den Parameter `LanguageIdSettings`. Verwenden Sie diese Option, wenn Sie kein benutzerdefiniertes Sprachmodell, kein benutzerdefiniertes Vokabular und keinen benutzerdefinierten Wortschatzfilter in Ihre Anfrage aufnehmen. `LanguageOptions` ist optional, wird aber empfohlen.

```
{
  "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"
    ]
}

```

Option 2: Mit dem Parameter `LanguageIdSettings`. Verwenden Sie diese Option, wenn Sie ein benutzerdefiniertes Sprachmodell, ein benutzerdefiniertes Vokabular oder einen benutzerdefinierten Wortschatzfilter in Ihre Anfrage aufnehmen.

```

{
  "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"
    }
  }
}

```

AWS SDK für Python (Boto3)

In diesem Beispiel wird die AWS SDK für Python (Boto3) Sprache Ihrer Datei mithilfe des `IdentifyLanguage` Arguments für die Methode [start_transcription_job](#) identifiziert. Weitere Informationen erhalten Sie unter [StartTranscriptionJob](#) und [LanguageIdSettings](#).

Weitere Beispiele für die Verwendung der AWS SDKs, einschließlich funktionspezifischer, szenarienspezifischer und serviceübergreifender Beispiele, finden Sie im Kapitel. [Codebeispiele für Amazon Transcribe mit AWS SDKs](#)

Option 1: Ohne den Parameter `LanguageIdSettings`. Verwenden Sie diese Option, wenn Sie kein benutzerdefiniertes Sprachmodell, kein benutzerdefiniertes Vokabular und keinen benutzerdefinierten Wortschatzfilter in Ihre Anfrage aufnehmen. `LanguageOptions` ist optional, wird aber empfohlen.

```
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)
```

Option 2: Mit dem Parameter `LanguageIdSettings`. Verwenden Sie diese Option, wenn Sie ein benutzerdefiniertes Sprachmodell, ein benutzerdefiniertes Vokabular oder einen benutzerdefinierten Wortschatzfilter in Ihre Anfrage aufnehmen.

```
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)

```

Sprachidentifizierung mit Streaming-Transkriptionen

Durch die Identifizierung der Streaming-Sprache kann die dominante Sprache identifiziert werden, die in Ihrem Medienstream gesprochen wird. Amazon Transcribe benötigt mindestens eine Sekunde Sprachzeit, um die Sprache zu identifizieren.

Wenn Ihr Stream nur eine Sprache enthält, können Sie die einsprachige Identifizierung aktivieren, die die vorherrschende Sprache in Ihrer Mediendatei identifiziert und Ihr Transkript nur mit dieser Sprache erstellt.

Wenn Ihr Stream mehr als eine Sprache enthält, können Sie die Identifizierung mehrerer Sprachen aktivieren, die alle Sprachen in Ihrem Stream identifiziert und Ihr Transkript unter Verwendung jeder identifizierten Sprache erstellt. Beachten Sie, dass ein mehrsprachiges Transkript erstellt wird. Sie können andere Dienste verwenden, z. B. Amazon Transcribe um Ihr Transkript zu übersetzen.

Um die Streaming-Sprachidentifizierung zu verwenden, müssen Sie mindestens zwei Sprachcodes angeben, und Sie können nur einen Sprachdialekt pro Sprache und Stream auswählen. Das bedeutet, dass Sie nicht en-US und en-AU als Sprachoptionen für dieselbe Transkription auswählen können.

Sie haben auch die Möglichkeit, eine bevorzugte Sprache aus der Reihe der von Ihnen angegebenen Sprachcodes auszuwählen. Das Hinzufügen einer bevorzugten Sprache kann den Prozess der Sprachidentifizierung beschleunigen, was bei kurzen Audioclips hilfreich ist.

Important

Wenn keiner der von Ihnen angegebenen Sprachcodes mit der Sprache oder den Sprachen übereinstimmt, die in Ihrem Audio identifiziert wurden, wird aus den von Ihnen angegebenen Sprachcodes die Sprache Amazon Transcribe ausgewählt, die am ehesten entspricht. Es wird dann eine Transkription in dieser Sprache erstellt. Wenn Ihre Medien beispielsweise in US-Englisch (en-US) verfasst sind und Sie die Sprachcodes angeben zh-CN, Amazon Transcribe ist es wahrscheinlich fr-FR, dass Ihre Medien Amazon Transcribe mit Deutsch (de-DE) übereinstimmen und eine German-language Transkription erstellt wird. de-DE Wenn Sprachcodes und gesprochene Sprachen nicht übereinstimmen, kann dies zu einem ungenauen Transkript führen, weshalb wir bei der Angabe von Sprachcodes zur Vorsicht raten.

Wenn Ihre Medien zwei Kanäle enthalten, Amazon Transcribe können Sie die dominante Sprache ermitteln, die in jedem Kanal gesprochen wird. In diesem Fall setzen Sie [ChannelIdentification](#) auf `true` und jeder Kanal wird separat transkribiert. Beachten Sie, dass der Standardwert für diesen Parameter `false` ist. Wenn Sie die Einstellung nicht ändern, wird nur der erste Kanal transkribiert und nur eine Sprache erkannt.

Die Streaming-Sprachidentifizierung kann nicht mit benutzerdefinierten Sprachmodellen oder Schwärzen kombiniert werden. Wenn Sie die Sprachidentifizierung mit anderen Features kombinieren, sind Sie auf die Sprachen beschränkt, die von diesen Features unterstützt werden, und auch auf Streaming-Transkriptionen. Siehe [Unterstützte Sprachen](#).

 Note

PCM und FLAC sind die einzigen unterstützten Audioformate für die Streaming-Sprachidentifizierung. Für die mehrsprachige Identifizierung wird nur PCM unterstützt.

Identifizierung von Sprachen in mehrsprachigen Audiodateien

Multi-language Die Identifizierung ist für mehrsprachige Streams vorgesehen und bietet Ihnen ein Transkript, das alle unterstützten Sprachen widerspiegelt, die in Ihrem Stream gesprochen werden. Das bedeutet, dass Ihre Transkriptionsausgabe jede Sprache korrekt erkennt und transkribiert, wenn die Sprecher während des Gesprächs die Sprache wechseln oder wenn jeder Teilnehmer eine andere Sprache spricht.

Wenn in Ihrem Stream beispielsweise ein zweisprachiger Sprecher abwechselnd US-Englisch (en-US) und Hindi (hi-IN) spricht, kann die Identifizierung mehrerer Sprachen gesprochenes US-Englisch als en-US und gesprochenes Hindi als hi-IN identifizieren und transkribieren. Dies unterscheidet sich von der einsprachigen Identifizierung, bei der nur eine dominante Sprache zur Erstellung eines Transkripts verwendet wird. In diesem Fall wird jede gesprochene Sprache, die nicht die dominante Sprache ist, falsch transkribiert.

 Note

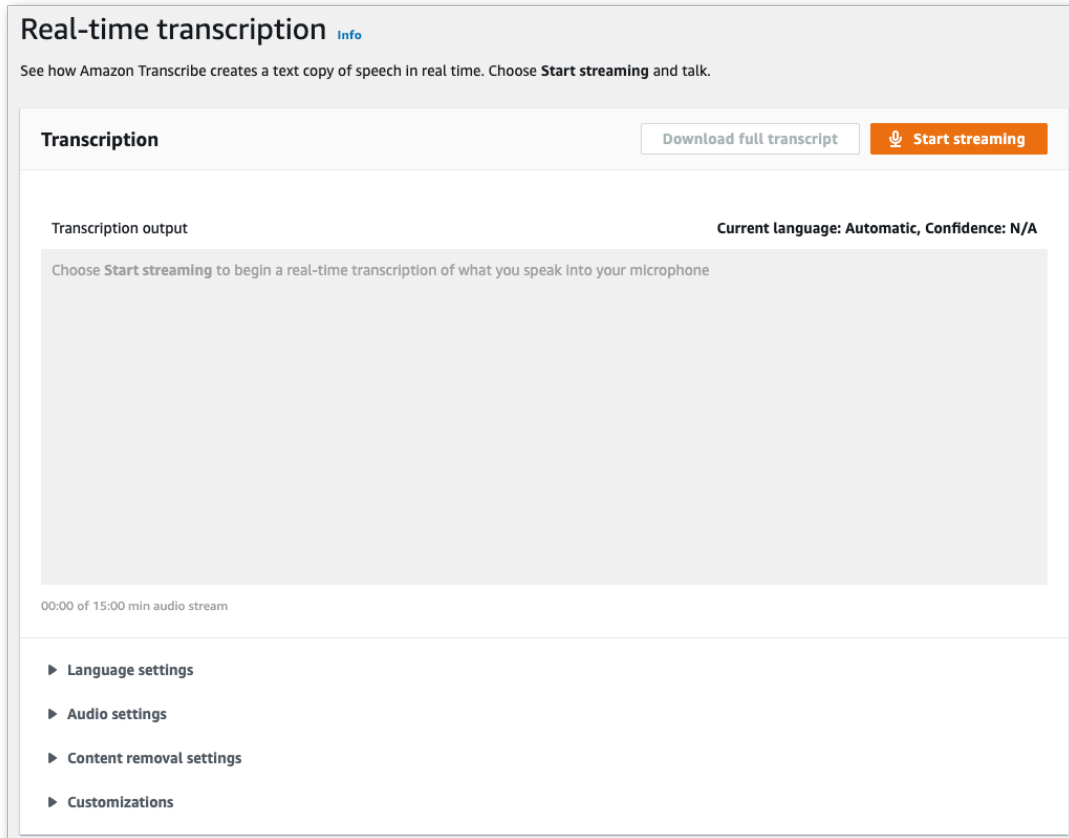
Schwärzen und benutzerdefinierte Sprachmodelle werden derzeit bei mehrsprachiger Identifizierung nicht unterstützt.

Sprachidentifizierung mit Streaming-Medien nutzen

Du kannst die automatische Sprachenerkennung in einer Streaming-Transkription mithilfe von AWS-Managementkonsole HTTP/2 WebSockets, oder verwenden. Im Folgenden findest du Beispiele:

AWS-Managementkonsole

1. Melden Sie sich an der [AWS-Managementkonsole](#) an.
2. Wählen Sie im Navigationsbereich Real-time Transkription aus. Blättern Sie nach unten zu den Spracheinstellungen und erweitern Sie dieses Feld, falls es minimiert ist.



3. Wählen Sie Automatische Sprachidentifikation oder Automatische Identifizierung mehrerer Sprachen aus.

▼ **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. Geben Sie mindestens zwei Sprachencodes für Ihre Transkription an. Beachten Sie, dass Sie nur einen Dialekt pro Sprache angeben können. Sie können zum Beispiel nicht sowohl en-US als auch en-GB als Sprachoptionen für dieselbe Transkription auswählen.

▼ **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. (Optional) Aus der Teilmenge der Sprachen, die Sie im vorherigen Schritt ausgewählt haben, können Sie eine bevorzugte Sprache für Ihr Transkript auswählen.

▼ 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. Jetzt können Sie Ihren Stream transkribieren. Wählen Sie Streaming starten und beginnen Sie zu sprechen. Um Ihr Diktat zu beenden, wählen Sie Streaming beenden.

HTTP/2 streamen

In diesem Beispiel wird eine HTTP/2 Anfrage mit aktivierter Sprachenidentifikation erstellt. Weitere Informationen zur Verwendung von HTTP/2 Streaming mit Amazon Transcribe finden Sie unter [Einen HTTP/2 Stream einrichten](#). Weitere Informationen zu spezifischen Parametern und Headern finden Sie Amazon Transcribe unter [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
```

In diesem Beispiel wird eine HTTP/2 Anfrage erstellt, bei der die Identifizierung mehrerer Sprachen aktiviert ist. Weitere Informationen zur Verwendung von HTTP/2 Streaming mit Amazon Transcribe finden Sie unter [Einen HTTP/2 Stream einrichten](#). Weitere Informationen zu spezifischen Parametern und Headern finden Sie Amazon Transcribe unter [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
```

Wenn Sie `identify-language` oder `identify-multiple-languages` in Ihrer Anforderung verwenden, müssen Sie auch `language-options` angeben. Sie können nicht sowohl `language-code` als auch `identify-language` in derselben Anfrage verwenden.

Parameterdefinitionen finden Sie in der [API-Referenz](#). Parameter, die allen AWS API-Vorgängen gemeinsam sind, sind im Abschnitt [Allgemeine Parameter](#) aufgeführt.

WebSocket streamen

In diesem Beispiel wird eine vorsignierte URL erstellt, die die Sprachenidentifikation in einem WebSocket Stream verwendet. Für eine bessere Lesbarkeit werden Zeilenumbrüche hinzugefügt. Weitere Informationen zur Verwendung von WebSocket Streams mit finden Sie Amazon Transcribe unter [Einen WebSocket Stream einrichten](#). Weitere Einzelheiten zu den Parametern finden Sie unter [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
```

In diesem Beispiel wird eine vorsignierte URL erstellt, die die Identifizierung mehrerer Sprachen in einem WebSocket Stream verwendet. Für eine bessere Lesbarkeit werden Zeilenumbrüche hinzugefügt. Weitere Informationen zur Verwendung von WebSocket Streams mit finden Sie Amazon Transcribe unter [Einen WebSocket Stream einrichten](#). Weitere Einzelheiten zu den Parametern finden Sie unter [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
```

Wenn Sie `identify-language` oder `identify-multiple-languages` in Ihrer Anforderung verwenden, müssen Sie auch `language-options` angeben. Sie können nicht sowohl `language-code` als auch `identify-language` in derselben Anfrage verwenden.

Parameterdefinitionen finden Sie in der [API-Referenz](#). Parameter, die allen AWS API-Vorgängen gemeinsam sind, sind im Abschnitt [Allgemeine Parameter](#) aufgeführt.

Alternative Transkriptionen

Bei der Amazon Transcribe Transkription von Audio werden verschiedene Versionen desselben Transkripts erstellt und jeder Version ein Konfidenzwert zugewiesen. Bei einer typischen Transkription erhalten Sie nur die Version mit dem höchsten Konfidenzwert.

Wenn Sie alternative Transkriptionen aktivieren, werden andere Versionen Ihres Transkripts Amazon Transcribe zurückgegeben, die ein niedrigeres Konfidenzniveau aufweisen. Sie können sich bis zu 10 alternative Transkriptionen zurückgeben lassen. Wenn Sie eine größere Anzahl von Alternativen angeben als angegeben, Amazon Transcribe wird nur die tatsächliche Anzahl von Alternativen zurückgegeben.

Alle Alternativen befinden sich in der gleichen Transkriptionsausgabedatei und werden auf Segmentebene dargestellt. Segmente sind natürliche Sprechpausen, z. B. wenn der Sprecher wechselt oder eine Pause im Ton gemacht wird.

Alternative Transkriptionen sind nur für Batch-Transkriptionen verfügbar.

Ihre Transkriptionsausgabe ist wie folgt aufgebaut. Die Auslassungspunkte (. . .) in den Codebeispielen geben der Kürze halber an, wo Inhalt entfernt wurde.

1. Eine vollständige abschließende Transkription für ein bestimmtes Segment.

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

2. Ein Konfidenzwert für jedes Wort im vorangegangenen transcript-Abschnitt .

```
"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. Ihre alternativen Transkriptionen befinden sich im `segments`-Teil Ihrer Transkriptionsausgabe. Die Alternativen für jedes Segment sind nach absteigendem Konfidenzwert geordnet.

```

"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. Ein Status am Ende der Transkriptionsausgabe.

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

Anforderung alternativer Transkriptionen

Sie können alternative Transkriptionen anfordern, indem Sie die SDKs AWS-Managementkonsole, AWS CLI oder AWS verwenden. Beispiele finden Sie im Folgenden:

AWS-Managementkonsole

1. Melden Sie sich an der [AWS-Managementkonsole](#) an.
2. Wählen Sie im Navigationsbereich Transkriptionsaufträge und dann Auftrag erstellen (oben rechts). Dies öffnet die Seite Auftragsdetails an.

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. Füllen Sie alle Felder aus, die Sie auf der Seite Auftragsdetails angeben möchten, und wählen Sie dann Weiter. Dadurch gelangen Sie zur Seite Auftrag konfigurieren – optional.

Wählen Sie Alternative Ergebnisse und geben Sie die maximale Anzahl der alternativen Transkriptionsergebnisse an, die Sie in Ihrem Transkript haben möchten.

Configure job - optional [Info](#)

Audio settings

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

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

Maximum alternatives
Provide the number of alternative transcription to provide in the text output.

The maximum number of alternative results is 10.

4. Wählen Sie Auftrag erstellen, um Ihren Transkriptionsauftrag auszuführen.

AWS CLI

In diesem Beispiel werden der Befehl [start-Transkription-job](#) und der Parameter ShowAlternatives verwendet. Weitere Informationen erhalten Sie unter [StartTranscriptionJob](#) und [ShowAlternatives](#).

Beachten Sie, dass Sie, wenn Sie ShowAlternatives=true in Ihrer Anfrage angeben, auch MaxAlternatives angeben müssen.

```
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
```

Hier ein weiteres Beispiel mit dem Befehl [start-Transkription-job](#) und einem Anforderungstext, der alternative Transkriptionen enthält.

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

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

Die Datei my-first-alt-Transkription-job.json enthält den folgenden Anforderungstext.

```
{
  "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
  }
}
```

AWS SDK für Python (Boto3)

[Im folgenden Beispiel werden alternative Transkriptionen mithilfe des AWS SDK für Python \(Boto3\) `ShowAlternatives` Arguments für die Methode `start\_transcription\_job` angefordert.](#) Weitere Informationen erhalten Sie unter [StartTranscriptionJob](#) und [ShowAlternatives](#).

Weitere Beispiele für die Verwendung der AWS SDKs, einschließlich funktionspezifischer, szenarienspezifischer und serviceübergreifender Beispiele, finden Sie im Kapitel. [Codebeispiele für Amazon Transcribe mit AWS SDKs](#)

Beachten Sie, dass Sie, wenn Sie 'ShowAlternatives':True in Ihrer Anfrage angeben, auch MaxAlternatives angeben müssen.

```
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)
```


Verbesserung der Transkriptionsgenauigkeit mit benutzerdefinierten Vokabularen und benutzerdefinierten Sprachmodellen

Wenn Ihre Medien domänenspezifische oder nicht standardmäßige Begriffe wie Markennamen, Akronyme, technische Wörter und Fachjargon enthalten, werden diese Begriffe Amazon Transcribe möglicherweise nicht korrekt in Ihrer Transkriptionsausgabe erfasst.

Um Ungenauigkeiten bei der Transkription zu korrigieren und die Ausgabe für Ihren speziellen Anwendungsfall anzupassen, können Sie [Benutzerdefinierte Vokabulare](#) und [Benutzerdefinierte Sprachmodelle](#) erstellen.

- [Benutzerdefinierte Vokabulare](#) sind so konzipiert, dass sie sowohl die Erkennung als auch die Formatierung bestimmter Wörter in allen Kontexten optimieren und verbessern. Dazu wird Amazon Transcribe mit Wörtern und optional mit Aussprache- und Darstellungsformen versorgt.

Wenn bestimmte Begriffe in Ihren Transkripten nicht Amazon Transcribe korrekt wiedergegeben werden, können Sie eine benutzerdefinierte Vokabeldatei erstellen, die angibt, wie diese Begriffe angezeigt werden sollen. Amazon Transcribe Dieser wortspezifische Ansatz eignet sich am besten für die Korrektur von Begriffen wie Markennamen und Akronymen.

- [Benutzerdefinierte Sprachmodelle](#) sind so konzipiert, dass sie den mit den Begriffen verbundenen Kontext erfassen. Dies beinhaltet die Bereitstellung Amazon Transcribe einer großen Menge domänenspezifischer Textdaten.

Wenn technische Begriffe nicht Amazon Transcribe korrekt wiedergegeben werden oder in Ihren Transkripten das falsche Homophon verwendet wird, können Sie ein benutzerdefiniertes Sprachmodell erstellen, das Ihre domänenspezifische Sprache vermittelt Amazon Transcribe . Ein benutzerdefiniertes Sprachmodell kann zum Beispiel lernen, wann „floe“ (Eisscholle) und wann „flow“ (linearer Fluss) verwendet werden soll.

Dieser kontextbewusste Ansatz eignet sich am besten für die Transkription großer Mengen an domainspezifischer Sprache. Benutzerdefinierte Sprachmodelle können die Genauigkeit gegenüber benutzerdefinierten Vokabularen allein erheblich verbessern. Bei der Verwendung von Batch-Transkriptionen können Sie sowohl ein benutzerdefiniertes Sprachmodell als auch ein benutzerdefiniertes Vokabular in Ihre Anfrage aufnehmen.

 Tip

Um die höchste Transkriptionsgenauigkeit zu erreichen, verwenden Sie benutzerdefinierte Vokabulare in Verbindung mit Ihren benutzerdefinierten Sprachmodellen.

[Eine Videodemo zur Erstellung eines benutzerdefinierten Vokabulars mit dem finden Sie unter Verwenden eines benutzerdefinierten Wortschatzes. AWS-Managementkonsole](#)

Eine Video-Demo zur Erstellung und Verwendung benutzerdefinierter Sprachmodelle finden Sie unter [Verwendung benutzerdefinierter Sprachmodelle \(CLM\) zur Verbesserung der Transkriptionsgenauigkeit](#).

 Tauchen Sie tiefer ein mit dem AWS Blog zum Machine Learning

Benutzerdefinierte Vokabulare:

- [Live-Transkriptionen von F1-Rennen mit Amazon Transcribe](#)

Benutzerdefinierte Sprachmodelle:

- [Erstellung benutzerdefinierter Sprachmodelle zur Steigerung der Sprache-zu-Text-Leistung Amazon Transcribe](#)
- [Erhöhen Sie die Transkriptionsgenauigkeit von Vorlesungen mit benutzerdefinierten Sprachmodellen für Amazon Transcribe](#)

Benutzerdefinierte Vokabulare

Verwenden Sie benutzerdefinierte Vokabulare, um die Transkriptionsgenauigkeit für ein bestimmtes Wort oder mehrere bestimmte Wörter zu verbessern. Dabei handelt es sich in der Regel um bereichsspezifische Begriffe wie Markennamen und Akronyme, Eigennamen und Wörter, die von Amazon Transcribe nicht korrekt wiedergegeben werden.

Benutzerdefinierte Vokabulare können mit allen unterstützten Sprachen verwendet werden. Beachten Sie, dass nur die im [Zeichensatz](#) Ihrer Sprache aufgeführten Zeichen in einem benutzerdefinierten Vokabular verwendet werden können.

 Important

Sie sind für die Integrität Ihrer eigenen Daten verantwortlich, wenn Sie Amazon Transcribe verwenden. Geben Sie keine vertraulichen Daten, personenbezogene Daten (PII) oder geschützte Gesundheitsinformationen (PHI) in ein benutzerdefiniertes Vokabular ein.

Überlegungen bei der Erstellung eines benutzerdefinierten Vokabulars:

- Sie können bis zu 100 benutzerdefinierte Vokabeldateien pro Datei haben AWS-Konto
- Die Größe jeder benutzerdefinierten Vokabulardatei ist auf 50 Kb begrenzt
- Wenn Sie Ihr benutzerdefiniertes Vokabular über die API erstellen, muss Ihre Vokabulardatei im Textformat (*.txt) vorliegen. Wenn Sie das verwenden AWS-Managementkonsole, kann Ihre Vokabeldatei im Textformat (*.txt) oder im Format mit kommagetrennten Werten (*.csv) vorliegen.
- Jeder Eintrag innerhalb eines benutzerdefinierten Vokabulars darf nicht länger als 256 Zeichen sein
- Um ein benutzerdefiniertes Vokabular verwenden zu können, muss es in derselben AWS-Region Weise wie Ihre Transkription erstellt worden sein.

 Tip

Sie können Ihr benutzerdefiniertes Vokabular mit dem AWS-Managementkonsole testen. Sobald Ihr benutzerdefiniertes Vokabular einsatzbereit ist, melden Sie sich bei an, wählen Sie Real-timeTranskription aus AWS-Managementkonsole, scrollen Sie zu Anpassungen, schalten Sie Benutzerdefiniertes Vokabular ein und wählen Sie Ihr benutzerdefiniertes Vokabular aus der Drop-down-Liste aus. Wählen Sie dann Streaming starten. Sprechen Sie einige der Wörter aus Ihrem benutzerdefinierten Vokabular in Ihr Mikrofon, um zu sehen, ob sie korrekt wiedergegeben werden.

Benutzerdefinierte Vokabulartabellen versus Listen

 Important

Benutzerdefinierte Vokabulare im Listenformat werden nicht mehr verwendet. Wenn Sie ein neues benutzerdefiniertes Vokabular erstellen, verwenden Sie das [Tabellenformat](#).

Tabellen bieten Ihnen mehr Optionen für die Eingabe und Ausgabe von Wörtern innerhalb Ihres benutzerdefinierten Vokabulars – und mehr Kontrolle darüber. Bei Tabellen müssen Sie mehrere Kategorien angeben (Phrase and DisplayAs), sodass Sie Ihre Ausgabe feiner abstimmen können.

Listen haben keine zusätzlichen Optionen, sodass Sie nur Einträge so eingeben können, wie sie in Ihrer Transkript erscheinen sollen, wobei alle Leerzeichen durch Bindestriche ersetzt werden.

Die AWS SDKs, AWS-Managementkonsole, AWS CLI, und verwenden benutzerdefinierte Vokabeltabellen auf die gleiche Weise. Listen werden für jede Methode unterschiedlich verwendet und erfordern daher möglicherweise zusätzliche Formatierungen, um erfolgreich zwischen den Methoden verwendet werden zu können.

Weitere Informationen erhalten Sie unter [Erstellen eines benutzerdefinierten Vokabulars mithilfe einer Tabelle](#) und [Erstellen eines benutzerdefinierten Vokabulars mithilfe einer Liste](#).

Wenn Sie etwas tiefer eintauchen und erfahren möchten, wie Sie Amazon Augmented AI mit benutzerdefinierten Vokabularen verwenden können, lesen Sie den Abschnitt [Erstellen einer menschlichen Bewertung zusammen mit Amazon Transcribe](#).

 API-Vorgänge speziell für benutzerdefinierte Vokabulare
[CreateVocabulary](#), [DeleteVocabulary](#), [GetVocabulary](#), [ListVocabularies](#),
[UpdateVocabulary](#)

Erstellen eines benutzerdefinierten Vokabulars mithilfe einer Tabelle

Ein Tabellenformat ist die bevorzugte Methode zur Erstellung Ihres benutzerdefinierten Vokabulars. Vokabulartabellen müssen aus vier Spalten (Phrase, SoundsLike, IPA, and DisplayAs) bestehen, die in beliebiger Reihenfolge angeordnet werden können:

Phrase	SoundsLike	IPA	DisplayAs
Erforderlich Jede Zeile in Ihrer Tabelle muss einen Eintrag in dieser Spalte enthalten.	SoundsLike wird für Custom Vocabulary nicht mehr unterstützt. Bitte lassen Sie die Spalte leer. Alle Werte in dieser Spalte	IPAwird für Custom Vocabulary nicht mehr unterstützt. Bitte lassen Sie die Spalte leer. Alle Werte in dieser Spalte werden	Optional. Zeilen in dieser Spalte können leer gelassen werden.

Phrase	SoundsLike	IPA	DisplayAs
Verwenden Sie keine Leerzeichen in dieser Spalte.	werden ignoriert. Wir werden die Unterstützung für diese Spalte in future entfernen.	ignoriert. Wir werden die Unterstützung für diese Spalte in future entfernen.	Sie können in dieser Spalte Leerzeichen verwenden.
Wenn Ihr Eintrag mehrere Wörter enthält, trennen Sie jedes Wort mit einem Bindestrich (-). Zum Beispiel Andorra-la-Vella oder Los-Angeles .			Legt fest, wie Ihr Eintrag in Ihrer Transkriptionsausgabe aussehen soll. Zum Beispiel ist Andorra-la-Vella in der Spalte Phrase gleich Andorra la Vella in der Spalte DisplayAs .
Bei Akronymen müssen alle ausgesprochenen Buchstaben durch einen Punkt getrennt werden. Auch der abschließende Punkt muss ausgesprochen werden. Wenn Ihr Akronym im Plural steht, müssen Sie einen Bindestrich zwischen dem Akronym und dem „s“ verwenden. Zum Beispiel ist „CLI“ C.L.I. (nicht C.L.I) und „ABCs“ ist A.B.C.-s (nicht A.B.C-s).			Wenn eine Zeile in dieser Spalte leer ist, Amazon Transcribe verwendet es den Inhalt der Phrase Spalte, um die Ausgabe zu bestimmen.
Wenn Ihr Satz aus einem Wort und			Sie können in dieser Spalte Ziffern (0-9) verwenden.

Phrase	SoundsLike	IPA	DisplayAs
einem Akronym besteht, müssen diese beiden Bestandteile durch einen Bindestrich getrennt werden. Zum Beispiel ist „DynamoDB“ Dynamo-D.B. . In dieser Spalte dürfen keine Ziffern stehen; die Zahlen müssen ausgesprochen werden. Zum Beispiel ist „VX02Q“ V.X. -zero-two-Q.			

Das sollten Sie bei der Erstellung Ihrer Tabelle beachten:

- Ihre Tabelle muss alle vier Spaltenüberschriften (Phrase, SoundsLike, IPA, and DisplayAs) enthalten. Die Phrase Spalte muss in jeder Zeile einen Eintrag enthalten. Die Möglichkeit, Ausspracheeingaben über IPA und SoundsLike einzugeben, wird nicht mehr unterstützt, und Sie können die Spalte leer lassen. Alle Werte in diesen Spalten werden ignoriert.
- Jede Spalte muss mit TAB oder Komma (,) getrennt sein. Dies gilt für jede Zeile in Ihrer benutzerdefinierten Vokabulardatei. Wenn eine Zeile leere Spalten enthält, müssen Sie trotzdem ein Trennzeichen (TAB oder Komma) für jede Spalte angeben.
- Leerzeichen sind nur innerhalb der Spalten IPA und DisplayAs erlaubt. Verwenden Sie keine Leerzeichen zur Trennung von Spalten.
- IPA und SoundsLike werden für Custom Vocabulary nicht mehr unterstützt. Bitte lassen Sie die Spalte leer. Alle Werte in dieser Spalte werden ignoriert. Wir werden die Unterstützung für diese Spalte in future entfernen.

- Die Spalte `DisplayAs` unterstützt Symbole und Sonderzeichen (z. B. C++). Alle anderen Spalten unterstützen die Zeichen, die auf der Seite mit dem [Zeichensatz](#) Ihrer Sprache aufgeführt sind.
- Die Phrase Spalte hat die folgenden Formatierungsregeln:
 - Kann nicht mit einem Punkt (.), Apostroph (') oder Bindestrich () beginnen. - Zum Beispiel - **hello** und **.test** sind ungültig.
 - Kann nicht mit einem Apostroph (') oder einem Bindestrich () enden. - Zum Beispiel **hello-** und **world'** sind ungültig.
 - Darf keine wiederholten Bindestriche (--), wiederholten Punkte (..) oder wiederholten Apostrophe () enthalten. ' ' Zum Beispiel und sind ungültig. **well--known can''t**
 - Punkte können nur zur Bezeichnung von Akronymen verwendet werden (einzelne Buchstaben, die durch Punkte getrennt sind, z. B.). **A.B.C.** Ein Punkt darf nicht hinter zwei oder mehr aufeinanderfolgenden Nicht-Sonderzeichen stehen (z. B. **AB.C** ist ungültig) und es können auch nicht drei oder mehr aufeinanderfolgende Nicht-Sonderzeichen folgen (z. B. ist ungültig). **A.BCD**
 - Es werden nur Zeichen unterstützt, die im [Zeichensatz](#) Ihrer Sprache aufgeführt sind.
- Wenn Sie in der Spalte `Phrase` Zahlen angeben möchten, müssen Sie diese buchstabieren. Ziffern (0-9) werden nur in der Spalte `DisplayAs` unterstützt.
- Sie müssen Ihre Tabelle als Klartextdatei (*.txt) speichern. LF Sowohl CRLF Zeilenenden als auch Zeilenenden werden unterstützt.
- Sie müssen Ihre benutzerdefinierte Vokabeldatei in einen Amazon S3 Bucket hochladen und sie mithilfe dieser Daten verarbeiten, [CreateVocabulary](#) bevor Sie sie in eine Transkriptionsanfrage aufnehmen können. Anweisungen finden Sie unter [Erstellen von benutzerdefinierten Vokabulartabellen](#).

Note

Geben Sie Akronyme oder andere Wörter, deren Buchstaben einzeln ausgesprochen werden müssen, als einzelne, durch Punkte getrennte Buchstaben ein (**A.B.C.**). Um die Pluralform eines Akronyms einzugeben, wie z. B. „ABCs“, trennen Sie das „s“ vom Akronym mit einem Bindestrich (**A.B.C. -s**). Sie können Groß- oder Kleinbuchstaben verwenden, um ein Akronym zu definieren. Akronyme werden nicht in allen Sprachen unterstützt; siehe [Unterstützte Sprachen und sprachspezifische Funktionen](#).

Hier ist eine Beispieltabelle für ein benutzerdefiniertes Vokabular (wobei **[TAB]** ein Tabulatorzeichen darstellt):

```
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
```

Zur Verdeutlichung sehen Sie hier die gleiche Tabelle mit ausgerichteten Spalten. Fügen Sie in Ihrer benutzerdefinierten Vokabulartabelle keine Leerzeichen zwischen den Spalten ein; Ihre Tabelle sollte wie im vorangegangenen Beispiel schief aussehen.

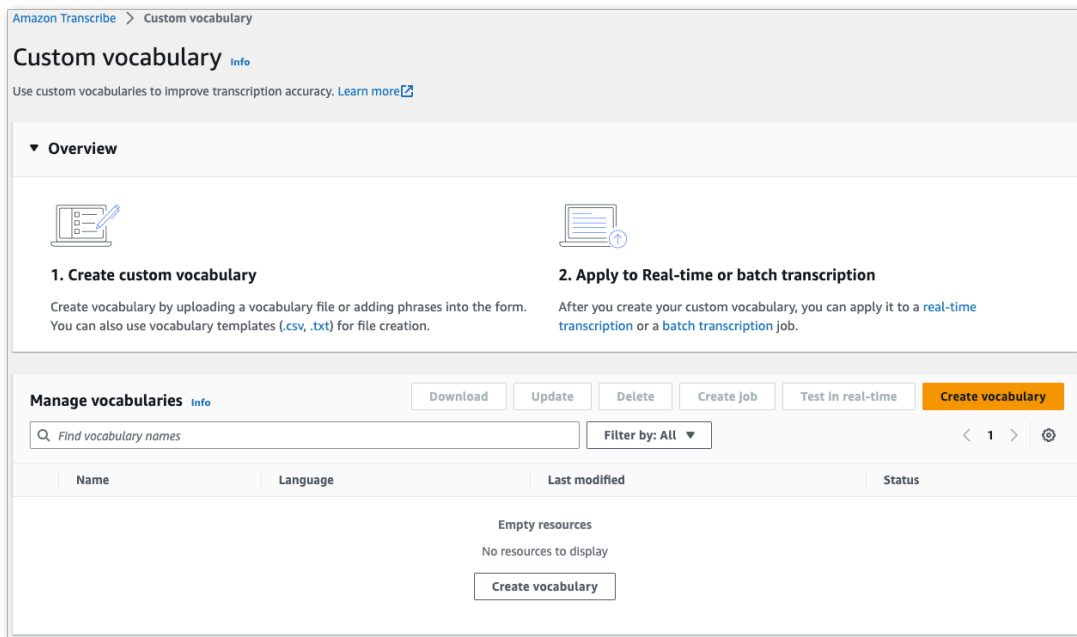
Phrase	[TAB]SoundsLike	[TAB]IPA	[TAB]DisplayAs
Los-Angeles	[TAB]	[TAB]	[TAB]Los Angeles
Eva-Maria	[TAB]	[TAB]	[TAB]
A.B.C.-s	[TAB]	[TAB]	[TAB]ABCs
amazon-dot-com	[TAB]	[TAB]	[TAB]amazon.com
C.L.I.	[TAB]	[TAB]	[TAB]CLI
Andorra-la-Vella	[TAB]	[TAB]	[TAB]Andorra la Vella
Dynamo-D.B.	[TAB]	[TAB]	[TAB]DynamoDB
V.X.-zero-two	[TAB]	[TAB]	[TAB]VX02
V.X.-zero-two-Q.	[TAB]	[TAB]	[TAB]VX02Q

Erstellen von benutzerdefinierten Vokabulartabellen

In den folgenden Beispielen erfahren Sie Amazon Transcribe, wie Sie eine benutzerdefinierte Vokabeltabelle für die Verwendung mit bearbeiten können:

AWS-Managementkonsole

1. Melden Sie sich an der [AWS-Managementkonsole](#) an.
2. Wählen Sie im Navigationsbereich die Option Benutzerdefiniertes Vokabular. Dies öffnet die Seite Benutzerdefiniertes Vokabular, auf der Sie vorhandene Vokabulare ansehen oder ein neues erstellen können.
3. Wählen Sie Vokabular erstellen.



Sie gelangen auf die Seite Vokabular erstellen. Geben Sie einen Namen für Ihr neues benutzerdefiniertes Vokabular ein.

Hier haben Sie drei Möglichkeiten:

- Laden Sie eine txt- oder csv-Datei von Ihrem Computer hoch.

Sie können entweder Ihr benutzerdefiniertes Vokabular von Grund auf neu erstellen oder eine Vorlage herunterladen, die Ihnen den Einstieg erleichtert. Ihr Vokabular wird dann automatisch im Bereich Vokabular anzeigen und bearbeiten ausgefüllt.

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

English, US (en-US) ▼

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 file

Choose File

File format: txt, csv, maximum size 50 KB.

- b. Importiert eine TXT- oder CSV-Datei von einem beliebigen Ort. Amazon S3

Sie können entweder Ihr benutzerdefiniertes Vokabular von Grund auf neu erstellen oder eine Vorlage herunterladen, die Ihnen den Einstieg erleichtert. Laden Sie Ihre fertige Vokabulardatei in einen Amazon S3 -Bucket hoch und geben Sie dessen URI in Ihrer Anfrage an. Ihr Vokabular wird dann automatisch im Bereich Vokabular anzeigen und bearbeiten ausgefüllt.

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. Erstellen Sie Ihr Vokabular manuell in der Konsole.

Blättern Sie zum Bereich Ansicht und Vokabular bearbeiten und wählen Sie 10 Zeilen hinzufügen. Sie können nun Begriffe manuell eingeben.

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. Sie können Ihr Vokabular im Bereich Ansicht und Vokabularbearbeitung bearbeiten. Um Änderungen vorzunehmen, klicken Sie auf den Eintrag, den Sie ändern möchten.

View and edit vocabulary - new (10) [Info](#) Reset vocabulary Delete Download latest ▼

Show all ▼ < 1 >

☐	Phrase ✎ ▼	SoundsLike - optional ✎ ▼	IPA - optional ✎ ▼	DisplayAs - optional ✎ ▼
☐	Amazon-E.-C.-two	-	-	Amazon EC2
☐	Amazon-S.-three	-	-	Amazon S3
☐	Amazon-elasticashe	-	-	Amazon ElastiCache
☐	Amazon-sagemaker	-	-	Amazon SageMaker
☐	A.-W.-S.-iam	-	-	AWS IAM
☐	A.-W.-S.-I.-o.-T.	-	-	AWS IoT
☐	A.-W.-S.-W.-A.-F.	-	-	AWS WAF
☐	c.-plus-plus	-	-	C++
☐	nice-d.-c.-v.	-	-	NICE DCV
☐	w.-w.-w.-dot-amazon-dot-com	-	-	www.amazon.com

Add row

Wenn Ihnen ein Fehler unterläuft, erhalten Sie eine detaillierte Fehlermeldung, sodass Sie eventuelle Probleme vor der Bearbeitung Ihres Vokabulars beheben können. Beachten Sie, dass Ihre Vokabularanfrage fehlschlägt, wenn Sie nicht alle Fehler korrigieren, bevor Sie Vokabular erstellen auswählen.

View and edit vocabulary - new (4) [Info](#) Reset vocabulary Delete Download latest ▼

Show all ▼ < 1 >

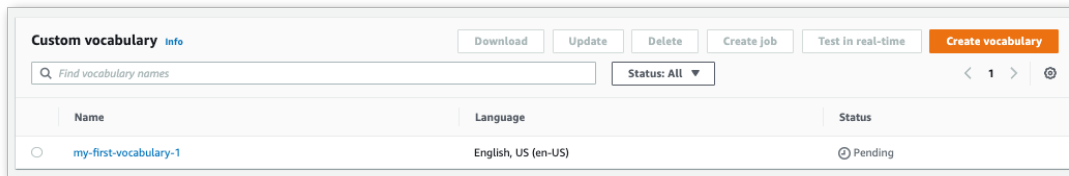
☐	Phrase ✎ ▼	SoundsLike - optional ✎ ▼	IPA - optional ✎ ▼	DisplayAs - optional ✎ ▼
☐	Amazon-E.-C. two ✕ ✓ ⚠ Phrase contains unsupported characters (" "). Phrase contains a formatting error.	-	-	Amazon EC2
☐	Amazon-S.-three	-	-	Amazon S3
☐	c.-plus-plus	-	-	C++
☐	w.-w.-w.-dot-amazon-dot-com	-	-	www.amazon.com

Add row

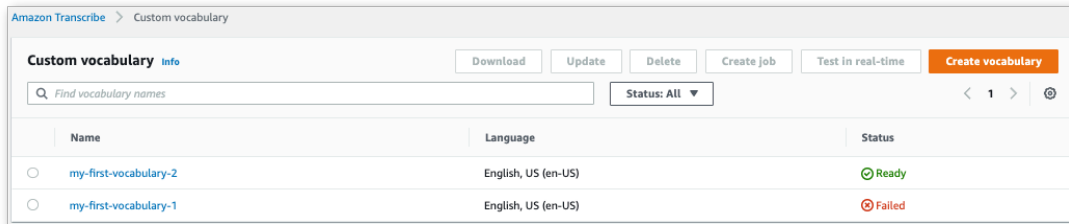
Wählen Sie das Häkchen (✓), um Ihre Änderungen zu speichern, oder das „X“, um Ihre Änderungen zu verwerfen.

- Optional können Sie Ihrem benutzerdefinierten Vokabular Tags hinzufügen. Wenn Sie alle Felder ausgefüllt haben und mit Ihrem Vokabular zufrieden sind, wählen Sie unten auf der Seite Vokabular erstellen. Dies bringt Sie zurück zur Seite Benutzerdefiniertes Vokabular, wo Sie den Status Ihres benutzerdefinierten Vokabulars einsehen können. Wenn der Status

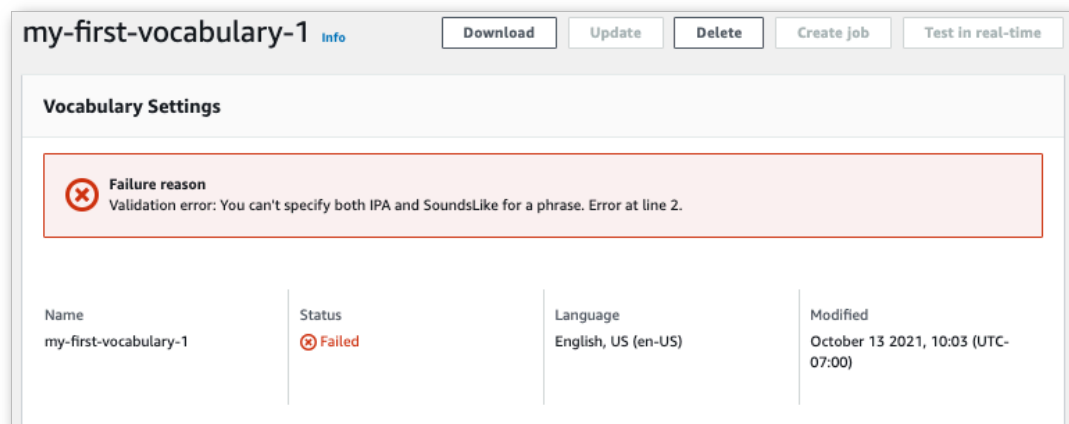
von „Ausstehend“ auf „Bereit“ wechselt, kann Ihr benutzerdefiniertes Vokabular mit einer Transkription verwendet werden.



6. Wenn sich der Status in „Fehlgeschlagen“ ändert, wählen Sie den Namen Ihres benutzerdefinierten Vokabulars aus, um zu dessen Informationsseite zu gelangen.



Oben auf dieser Seite befindet sich ein Banner mit dem Grund für das Scheitern, das Auskunft darüber gibt, warum Ihr benutzerdefiniertes Vokabular nicht funktioniert hat. Korrigieren Sie den Fehler in Ihrer Textdatei und versuchen Sie es erneut.



AWS CLI

In diesem Beispiel wird der Befehl [create-vocabulary](#) mit einer tabellenformatierten Vokabulardatei verwendet. Weitere Informationen finden Sie unter [CreateVocabulary](#).

Um ein vorhandenes benutzerdefiniertes Vokabular in einem Transkriptionsauftrag zu verwenden, geben Sie das `VocabularyName` in das [Settings](#)-Feld ein, wenn Sie den

[StartTranscriptionJob](#)Vorgang aufrufen, oder wählen Sie das AWS-Managementkonsole benutzerdefinierte Vokabular aus der Dropdownliste aus.

```
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
```

Hier ein weiteres Beispiel mit dem Befehl [create-vocabulary](#) und einem Anforderungstext, der Ihr benutzerdefiniertes Vokabular erstellt.

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

Die Datei *my-first-vocab-table.json* enthält den folgenden Anforderungstext.

```
{  
  "VocabularyName": "my-first-vocabulary",  
  "VocabularyFileUri": "s3://amzn-s3-demo-bucket/my-vocabularies/my-vocabulary-table.txt",  
  "LanguageCode": "en-US"  
}
```

Sobald VocabularyState von PENDING auf READYwechselt, ist Ihr benutzerdefiniertes Vokabular bereit für die Verwendung mit einer Transkription. Um den aktuellen Status Ihres benutzerdefinierten Vokabulars anzuzeigen, führen Sie Folgendes aus:

```
aws transcribe get-vocabulary \  
--vocabulary-name my-first-vocabulary
```

AWS SDK für Python (Boto3)

In diesem Beispiel wird mithilfe der AWS SDK für Python (Boto3) Methode [create_vocabulary](#) ein benutzerdefiniertes Vokabular aus einer Tabelle erstellt. Weitere Informationen finden Sie unter [CreateVocabulary](#).

Um ein vorhandenes benutzerdefiniertes Vokabular in einem Transkriptionsauftrag zu verwenden, legen Sie das VocabularyName in dem [Settings](#)Feld fest, wenn Sie den [StartTranscriptionJob](#)Vorgang aufrufen, oder wählen Sie das AWS-Managementkonsole benutzerdefinierte Vokabular aus der Dropdownliste aus.

Weitere Beispiele für die Verwendung der AWS SDKs, einschließlich funktionspezifischer, szenarienspezifischer und serviceübergreifender Beispiele, finden Sie im Kapitel. [Codebeispiele für Amazon Transcribe mit AWS SDKs](#)

```
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

Wenn Sie einen neuen Amazon S3 Bucket für Ihre benutzerdefinierten Vokabeldateien erstellen, stellen Sie sicher, dass die IAM Rolle, die die [CreateVocabulary](#)Anfrage stellt, über Zugriffsberechtigungen für diesen Bucket verfügt. Wenn die Rolle nicht über die richtigen Berechtigungen verfügt, schlägt Ihre Anfrage fehl. Sie können optional eine IAM Rolle in Ihrer Anfrage angeben, indem Sie den `DataAccessRoleArn` Parameter angeben. Weitere Informationen zu IAM Rollen und Richtlinien finden Sie unter [Amazon Transcribe Beispiele für identitätsbasierte Politik](#). Amazon Transcribe

Erstellen eines benutzerdefinierten Vokabulars mithilfe einer Liste

Important

Benutzerdefinierte Vokabulare im Listenformat sind veraltet. Wenn Sie also ein neues benutzerdefiniertes Vokabular erstellen, empfehlen wir Ihnen dringend, das [Tabellenformat](#) zu verwenden.

Mithilfe der SDKs, oder AWS können Sie benutzerdefinierte Vokabulare aus Listen erstellen. AWS-Managementkonsole AWS CLI

- AWS-Managementkonsole: Sie müssen eine Textdatei mit Ihrem individuellen Vokabular erstellen und hochladen. Sie können zeilengetrennte oder kommasetrennte Einträge verwenden. Beachten Sie, dass Ihre Liste als Textdatei (*.txt) gespeichert werden muss. LFSowohl CRLF Zeilenenden als auch Zeilenenden werden unterstützt.
- SDKs AWS CLI und AWS : Sie müssen Ihr benutzerdefiniertes Vokabular als kommasetrennte Einträge in Ihren API-Aufruf einfügen, indem Sie die [Phrases](#)-Markierung nutzen.

Wenn ein Eintrag mehrere Wörter enthält, müssen Sie jedes Wort mit einem Bindestrich trennen. Sie nehmen zum Beispiel „Los Angeles“ als **Los-Angeles** und „Andorra la Vella“ als **Andorra-la-Vella** auf.

Hier sind Beispiele für die beiden gültigen Listenformate. Siehe [Benutzerdefinierte Vokabularlisten erstellen](#) für methodenspezifische Beispiele.

- Comma-separated Einträge:

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

- Line-separated Einträge:

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


⚠ Important

Sie können nur Zeichen verwenden, die für Ihre Sprache unterstützt werden. Einzelheiten dazu finden Sie im [Zeichensatz](#) Ihrer Sprache.

Benutzerdefinierte Vokabularlisten werden bei dem [CreateMedicalVocabulary](#)-Vorgang nicht unterstützt. Wenn Sie ein benutzerdefiniertes medizinisches Vokabular erstellen, müssen Sie ein Tabellenformat verwenden. Anweisungen finden Sie unter [Erstellen eines benutzerdefinierten Vokabulars mithilfe einer Tabelle](#).

Benutzerdefinierte Vokabularlisten erstellen

In den folgenden Beispielen erfahren Sie Amazon Transcribe, wie Sie eine benutzerdefinierte Vokabelliste für die Verwendung mit bearbeiten können:

AWS CLI

In diesem Beispiel wird der Befehl [create-vocabulary](#) mit einer als Liste formatierten benutzerdefinierten Vokabulardatei verwendet. Weitere Informationen finden Sie unter [CreateVocabulary](#).

```
aws transcribe create-vocabulary \  
--vocabulary-name my-first-vocabulary \  
--language-code en-US \  
--phrases {CLI,Eva-Maria,ABCs}
```

Hier ein weiteres Beispiel mit dem Befehl [create-vocabulary](#) und einem Anforderungstext, der Ihr benutzerdefiniertes Vokabular erstellt.

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

Die Datei `my-first-vocab-list.json` enthält den folgenden Anforderungstext.

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

```

        "CLI", "Eva-Maria", "ABCs"
    ]
}
```

Sobald `VocabularyState` von `PENDING` auf `READY` wechselt, ist Ihr benutzerdefiniertes Vokabular bereit für die Verwendung mit einer Transkription. Um den aktuellen Status Ihres benutzerdefinierten Vokabulars anzuzeigen, führen Sie Folgendes aus:

```
aws transcribe get-vocabulary \
--vocabulary-name my-first-vocabulary
```

AWS SDK für Python (Boto3)

In diesem Beispiel wird mithilfe der AWS SDK für Python (Boto3) Methode [create_vocabulary](#) ein benutzerdefiniertes Vokabular aus einer Liste erstellt. Weitere Informationen finden Sie unter [CreateVocabulary](#).

Weitere Beispiele für die Verwendung der AWS SDKs, einschließlich funktionspezifischer, szenarienspezifischer und serviceübergreifender Beispiele, finden Sie im Kapitel. [Codebeispiele für Amazon Transcribe mit AWS SDKs](#)

```

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

Wenn Sie einen neuen Amazon S3 Bucket für Ihre benutzerdefinierten Vokabeldateien erstellen, stellen Sie sicher, dass die IAM Rolle, die die [CreateVocabulary](#)Anfrage stellt, über Zugriffsberechtigungen für diesen Bucket verfügt. Wenn die Rolle nicht über die richtigen Berechtigungen verfügt, schlägt Ihre Anfrage fehl. Sie können optional eine IAM Rolle in Ihrer Anfrage angeben, indem Sie den `DataAccessRoleArn` Parameter angeben. Weitere Informationen zu IAM Rollen und Richtlinien finden Sie unter [Amazon Transcribe Beispiele für identitätsbasierte Politik](#). Amazon Transcribe

Verwenden eines benutzerdefinierten Vokabulars

Sobald Sie Ihr benutzerdefiniertes Vokabular erstellt haben, können Sie es in Ihre Transkriptionsanfragen aufnehmen; Beispiele finden Sie in den folgenden Abschnitten.

Die Sprache des benutzerdefinierten Vokabulars, das Sie in Ihre Anfrage aufnehmen, muss dem Sprachcode entsprechen, den Sie für Ihre Medien angeben. Wenn die Sprachen nicht übereinstimmen, wird Ihr benutzerdefiniertes Vokabular nicht auf Ihre Transkription angewendet und es gibt keine Warnungen oder Fehler.

Verwenden eines benutzerdefinierten Vokabulars in einer Batch-Transkription

Beispiele für die Verwendung eines benutzerdefinierten Vokabulars mit einer Batch-Transkription finden Sie im Folgenden:

AWS-Managementkonsole

1. Melden Sie sich an der [AWS-Managementkonsole](#) an.
2. Wählen Sie im Navigationsbereich Transkriptionsaufträge und dann Auftrag erstellen (oben rechts). Dies öffnet die Seite Auftragsdetails angeben.

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**

Benennen Sie Ihren Auftrag und geben Sie Ihr Eingabemedium an. Fügen Sie optional weitere Felder hinzu und wählen Sie dann Weiter.

3. Aktivieren Sie unten auf der Seite Auftrag konfigurieren im Bereich Anpassung die Option Benutzerdefiniertes Vokabular.

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. Wählen Sie Ihr benutzerdefiniertes Vokabular aus dem Dropdown-Menü.

Wählen Sie Auftrag erstellen, um Ihren Transkriptionsauftrag auszuführen.

AWS CLI

In diesem Beispiel werden der Befehl [start-Transkription-job](#) und der Parameter Settings mit dem Unterparameter VocabularyName verwendet. Weitere Informationen erhalten Sie unter [StartTranscriptionJob](#) und [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
```

Hier ein weiteres Beispiel mit dem Befehl [start-Transkription-job](#) und einem Anforderungstext, der Ihr benutzerdefiniertes Vokabular mit diesem Auftrag enthält.

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

Die Datei *my-first-vocabulary-job.json* enthält den folgenden Anforderungstext.

```
{  
  "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"  
  }  
}
```

AWS SDK für Python (Boto3)

In diesem Beispiel wird mithilfe des Settings Arguments für AWS SDK für Python (Boto3) die Methode [start_transcription_job](#) ein benutzerdefiniertes Vokabular hinzugefügt. Weitere Informationen erhalten Sie unter [StartTranscriptionJob](#) und [Settings](#).

Weitere Beispiele für die Verwendung der AWS SDKs, einschließlich funktionspezifischer, szenarienspezifischer und serviceübergreifender Beispiele, finden Sie im Kapitel. [Codebeispiele für Amazon Transcribe mit AWS SDKs](#)

```
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'
    }
)

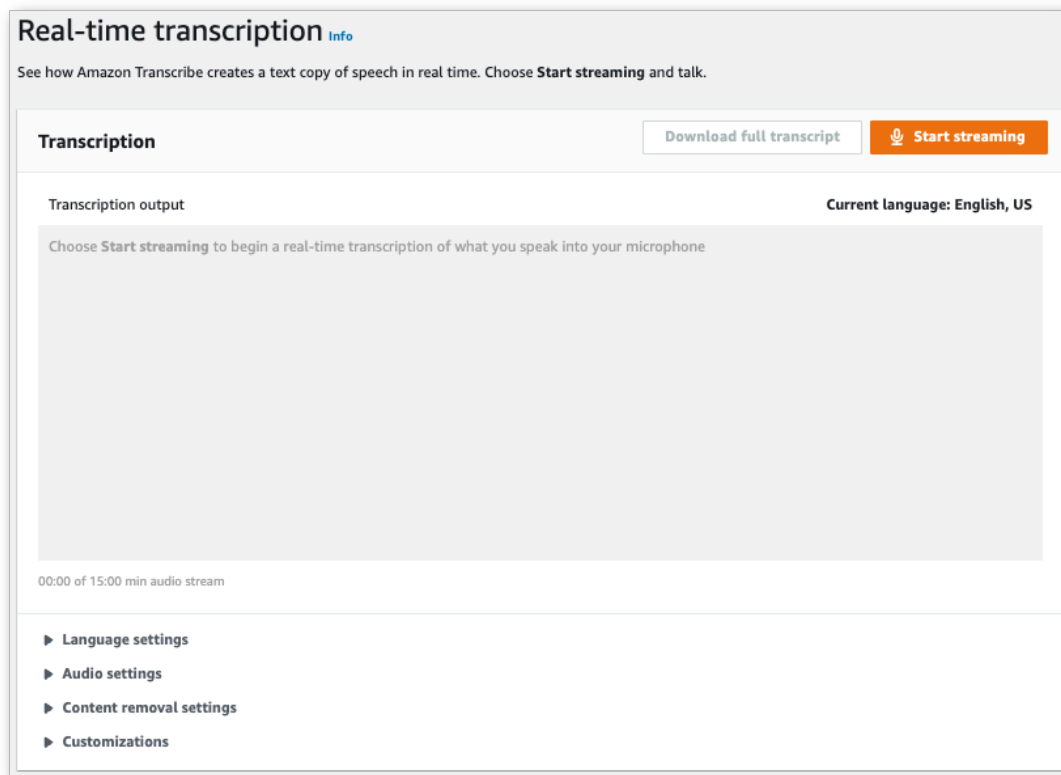
while True:
    status = transcribe.get_transcription_job(TranscriptionJobName = job_name)
    if status['TranscriptionJob']['TranscriptionJobStatus'] in ['COMPLETED', 'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
print(status)
```

Verwenden eines benutzerdefinierten Vokabulars in einer Streaming-Transkription

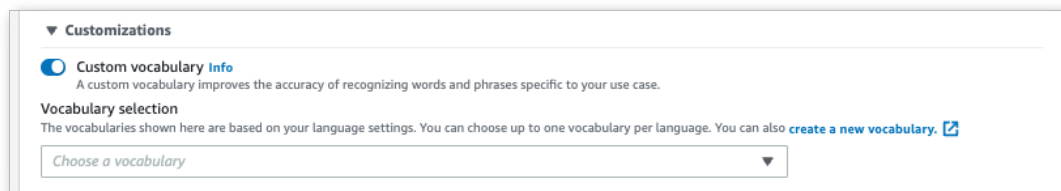
Beispiele für die Verwendung eines benutzerdefinierten Vokabulars mit einer Streaming-Transkription finden Sie im Folgenden:

AWS-Managementkonsole

1. Melden Sie sich beim [AWS-Managementkonsole](#) an.
2. Wählen Sie im Navigationsbereich Real-time Transkription aus. Blättern Sie nach unten zu Anpassungen und erweitern Sie dieses Feld, falls es minimiert ist.



3. Aktivieren Sie Benutzerdefiniertes Vokabular und wählen Sie ein benutzerdefiniertes Vokabular aus dem Dropdown-Menü.



Fügen Sie alle anderen Einstellungen hinzu, die Sie auf Ihren Stream anwenden möchten.

4. Jetzt können Sie Ihren Stream transkribieren. Wählen Sie Streaming starten und beginnen Sie zu sprechen. Um Ihr Diktat zu beenden, wählen Sie Streaming beenden.

HTTP/2 streamen

In diesem Beispiel wird eine HTTP/2 Anfrage erstellt, die Ihr benutzerdefiniertes Vokabular enthält. Weitere Informationen zur Verwendung von HTTP/2 Streaming mit Amazon Transcribe finden Sie unter [Einen HTTP/2 Stream einrichten](#). Weitere Einzelheiten zu Parametern und Kopfzeilen, die speziell für Amazon Transcribe gelten, finden Sie unter [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

```

Parameterdefinitionen finden Sie in der [API-Referenz](#). Parameter, die allen AWS API-Vorgängen gemeinsam sind, sind im Abschnitt [Allgemeine Parameter](#) aufgeführt.

WebSocket streamen

In diesem Beispiel wird eine vorsignierte URL erstellt, die Ihr benutzerdefiniertes Vokabular auf einen WebSocket Stream anwendet. Für eine bessere Lesbarkeit werden Zeilenumbrüche hinzugefügt. Weitere Informationen zur Verwendung von WebSocket Streams mit finden Sie Amazon Transcribe unter [Einen WebSocket Stream einrichten](#). Weitere Einzelheiten zu den Parametern finden Sie unter [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

```

Parameterdefinitionen finden Sie in der [API-Referenz](#). Parameter, die allen AWS API-Vorgängen gemeinsam sind, sind im Abschnitt [Allgemeine Parameter](#) aufgeführt.

Benutzerdefinierte Sprachmodelle

Benutzerdefinierte Sprachmodelle wurden entwickelt, um die Transkriptionsgenauigkeit für domainspezifische Sprache zu verbessern. Dazu gehören alle Inhalte, die über das hinausgehen, was man in normalen, alltäglichen Gesprächen hören würde. Wenn Sie z. B. den Bericht einer wissenschaftlichen Konferenz transkribieren, ist es unwahrscheinlich, dass eine Standardtranskription viele der von den Vortragenden verwendeten wissenschaftlichen Begriffe erkennt. In diesem Fall können Sie ein benutzerdefiniertes Sprachmodell trainieren, um die in Ihrem Fachgebiet verwendeten Fachbegriffe zu erkennen.

Im Gegensatz zu benutzerdefinierten Vokabularen, die die Erkennung eines Wortes durch Hinweise (wie z. B. die Aussprache) verbessern, lernen benutzerdefinierte Sprachmodelle den mit einem bestimmten Wort verbundenen Kontext. Dazu gehört, wie und wann ein Wort verwendet wird und in welcher Beziehung ein Wort zu anderen Wörtern steht. Wenn Sie Ihr Modell beispielsweise anhand von klimawissenschaftlichen Forschungsberichten trainieren, kann Ihr Modell lernen, dass "Eisscholle" ein wahrscheinlicheres Wortpaar ist als "Eisfluss".

Informationen zu den unterstützten Sprachen für benutzerdefinierte Sprachmodelle finden Sie unter [Unterstützte Sprachen und sprachspezifische Funktionen](#). Beachten Sie, dass Sie die Sprachidentifizierung nicht aktivieren können, wenn Sie ein benutzerdefiniertes Sprachmodell in Ihre Anfrage aufnehmen (Sie müssen einen Sprachcode angeben).

 API-Vorgänge speziell für benutzerdefinierte Sprachmodelle

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

Datenquellen

Sie können jede Art von Textdaten verwenden, die Sie für das Training Ihres Modells benötigen. Je näher Ihr Textinhalt jedoch an Ihrem Audioinhalt liegt, desto genauer ist Ihr Modell. Daher ist es wichtig, Textdaten zu wählen, die die gleichen Begriffe im gleichen Kontext wie Ihr Audio verwenden.

Die besten Daten für das Training eines Modells sind genaue Transkripte. Dies wird als domäneninterne Daten betrachtet. In-domain Textdaten haben genau dieselben Begriffe, dieselbe Verwendung und denselben Kontext wie das Audio, das Sie transkribieren möchten.

Wenn Sie keine genauen Transkripte haben, verwenden Sie Zeitschriftenartikel, technische Berichte, Whitepaper, Konferenzprotokolle, Bedienungsanleitungen, Nachrichtenartikel, Website-Inhalte und jeden anderen Text, der die gewünschten Begriffe enthält, die in einem ähnlichen Kontext wie Ihr Audio verwendet werden. Dies wird als bereichsbezogene Daten betrachtet.

Die Erstellung eines robusten benutzerdefinierten Sprachmodells kann eine beträchtliche Menge an Textdaten erfordern, die die in Ihrem Audio gesprochenen Begriffe enthalten müssen. Sie können bis Amazon Transcribe zu 2 GB an Textdaten bereitstellen, um Ihr Modell zu trainieren — diese Daten werden als Trainingsdaten bezeichnet. Wenn Sie über keine (oder nur wenige) In-Domain-Transkripte verfügen, können Sie optional bis zu 200 MB an Textdaten zur Optimierung Ihres Modells bereitstellen Amazon Transcribe — dies wird als Tuning-Daten bezeichnet.

Trainings- versus Tuningdaten

Der Zweck von Trainingsdaten besteht darin, zu vermitteln, neue Begriffe Amazon Transcribe zu erkennen und den Kontext zu verstehen, in dem diese Begriffe verwendet werden. Um ein robustes Modell zu erstellen, benötigt Amazon Transcribe unter Umständen eine große Menge an relevanten Textdaten. Es wird dringend empfohlen, so viele Trainingsdaten wie möglich zur Verfügung zu stellen, bis zu einer Grenze von 2 GB.

Der Zweck von Tuningdaten ist die Verfeinerung und Optimierung der kontextuellen Beziehungen, die Sie aus Ihren Trainingsdaten gelernt haben. Tuningdaten sind nicht erforderlich, um ein benutzerdefiniertes Sprachmodell zu erstellen.

Es liegt an Ihnen zu entscheiden, wie Sie die Trainings- und optional die Tuningdaten am besten auswählen. Jeder Fall ist einzigartig und hängt von der Art und Menge der Daten ab, die Sie haben. Tuningdaten werden empfohlen, wenn es an Trainingsdaten in einer Domain mangelt.

Wenn Sie sich dafür entscheiden, beide Datentypen einzubeziehen, dürfen sich Ihre Trainings- und Abstimmungsdaten nicht überschneiden; Trainings- und Tuningdaten sollten eindeutig sein. Sich überschneidende Daten können Ihr benutzerdefiniertes Sprachmodell verzerren und verzerren und so seine Genauigkeit beeinträchtigen.

Generell empfehlen wir, wann immer es möglich ist, akkurate Texte aus dem betreffenden Gebiet als Trainingsdaten zu verwenden. Hier sind einige allgemeine Szenarien, die in der Reihenfolge ihrer Präferenz aufgeführt sind:

- Wenn Sie über mehr als 10.000 Wörter exakten, domaininternen Transkripttextes verfügen, verwenden Sie diese als Trainingsdaten. In diesem Fall ist es nicht erforderlich, Tuningdaten aufzunehmen. Dies ist das ideale Szenario für das Training eines benutzerdefinierten Sprachmodells.
- Wenn Sie über exakten, domaininternen Transkripttext mit weniger als 10.000 Wörtern verfügen und nicht die gewünschten Ergebnisse erzielen, sollten Sie Ihre Trainingsdaten mit domainbezogenen schriftlichen Texten, z. B. technischen Berichten, ergänzen. Reservieren Sie in diesem Fall einen kleinen Teil (10-25 %) Ihrer bereichsinternen Transkriptdaten für die Verwendung als Tuningdaten.
- Wenn Sie keinen domaininternen Transkripttext haben, laden Sie Ihren gesamten domainbezogenen Text als Trainingsdaten hoch. In diesem Fall ist Text im Transkriptstil dem geschriebenen Text vorzuziehen. Dies ist das am wenigsten effektive Szenario für das Training eines benutzerdefinierten Sprachmodells.

Wenn Sie bereit sind, Ihr Modell zu erstellen, siehe [Erstellen eines benutzerdefinierten Sprachmodells](#).

Erstellen eines benutzerdefinierten Sprachmodells

Bevor Sie Ihr benutzerdefiniertes Sprachmodell erstellen können, müssen Sie:

- Bereiten Sie Ihre Daten vor. Die Daten müssen im reinen Textformat gespeichert werden und dürfen keine Sonderzeichen enthalten.
- Laden Sie Ihre Daten in einen Amazon S3 Bucket hoch. Es wird empfohlen, getrennte Ordner für Trainings- und Tuningdaten anzulegen.
- Stellen Sie sicher, Amazon Transcribe dass Sie Zugriff auf Ihren Amazon S3 Bucket haben. Sie müssen eine IAM Rolle angeben, die über Zugriffsberechtigungen für die Verwendung Ihrer Daten verfügt.

Aufbereitung Ihrer Daten

Sie können alle Ihre Daten in einer Datei zusammenfassen oder in mehreren Dateien speichern. Beachten Sie, dass die Abstimmungsdaten in einer von den Trainingsdaten getrennten Datei gespeichert werden müssen, wenn Sie sie einbeziehen möchten.

Es spielt keine Rolle, wie viele Textdateien Sie für Ihre Trainings- oder Tuningdaten verwenden. Das Hochladen einer Datei mit 100.000 Wörtern ergibt das gleiche Ergebnis wie das Hochladen von 10 Dateien mit 10.000 Wörtern. Bereiten Sie Ihre Textdaten so auf, wie es für Sie am bequemsten ist.

Vergewissern Sie sich, dass alle Ihre Datendateien die folgenden Kriterien erfüllen:

- Sie sind alle in der gleichen Sprache wie das Modell, das Sie erstellen möchten. Wenn Sie beispielsweise ein benutzerdefiniertes Sprachmodell erstellen möchten, das Audio in US-Englisch transkribiert (en-US), müssen alle Ihre Textdaten in US-Englisch sein.
- Sie sind im Klartextformat mit UTF-8 Kodierung.
- Sie enthalten keine Sonderzeichen oder Formatierungen, wie z. B. HTML-Tags.
- Sie sind zusammen maximal 2 GB groß für Trainingsdaten und 200 MB für Tuningdaten.

Wenn eines dieser Kriterien nicht erfüllt ist, schlägt Ihr Modell fehl.

Hochladen Ihrer Daten

Bevor Sie Ihre Daten hochladen, erstellen Sie einen neuen Ordner für Ihre Trainingsdaten. Wenn Sie Tuning-Daten verwenden, erstellen Sie einen weiteren separaten Ordner.

Die URIs für Ihre Buckets könnten wie folgt aussehen:

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

Laden Sie Ihre Trainings- und Abstimmungsdaten in die entsprechenden Buckets hoch.

Sie können zu einem späteren Zeitpunkt weitere Daten zu diesen Buckets hinzufügen. In diesem Fall müssen Sie Ihr Modell jedoch mit den neuen Daten neu erstellen. Bestehende Modelle können nicht mit neuen Daten aktualisiert werden.

Ermöglichung des Zugangs zu Ihren Daten

Um ein benutzerdefiniertes Sprachmodell zu erstellen, müssen Sie eine IAM Rolle angeben, die über Berechtigungen für den Zugriff auf Ihren Amazon S3 Bucket verfügt. Wenn Sie noch keine Rolle mit Zugriff auf den Amazon S3 Bucket haben, in dem Sie Ihre Trainingsdaten abgelegt haben, müssen Sie eine erstellen. Nachdem Sie eine Rolle erstellt haben, können Sie eine Richtlinie anhängen, um dieser Rolle Berechtigungen zu erteilen. Weisen Sie einem Benutzer keine Richtlinie zu.

Beispiele für Richtlinien finden Sie unter [Amazon Transcribe Beispiele für identitätsbasierte Politik](#).

Informationen zum Erstellen einer neuen IAM Identität finden Sie unter [IAM Identitäten \(Benutzer, Benutzergruppen und Rollen\)](#).

Weitere Informationen zur Richtlinie finden Sie unter:

- [Richtlinien und Genehmigungen in IAM](#)
- [Erstellen von IAM -Richtlinien](#)
- [Zugriffsverwaltung für AWS -Ressourcen](#)

Erstellen eines eigenen Sprachmodells

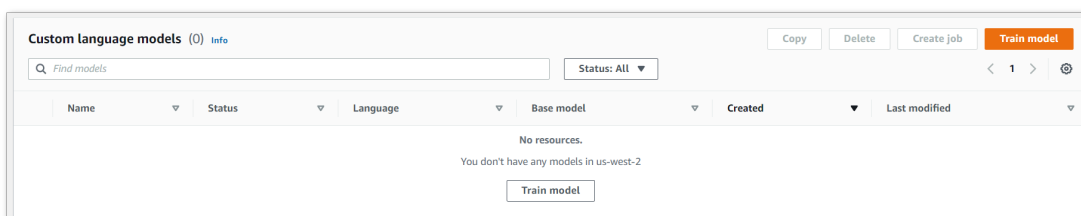
Wenn Sie Ihr eigenes Sprachmodell erstellen, müssen Sie ein Basismodell wählen. Es gibt zwei Optionen für das Basismodell:

- **NarrowBand:** Verwenden Sie diese Option für Audio mit einer Samplerate von weniger als 16.000 Hz. Dieser Modelltyp wird in der Regel für Telefongespräche verwendet, die mit 8.000 Hz aufgezeichnet werden.
- **WideBand:** Verwenden Sie diese Option für Audiodaten mit einer Samplerate größer oder gleich 16.000 Hz.

Mithilfe der AWS SDKs AWS-Managementkonsole AWS CLI, oder können Sie benutzerdefinierte Sprachmodelle erstellen. Weitere Beispiele finden Sie in den folgenden Beispielen:

AWS-Managementkonsole

1. Melden Sie sich an der [AWS-Managementkonsole](#) an.
2. Wählen Sie im Navigationsbereich die Option Benutzerdefiniertes Sprachmodell. Dadurch wird die Seite Benutzerdefinierte Sprachmodelle geöffnet, auf der Sie vorhandene benutzerdefinierte Sprachmodelle anzeigen oder ein neues benutzerdefiniertes Sprachmodell trainieren können.
3. Um ein neues Modell zu trainieren, wählen Sie Modell trainieren.



Dies führt Sie zur Seite Zugmodell. Fügen Sie einen Namen hinzu, geben Sie die Sprache an, und wählen Sie das gewünschte Basismodell für Ihr Modell. Fügen Sie dann den Pfad zu Ihren Trainings- und optional zu Ihren Tuning-Daten hinzu. Sie müssen eine IAM Rolle angeben, die über Berechtigungen für den Zugriff auf Ihre Daten verfügt.

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.

English, US (en-US) ▼

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.

 ▼

- Wenn Sie alle Felder ausgefüllt haben, wählen Sie unten auf der Seite das Zugmodell aus.

AWS CLI

In diesem Beispiel wird der Befehl [create-language-model](#) verwendet. Weitere Informationen erhalten Sie unter [CreateLanguageModel](#) und [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
```

Hier ein weiteres Beispiel mit dem Befehl [create-language-model](#) und einem Anforderungstext, der Ihr benutzerdefiniertes Sprachmodell erstellt.

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

Die Datei *my-first-language-model.json* enthält den folgenden Anforderungstext.

```
{  
  "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"  
}
```

AWS SDK für Python (Boto3)

In diesem Beispiel wird mithilfe der Methode AWS SDK für Python (Boto3) `create_language_model` ein CLM erstellt. Weitere Informationen erhalten Sie unter [CreateLanguageModel](#) und [LanguageModel](#).

Weitere Beispiele für die Verwendung der AWS SDKs, einschließlich funktionspezifischer, szenarienspezifischer und serviceübergreifender Beispiele, finden Sie im Kapitel. [Codebeispiele für Amazon Transcribe mit AWS SDKs](#)


```
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)
```

Aktualisierung des benutzerdefinierten Sprachmodells

Amazon Transcribe aktualisiert kontinuierlich die für benutzerdefinierte Sprachmodelle verfügbaren Basismodelle. Um von diesen Aktualisierungen zu profitieren, empfehlen wir, alle 6 bis 12 Monate neue benutzerdefinierte Sprachmodelle zu trainieren.

Um zu sehen, ob Ihr benutzerdefiniertes Sprachmodell das neueste Basismodell verwendet, führen Sie eine [DescribeLanguageModel](#)Anfrage mit dem AWS CLI oder einem AWS SDK aus und suchen Sie dann das UpgradeAvailability Feld in Ihrer Antwort.

Wenn UpgradeAvailability true lautet, läuft auf Ihrem Modell nicht die neueste Version des Basismodells. Um das neueste Basismodell in einem benutzerdefinierten Sprachmodell zu verwenden, müssen Sie ein neues benutzerdefiniertes Sprachmodell erstellen. Benutzerdefinierte Sprachmodelle können nicht aktualisiert werden.

Verwenden eines benutzerdefinierten Sprachmodells

Sobald Sie Ihr benutzerdefiniertes Sprachmodell erstellt haben, können Sie es in Ihre Transkriptionsanfragen einbeziehen. Beispiele finden Sie in den folgenden Abschnitten.

Die Sprache des Modells, das Sie in Ihre Anforderung aufnehmen, muss mit dem Sprachcode übereinstimmen, den Sie für Ihre Medien angeben. Wenn die Sprachen nicht übereinstimmen, wird Ihr benutzerdefiniertes Sprachmodell nicht auf Ihre Transkription angewendet und es gibt keine Warnungen oder Fehler.

Verwenden eines benutzerdefinierten Sprachmodells in einer Batch-Transkription

Beispiele für die Verwendung eines benutzerdefinierten Sprachmodells mit einer Batch-Transkription finden Sie im Folgenden:

AWS-Managementkonsole

1. Melden Sie sich an der [AWS-Managementkonsole](#) an.
2. Wählen Sie im Navigationsbereich Transkriptionsaufträge und dann Auftrag erstellen (oben rechts). Dies öffnet die Seite Auftragsdetails angeben.
3. Markieren Sie im Bereich Auftragseinstellungen unter Modelltyp das Feld Benutzerdefiniertes Sprachmodell.

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**

Sie müssen auch eine Eingabesprache aus dem Dropdown-Menü auswählen.

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)

4. Wählen Sie unter Auswahl des benutzerdefinierten Modells ein vorhandenes benutzerdefiniertes Sprachmodell aus dem Dropdown-Menü aus oder erstellen Sie ein neues Modell.

Fügen Sie den Amazon S3 Speicherort Ihrer Eingabedatei im Eingabedatenbereich hinzu.

5. Wählen Sie Weiter für weitere Konfigurationsoptionen.

Wählen Sie Auftrag erstellen, um Ihren Transkriptionsauftrag auszuführen.

AWS CLI

In diesem Beispiel werden der Befehl [start-Transkription-job](#) und der Parameter `ModelSettings` mit dem Unterparameter `VocabularyName` verwendet. Weitere Informationen erhalten Sie unter [StartTranscriptionJob](#) und [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
```

Hier ein weiteres Beispiel mit dem Befehl [start-Transkription-job](#) und einem Anforderungstext, der Ihr benutzerdefiniertes Sprachmodell mit diesem Auftrag enthält.

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

Die Datei *my-first-model-job.json* enthält den folgenden Anforderungstext.

```
{  
  "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"  
  }  
}
```

AWS SDK für Python (Boto3)

In diesem Beispiel wird mithilfe des AWS SDK für Python (Boto3) `ModelSettings` Arguments für die Methode [start_transcription_job](#) ein benutzerdefiniertes Sprachmodell eingefügt. Weitere Informationen erhalten Sie unter [StartTranscriptionJob](#) und [ModelSettings](#).

Weitere Beispiele für die Verwendung der AWS SDKs, einschließlich funktionspezifischer, szenarienspezifischer und serviceübergreifender Beispiele, finden Sie im Kapitel. [Codebeispiele für Amazon Transcribe mit AWS SDKs](#)

```
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'
    }
)

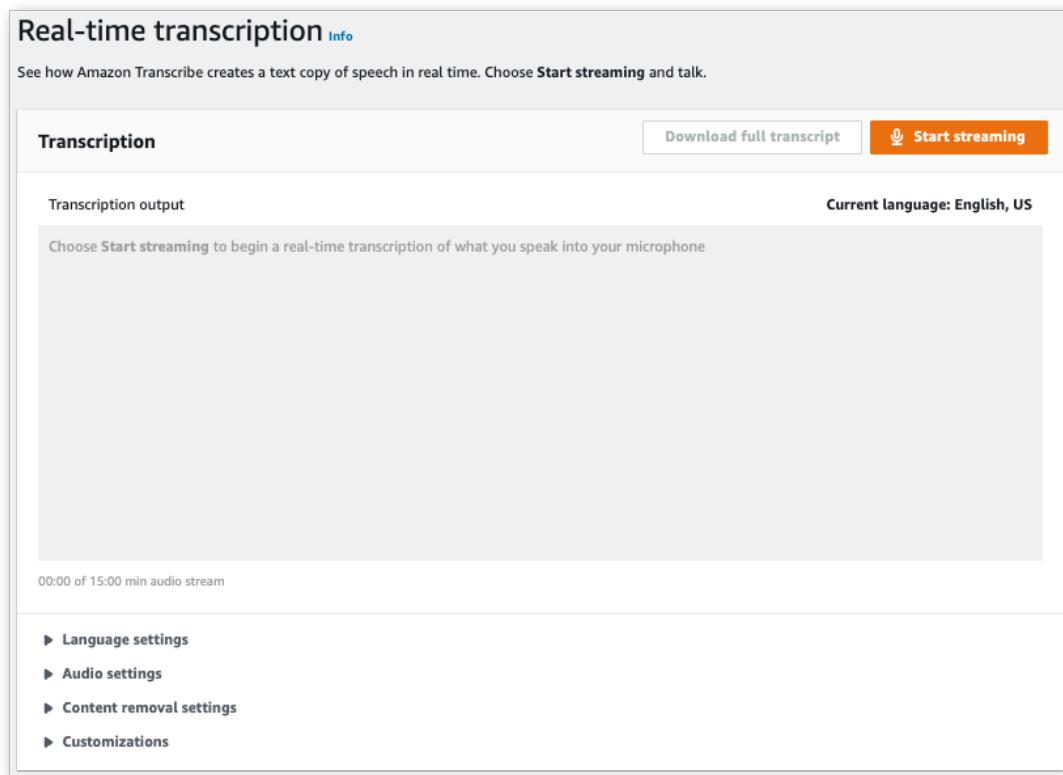
while True:
    status = transcribe.get_transcription_job(TranscriptionJobName = job_name)
    if status['TranscriptionJob']['TranscriptionJobStatus'] in ['COMPLETED', 'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
print(status)
```

Verwenden eines benutzerdefinierten Sprachmodells in einer Streaming-Transkription

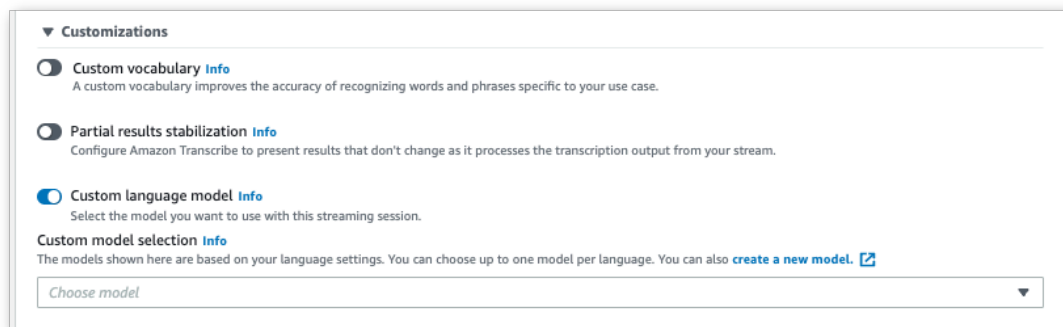
Beispiele für die Verwendung eines benutzerdefinierten Sprachmodells mit einer Streaming-Transkription finden Sie im Folgenden:

AWS-Managementkonsole

1. Melden Sie sich beim [AWS-Managementkonsole](#) an.
2. Wählen Sie im Navigationsbereich Real-time Transkription aus. Blättern Sie nach unten zu Anpassungen und erweitern Sie dieses Feld, falls es minimiert ist.



3. Aktivieren Sie die Option Benutzerdefiniertes Sprachmodell und wählen Sie ein Modell aus dem Dropdown-Menü.



Fügen Sie alle anderen Einstellungen hinzu, die Sie auf Ihren Stream anwenden möchten.

4. Jetzt können Sie Ihren Stream transkribieren. Wählen Sie Streaming starten und beginnen Sie zu sprechen. Um Ihr Diktat zu beenden, wählen Sie Streaming beenden.

HTTP/2 streamen

In diesem Beispiel wird eine HTTP/2 Anfrage erstellt, die Ihr benutzerdefiniertes Sprachmodell enthält. Weitere Informationen zur Verwendung von HTTP/2 Streaming mit Amazon Transcribe finden

Sie unter [Einen HTTP/2 Stream einrichten](#). Weitere Informationen zu spezifischen Parametern und Headern finden Sie Amazon Transcribe unter [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
```

Parameterdefinitionen finden Sie in der [API-Referenz](#). Parameter, die allen AWS API-Vorgängen gemeinsam sind, sind im Abschnitt [Allgemeine Parameter](#) aufgeführt.

WebSocket streamen

In diesem Beispiel wird eine vorsignierte URL erstellt, die Ihr benutzerdefiniertes Sprachmodell auf einen WebSocket Stream anwendet. Für eine bessere Lesbarkeit werden Zeilenumbrüche hinzugefügt. Weitere Informationen zur Verwendung von WebSocket Streams mit finden Sie Amazon Transcribe unter [Einen WebSocket Stream einrichten](#). Weitere Einzelheiten zu den Parametern finden Sie unter [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
```

Parameterdefinitionen finden Sie in der [API-Referenz](#). Parameter, die allen AWS API-Vorgängen gemeinsam sind, sind im Abschnitt [Allgemeine Parameter](#) aufgeführt.

Verwendung benutzerdefinierter Wortschatzfilter zum Löschen, Maskieren oder Markieren von Wörtern

Ein benutzerdefinierter Wortschatzfilter ist eine Textdatei, die eine benutzerdefinierte Liste von einzelnen Wörtern enthält, die Sie in Ihrer Transkriptionsausgabe ändern möchten.

Ein gängiger Anwendungsfall ist das Entfernen von anstößigen oder profanen Begriffen; benutzerdefinierte Wortschatzfilter sind jedoch vollständig benutzerdefiniert, sodass Sie alle gewünschten Wörter auswählen können. Wenn Sie zum Beispiel ein neues Produkt auf den Markt bringen wollen, können Sie den Produktnamen in Besprechungsprotokollen ausblenden. In diesem Fall halten Sie die Beteiligten auf dem Laufenden, während Sie den Produktnamen bis zur Markteinführung geheim halten.

Die Wortschatzfilterung verfügt über drei Anzeigemethoden: `mask`, `remove` und `tag`. Anhand der folgenden Beispiele können Sie sehen, wie die einzelnen Methoden funktionieren.

- Maskieren: Ersetzt die angegebenen Wörter durch drei Sternchen (***).

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

- Entfernen: Löscht die angegebenen Wörter und lässt nichts an deren Stelle stehen.

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

- Markierung: Fügt jedem angegebenen Wort eine Markierung (`"vocabularyFilterMatch": true`) hinzu, ändert aber das Wort selbst nicht. Tagging ermöglicht schnelle Transkripts substitutionen und -bearbeitungen.

```
"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
```

Wenn Sie eine Transkriptionsanfrage einreichen, können Sie einen benutzerdefinierten Vokabelfilter und die Filtermethode angeben, die Sie anwenden möchten. Amazon Transcribe ändert dann die genauen Wortübereinstimmungen, wenn sie in Ihrem Transkript erscheinen, entsprechend der von Ihnen angegebenen Filtermethode.

Benutzerdefinierte Wortschatzfilter können auf Batch- und Streaming-Transkriptionsanfragen angewendet werden. Wie Sie einen benutzerdefinierten Wortschatzfilter erstellen können, erfahren Sie unter [Erstellen eines Wortschatzfilters](#). Wie Sie Ihren benutzerdefinierten Wortschatzfilter anwenden können, erfahren Sie unter [Einen benutzerdefinierten Vokabularfilter verwenden](#).

Note

Amazon Transcribe maskiert automatisch rassistisch sensible Begriffe. Sie können diesen Standardfilter jedoch deaktivieren, indem Sie sich an den [AWS technischen](#) Support wenden.

Ein Video zur Erläuterung der Wortschatzfilterung finden Sie unter [Wortschatzfilter verwenden](#).

Spezifische API-Vorgänge für die Wortschatzfilterung

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

Erstellen eines Wortschatzfilters

Es gibt zwei Möglichkeiten, einen benutzerdefinierten Wortschatzfilter zu erstellen:

1. Speichern Sie eine Liste mit durch Zeilen getrennten Wörtern als Klartextdatei mit UTF-8 Kodierung.
 - Sie können diesen Ansatz mit den AWS SDKs AWS-Managementkonsole AWS CLI, oder verwenden.
 - Wenn Sie das verwenden AWS-Managementkonsole, können Sie einen lokalen Pfad oder eine Amazon S3 URI für Ihre benutzerdefinierte Vokabeldatei angeben.

- Wenn Sie die AWS SDKs AWS CLI oder verwenden, müssen Sie Ihre benutzerdefinierte Vokabeldatei in einen Amazon S3 Bucket hochladen und den Amazon S3 URI in Ihre Anfrage aufnehmen.
2. Fügen Sie eine Liste mit kommasetrennten Wörtern direkt in Ihre API-Anfrage ein.
- Sie können diesen Ansatz mit den AWS CLI oder AWS SDKs verwenden, indem Sie den [Words](#)Parameter verwenden.

Beispiele für die einzelnen Methoden finden Sie unter [Erstellen benutzerdefinierter Wortschatzfilter](#)

Das sollten Sie bei der Erstellung Ihres benutzerdefinierten Wortschatzfilters beachten:

- Bei Wörtern wird nicht zwischen Groß- und Kleinschreibung unterschieden. Zum Beispiel werden „curse“ und „CURSE“ gleich behandelt.
- Es werden nur exakte Wortübereinstimmungen gefiltert. Wenn Ihr Filter z. B. „swear“ enthält, Ihre Medien aber das Wort „swears“ oder „swearing“ enthalten, werden diese nicht gefiltert. Es werden nur Instanzen von „swear“ gefiltert. Sie müssen daher alle Varianten der Wörter, die Sie filtern möchten, einbeziehen.
- Die Filter gelten nicht für Wörter, die in anderen Wörtern enthalten sind. Wenn beispielsweise ein benutzerdefinierter Wortschatzfilter „Boot“, aber nicht „U-Boot“ enthält, wird „U-Boot“ im Transkript nicht geändert.
- Jeder Eintrag darf nur ein Wort enthalten (keine Leerzeichen).
- Wenn Sie Ihren benutzerdefinierten Vokabelfilter als Textdatei speichern, muss er im Klartextformat mit UTF-8 Kodierung vorliegen.
- Sie können bis zu 100 benutzerdefinierte Vokabelfilter verwenden AWS-Konto , die jeweils bis zu 50 KB groß sein können.
- Sie können nur Zeichen verwenden, die für Ihre Sprache unterstützt werden. Einzelheiten dazu finden Sie im [Zeichensatz](#) Ihrer Sprache.

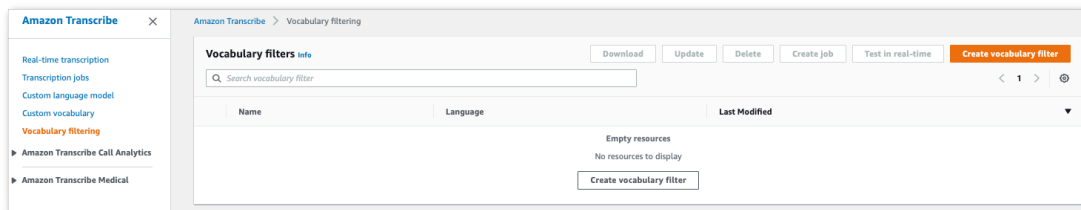
Erstellen benutzerdefinierter Wortschatzfilter

In den folgenden Beispielen erfahren Sie Amazon Transcribe, wie Sie einen benutzerdefinierten Vokabelfilter für die Verwendung mit verarbeiten können:

AWS-Managementkonsole

Bevor Sie fortfahren, speichern Sie Ihren benutzerdefinierten Wortschatzfilter als Textdatei (*.txt). Sie können Ihre Datei optional in einen Amazon S3 Bucket hochladen.

1. Melden Sie sich an der [AWS-Managementkonsole](#) an.
2. Wählen Sie im Navigationsbereich die Option Vokabularfilter aus. Dadurch wird die Seite Wortschatzfilter geöffnet, auf der Sie vorhandene benutzerdefinierte Wortschatzfilter anzeigen oder neue Filter erstellen können.
3. Wählen Sie Wortschatzfilter erstellen.



Sie gelangen auf die Seite Wortschatzfilter erstellen. Geben Sie einen Namen für Ihren neuen benutzerdefinierten Wortschatzfilter ein.

Wählen Sie unter Eingabequelle für den Wortschatz die Option Datei-Upload oder S3-Speicherort. Geben Sie dann den Speicherort für Ihre benutzerdefinierte Vokabulardatei an.

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.

- Optional können Sie Ihrem benutzerdefinierten Wortschatzfilter Tags hinzufügen. Sobald Sie alle Felder ausgefüllt haben, wählen Sie unten auf der Seite Wortschatzfilter erstellen. Wenn bei der Bearbeitung Ihrer Datei keine Fehler aufgetreten sind, gelangen Sie zurück zur Seite Wortschatzfilter.

Ihr benutzerdefinierter Wortschatzfilter ist nun einsatzbereit.

AWS CLI

In diesem Beispiel wird der Befehl [create-vocabulary-filter](#) verwendet, um eine Wortliste in einen benutzerdefinierten Wortschatzfilter zu verwandeln. Weitere Informationen finden Sie unter [CreateVocabularyFilter](#).

Option 1: Sie können Ihre Wortliste mit dem Parameter `words` in Ihre Anfrage aufnehmen.

```
aws transcribe create-vocabulary-filter \
--vocabulary-filter-name my-first-vocabulary-filter \
--language-code en-US \
--words profane,offensive,Amazon,Transcribe
```

Option 2: Sie können Ihre Wortliste als Textdatei speichern und in einen Amazon S3 -Bucket hochladen und dann den URI der Datei mit dem Parameter `vocabulary-filter-file-uri` in Ihre Anfrage aufnehmen.

```
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
```

Hier ein weiteres Beispiel mit dem Befehl [create-vocabulary-filter](#) und einem Anforderungstext, der Ihren benutzerdefinierten Wortschatzfilter erstellt.

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

Die Datei `my-first-vocab-filter.json` enthält den folgenden Anforderungstext.

Option 1: Sie können Ihre Wortliste mit dem Parameter `Words` in Ihre Anfrage aufnehmen.

```
{  
  "VocabularyFilterName": "my-first-vocabulary-filter",  
  "LanguageCode": "en-US",  
  "Words": [  
    "profane", "offensive", "Amazon", "Transcribe"  
  ]  
}
```

Option 2: Sie können Ihre Wortliste als Textdatei speichern und in einen Amazon S3 -Bucket hochladen und dann den URI der Datei mit dem Parameter `VocabularyFilterFileUri` in Ihre Anfrage aufnehmen.

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

 Note

Wenn Sie `VocabularyFilterFileUri` in Ihrer Anfrage angeben, können Sie nicht Words verwenden. Sie müssen entweder das eine oder das andere wählen.

AWS SDK für Python (Boto3)

In diesem Beispiel wird mithilfe der Methode AWS SDK für Python (Boto3) [create_vocabulary_filter](#) ein benutzerdefinierter Vokabelfilter erstellt. Weitere Informationen finden Sie unter [CreateVocabularyFilter](#).

Weitere Beispiele für die Verwendung der AWS SDKs, einschließlich funktionspezifischer, szenarienspezifischer und serviceübergreifender Beispiele, finden Sie im Kapitel. [Codebeispiele für Amazon Transcribe mit AWS SDKs](#)

Option 1: Sie können Ihre Wortliste mit dem Parameter Words in Ihre Anfrage aufnehmen.

```
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'
    ]
)
```

Option 2: Sie können Ihre Wortliste als Textdatei speichern und in einen Amazon S3 -Bucket hochladen und dann den URI der Datei mit dem Parameter `VocabularyFilterFileUri` in Ihre Anfrage aufnehmen.

```
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

Wenn Sie `VocabularyFilterFileUri` in Ihrer Anfrage angeben, können Sie nicht Words verwenden. Sie müssen entweder das eine oder das andere wählen.

Note

Wenn Sie einen neuen Amazon S3 Bucket für Ihre benutzerdefinierten Vokabelfilterdateien erstellen, stellen Sie sicher, dass die IAM Rolle, die die [CreateVocabularyFilter](#) Anfrage stellt, über Zugriffsberechtigungen für diesen Bucket verfügt. Wenn die Rolle nicht über die richtigen Berechtigungen verfügt, schlägt Ihre Anfrage fehl. Sie können optional eine IAM Rolle in Ihrer Anfrage angeben, indem Sie den `DataAccessRoleArn` Parameter angeben. Weitere Informationen zu IAM Rollen und Richtlinien finden Sie unter [Amazon Transcribe Beispiele für identitätsbasierte Politik](#). Amazon Transcribe

Einen benutzerdefinierten Vokalbularfilter verwenden

Sobald Ihr benutzerdefinierter Vokalbularfilter erstellt ist, können Sie ihn in Ihre Transkriptionsanfragen einfügen. Beispiele finden Sie in den folgenden Abschnitten.

Die Sprache des benutzerdefinierten Wortschatzfilters, den Sie in Ihre Anfrage aufnehmen, muss dem Sprachcode entsprechen, den Sie für Ihre Medien angeben. Wenn Sie die Sprachidentifikation verwenden und mehrere Sprachoptionen angeben, können Sie einen benutzerdefinierten Vokalbularfilter pro angegebener Sprache einfügen. Wenn die Sprachen Ihrer benutzerdefinierten Wortschatzfilter nicht mit der in Ihrem Audio identifizierten Sprache übereinstimmen, werden Ihre Filter nicht auf Ihre Transkription angewendet und es gibt keine Warnungen oder Fehler.

Verwenden eines benutzerdefinierten Vokabularfilters in einer Batch-Transkription

Beispiele für die Verwendung eines benutzerdefinierten Vokabularfilters bei einer Batch-Transkription finden Sie im Folgenden:

AWS-Managementkonsole

1. Melden Sie sich an der [AWS-Managementkonsole](#) an.
2. Wählen Sie im Navigationsbereich Transkriptionsaufträge und dann Auftrag erstellen (oben rechts). Dies öffnet die Seite Auftragsdetails angeben.

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**

Benennen Sie Ihren Auftrag und geben Sie Ihr Eingabemedium an. Fügen Sie optional weitere Felder hinzu und wählen Sie dann Weiter.

3. Aktivieren Sie auf der Seite Auftrag konfigurieren im Bereich Inhaltse Entfernung die Option Vokabularfilterung.

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. Wählen Sie Ihren benutzerdefinierten Vokabularfilter aus dem Dropdown-Menü und geben Sie die Filtermethode an.

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](#).

my-vocab-filter ▼

Vocabulary filtering method
Use a vocabulary filter to filter vocabularies from your transcript. For example, in the sentence "The quick brown fox jumps over the lazy dog," you remove the word "lazy" using the following options.

- ☒ **Mask vocabulary**
Example: The quick brown fox jumps over the *** dog.
- ☐ **Remove vocabulary**
Example: The quick brown fox jumps over the dog.
- ☐ **Tag vocabulary**
Example: The quick brown fox jumps over the lazy dog.

5. Wählen Sie Auftrag erstellen, um Ihren Transkriptionsauftrag auszuführen.

AWS CLI

In diesem Beispiel werden der Befehl [start-transcription-job](#) und der Parameter Settings mit den Unterparametern VocabularyFilterName und VocabularyFilterMethod verwendet. Weitere Informationen erhalten Sie unter [StartTranscriptionJob](#) und [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
```

Hier ein weiteres Beispiel mit dem Befehl [start-transcription-job](#) und einem Anforderungstext, der Ihren benutzerdefinierten Wortschatzfilter mit diesem Auftrag enthält.

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

Die Datei my-first-vocabulary-filter-job.json enthält den folgenden Anforderungstext.

```
{
  "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"
  }
}
```

AWS SDK für Python (Boto3)

In diesem Beispiel wird mithilfe des AWS SDK für Python (Boto3) `Settings` Arguments für die Methode [start_transcription_job](#) ein benutzerdefinierter Vokabelfilter eingefügt. Weitere Informationen erhalten Sie unter [StartTranscriptionJob](#) und [Settings](#).

Weitere Beispiele für die Verwendung der AWS SDKs, einschließlich funktionspezifischer, szenarienspezifischer und serviceübergreifender Beispiele, finden Sie im Kapitel. [Codebeispiele für Amazon Transcribe mit AWS SDKs](#)

```
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'
    }
)
```

```
)

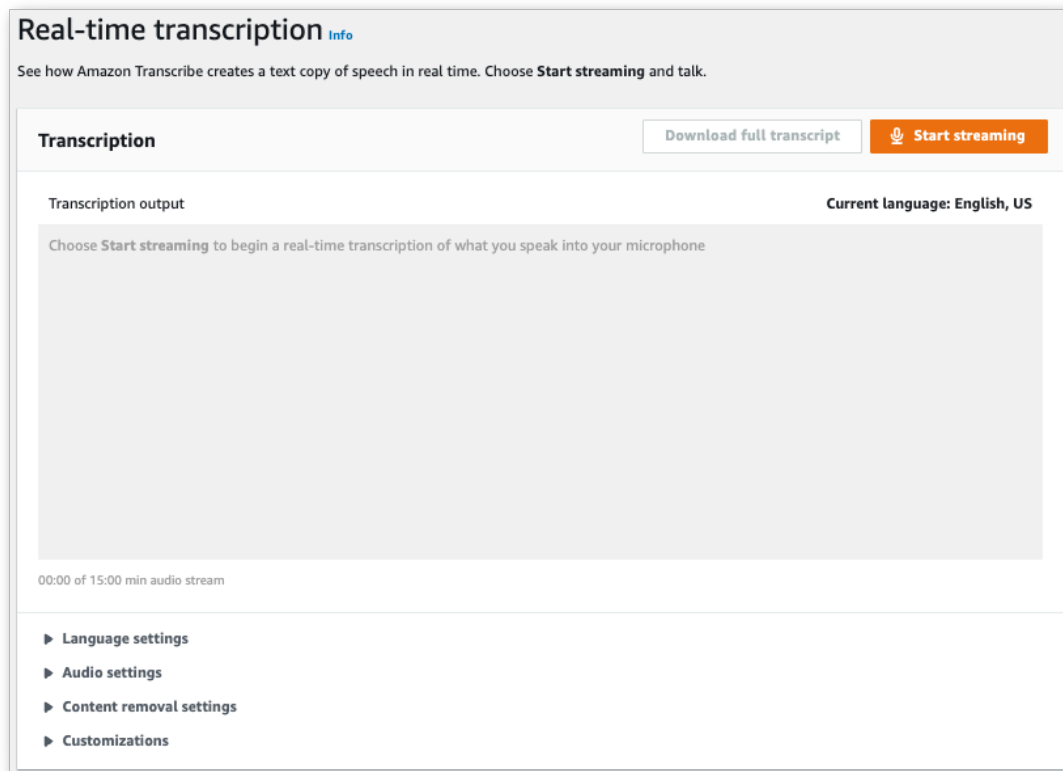
while True:
    status = transcribe.get_transcription_job(TranscriptionJobName = job_name)
    if status['TranscriptionJob']['TranscriptionJobStatus'] in ['COMPLETED', 'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
print(status)
```

Verwenden eines benutzerdefinierten Wortschatzfilters in einer Streaming-Transkription

Beispiele für die Verwendung eines benutzerdefinierten Vokabularfilters mit einer Streaming-Transkription finden Sie im Folgenden:

AWS-Managementkonsole

1. Melden Sie sich beim [AWS-Managementkonsole](#) an.
2. Wählen Sie im Navigationsbereich Real-time Transkription aus. Scrollen Sie nach unten zu Einstellungen für das Entfernen von Inhalten und erweitern Sie dieses Feld, falls es minimiert ist.



3. Aktivieren Sie die Vokabularfilterung. Wählen Sie einen benutzerdefinierten Vokabularfilter aus dem Dropdown-Menü und geben Sie die Filtermethode an.

Content removal settings

Vocabulary filtering [Info](#)
Vocabulary filtering removes, masks, or tags words that you specify in your vocabulary filter. Choose a vocabulary filter to see an example.

Filter selection
The vocabulary filters shown here are based on your language settings. You can choose up to one vocabulary filter per language. You can also [create a new vocabulary filter](#).

Vocabulary filtering method [Info](#)
Use a vocabulary filter to filter vocabularies from your transcript. For example, in the sentence "The quick brown fox jumps over the lazy dog," you remove the word "lazy" using the following options.

☒ **Mask vocabulary**
Example: The quick brown fox jumps over the *** dog.

☐ **Remove vocabulary**
Example: The quick brown fox jumps over the dog.

☐ **Tag vocabulary**
Example: The quick brown fox jumps over the lazy dog.

Fügen Sie alle anderen Einstellungen hinzu, die Sie auf Ihren Stream anwenden möchten.

4. Jetzt können Sie Ihren Stream transkribieren. Wählen Sie Streaming starten und beginnen Sie zu sprechen. Um Ihr Diktat zu beenden, wählen Sie Streaming beenden.

HTTP/2 streamen

In diesem Beispiel wird eine HTTP/2 Anfrage erstellt, die Ihren benutzerdefinierten Vokabelfilter und Ihre Filtermethode enthält. Weitere Informationen zur Verwendung von HTTP/2 Streaming mit Amazon Transcribe finden Sie unter [Einen HTTP/2 Stream einrichten](#). Weitere Informationen zu spezifischen Parametern und Headern finden Sie Amazon Transcribe unter [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
```

Parameterdefinitionen finden Sie in der [API-Referenz](#). Parameter, die allen AWS API-Vorgängen gemeinsam sind, sind im Abschnitt [Allgemeine Parameter](#) aufgeführt.

WebSocket streamen

In diesem Beispiel wird eine vorsignierte URL erstellt, die Ihren benutzerdefinierten Vokabelfilter auf einen WebSocket Stream anwendet. Für eine bessere Lesbarkeit werden Zeilenumbrüche hinzugefügt. Weitere Informationen zur Verwendung von WebSocket Streams mit finden Sie Amazon Transcribe unter [Einen WebSocket Stream einrichten](#). Weitere Einzelheiten zu den Parametern finden Sie unter [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
```

Parameterdefinitionen finden Sie in der [API-Referenz](#). Parameter, die allen AWS API-Vorgängen gemeinsam sind, sind im Abschnitt [Allgemeine Parameter](#) aufgeführt.

Erkennen von toxischer Sprache

Die Erkennung toxischer Sprache soll helfen, soziale Medienplattformen zu moderieren, die einen Dialog zwischen Gleichgesinnten beinhalten, wie z. B. Online-Spiele und soziale Chat-Plattformen. Die Verwendung toxischer Sprache kann für den Einzelnen, für Gleichaltrige und für Gemeinschaften sehr schädlich sein. Die Kennzeichnung von verletzendem Sprachgebrauch hilft den Organisationen, die Konversationen zivilisiert zu halten und ein sicheres und inklusives Online-Umfeld aufrechtzuerhalten, in dem die Nutzer frei kreieren, sich austauschen und beteiligen können.

Amazon Transcribe Toxicity Detection nutzt sowohl akustische als auch textbasierte Hinweise, um sprachbasierte toxische Inhalte in sieben Kategorien zu identifizieren und zu klassifizieren, darunter sexuelle Belästigung, Hassreden, Bedrohung, Missbrauch, Obszönität, Beleidigung und Grafik. Zusätzlich zum Text verwendet Amazon Transcribe -Erkennung der Toxizität Sprachhinweise, wie z. B. Töne und Tonhöhen, um toxische Absichten in Sprache zu erkennen. Dies ist eine Verbesserung gegenüber herkömmlichen Systemen zur Inhaltsmoderation, die nur auf bestimmte Begriffe abzielen, ohne die Absicht zu berücksichtigen.

Amazon Transcribe kennzeichnet und kategorisiert giftige Sprache, wodurch die Menge an Daten, die manuell verarbeitet werden müssen, minimiert wird. Dies ermöglicht es den Moderatoren, den Diskurs auf ihren Plattformen schnell und effizient zu steuern.

Zu den Kategorien der toxischen Sprache gehören:

- **Obszönität:** Sprache, die unhöfliche, vulgäre oder beleidigende Wörter, Formulierungen oder Abkürzungen enthält.
- **Hassrede:** Äußerungen, die eine Person oder Gruppe aufgrund ihrer Identität (z. B. Rasse, ethnische Zugehörigkeit, Geschlecht, Religion, sexuelle Orientierung, Fähigkeiten und nationale Herkunft) kritisieren, beleidigen, anprangern oder entmenslichen
- **Sexuell:** Sprache, die sexuelles Interesse, Aktivität oder Erregung durch direkte oder indirekte Anspielungen auf Körperteile, körperliche Merkmale oder Geschlecht anzeigt
- **Beleidigungen:** Äußerungen, die eine erniedrigende, demütigende, spöttische, beleidigende oder herabsetzende Sprache enthalten. Diese Art von Sprache wird auch als Mobbing bezeichnet.
- **Gewalt oder Bedrohung:** Äußerungen, die Drohungen enthalten, die darauf abzielen, einer Person oder Gruppe Schmerzen, Verletzungen oder Feindseligkeit zuzufügen.
- **Grafisch:** Sprache, die visuell beschreibende und unangenehm anschauliche Bilder verwendet. Diese Art von Sprache ist oft absichtlich bildreich, um das Unbehagen des Empfängers zu verstärken.

- **Belästigung oder Beleidigung:** Äußerungen, die darauf abzielen, das psychische Wohlbefinden des Empfängers zu beeinträchtigen, einschließlich erniedrigender und objektivierender Begriffe. Diese Art von Sprache wird auch als Belästigung bezeichnet.

Bei der Toxizitätserkennung werden Sprachsegmente (die Sprache zwischen natürlichen Pausen) analysiert und diesen Segmenten Konfidenzwerte zugewiesen. Konfidenzwerte sind Werte zwischen 0 und 1. Ein höherer Konfidenzwert bedeutet eine größere Wahrscheinlichkeit, dass es sich bei dem Inhalt um toxische Sprache in der entsprechenden Kategorie handelt. Sie können diese Konfidenzwerte verwenden, um den geeigneten Schwellenwert für die Erkennung von Toxizität für Ihren Anwendungsfall festzulegen.

Note

Die Erkennung von Toxizität ist nur für Batch-Transkriptionen in US-Englisch verfügbar (en-US).

[Beispielausgabe](#) im JSON-Format anzeigen.

Erkennen toxischer Sprache

Verwenden der Erkennung toxischer Sprache in einer Batch-Transkription

Beispiele für die Verwendung der Erkennung toxischer Sprache während einer Batch-Transkription finden Sie im Folgenden:

AWS-Managementkonsole

1. Melden Sie sich an der [AWS-Managementkonsole](#) an.
2. Wählen Sie im Navigationsbereich Transkriptionsaufträge und dann Auftrag erstellen (oben rechts). Dies öffnet die Seite Auftragsdetails angeben.

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. Auf der Seite Auftragsdetails angeben können Sie auch die PII-Schwärzung aktivieren, wenn Sie dies wünschen. Beachten Sie, dass die anderen aufgeführten Optionen bei der Toxizitätserkennung nicht unterstützt werden. Klicken Sie auf Weiter. Hier gelangen Sie zur Seite Auftrag konfigurieren – optional . Wählen Sie im Bereich Audioeinstellungen die Option Toxizitätserkennung.

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

Previous

Create job

4. Wählen Sie Auftrag erstellen, um Ihren Transkriptionsauftrag auszuführen.
5. Sobald Ihr Transkriptionsauftrag abgeschlossen ist, können Sie Ihr Transkript über das Dropdown-Menü Download auf der Detailseite des Transkriptionsauftrags herunterladen.

AWS CLI

In diesem Beispiel werden der Befehl [start-transcription-job](#) und der Parameter `ToxicityDetection` verwendet. Weitere Informationen erhalten Sie unter [StartTranscriptionJob](#) und [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
```

Hier ein weiteres Beispiel mit dem Befehl [start-transcription-job](#) und einem Anforderungstext, der die Toxizitätserkennung enthält.

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

Die Datei `my-first-toxicity-job.json` enthält den folgenden Anforderungstext.

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

AWS SDK für Python (Boto3)

In diesem Beispiel wird die Methode AWS SDK für Python (Boto3) `enable ToxicityDetection` für die Methode [start_transcription_job](#) verwendet. Weitere Informationen erhalten Sie unter [StartTranscriptionJob](#) und [ToxicityDetection](#).

Weitere Beispiele für die Verwendung der AWS SDKs, einschließlich funktionspezifischer, szenarienspezifischer und serviceübergreifender Beispiele, finden Sie im Kapitel. [Codebeispiele für Amazon Transcribe mit AWS SDKs](#)

```
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)
```

Beispielausgabe

Negative Sprache wird in Ihrer Transkriptionsausgabe mit Tags versehen und kategorisiert. Jede Instance der negativen Sprache wird kategorisiert und mit einem Konfidenzwert (einem Wert zwischen 0 und 1) versehen. Ein größerer Konfidenzwert zeigt an, dass es sich bei dem Inhalt mit größerer Wahrscheinlichkeit um negative Sprache innerhalb der angegebenen Kategorie handelt.

Beispielausgabe (JSON)

Nachfolgend finden Sie eine Beispielausgabe im JSON-Format, die kategorisierte negative Sprache mit den zugehörigen Konfidenzwerten anzeigt.

```
{
  "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
      },
      {
        "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"
}
```


Schwärzung oder Identifizierung persönlich identifizierbarer Informationen

Die Schwärzung wird verwendet, um sensible Inhalte in Form von persönlich identifizierbaren Informationen (PII) in Ihren Transkripten zu maskieren oder daraus zu entfernen. Die Arten von personenbezogenen Daten, die redigiert Amazon Transcribe werden können, variieren zwischen Batch- und Streaming-Transkriptionen. Die PII-Liste für jede Transkriptionsmethode finden Sie unter [Schwärzen von PII in Ihrem Batch-Job](#) und [Schwärzen oder Identifizieren von PII in einem Echtzeit-Datenstrom](#). Bei Streaming-Transkriptionen haben Sie auch die Möglichkeit, PII zu kennzeichnen, ohne sie zu schwärzen; ein Beispiel für die Ausgabe finden Sie unter [Beispiel für die Ausgabe der PII-Identifikation](#).

Wenn die Schwärzung aktiviert ist, haben Sie die Möglichkeit, nur ein geschwärztes Transkript oder sowohl ein geschwärztes als auch ein ungeschwärztes Transkript zu erstellen. Wenn Sie sich dafür entscheiden, nur ein geschwärztes Transkript zu erstellen, beachten Sie, dass Ihr Datenträger der einzige Ort ist, an dem das vollständige Gespräch gespeichert wird. Wenn Sie Ihre Originaldatenträger löschen, gibt es keine Aufzeichnungen über die ungeschwärzten PII. Aus diesem Grund kann es sinnvoll sein, zusätzlich zu einem geschwärzten Transkript auch ein ungeschwärztes Transkript zu erstellen.

Weitere Informationen über die Schwärzung von PII bei Batch-Transkriptionen finden Sie unter: [Schwärzen von PII in Ihrem Batch-Job](#).

Weitere Informationen zur Schwärzung oder Identifizierung von personenbezogenen Daten bei Streaming-Transkriptionen finden Sie unter: [Schwärzen oder Identifizieren von PII in einem Echtzeit-Datenstrom](#).

Important

Die SchwärzungsFeature dient dazu, sensible Daten zu identifizieren und zu entfernen. Aufgrund des prädiktiven Charakters des maschinellen Lernens können jedoch Amazon Transcribe möglicherweise nicht alle Fälle sensibler Daten in Ihrem Transkript identifiziert und entfernt werden. Wir empfehlen Ihnen dringend, alle geschwärzten Ausgaben zu überprüfen, um sicherzustellen, dass sie Ihren Anforderungen entsprechen.

Die Schwärzungsfunktion erfüllt nicht die Anforderungen an die Anonymisierung nach Gesetzen zum Schutz medizinischer Daten, wie z. B. U.S Gesetz über die Portabilität und Rechenschaftspflicht von Krankenversicherungen von 1996 (HIPAA).

Eine Videoanleitung zur Redaktionsfunktion von finden Sie unter [Verwenden Amazon Transcribe der Inhaltsredaktion](#) zur Identifizierung und Schwärzung personenbezogener Daten.

Schwärzen von PII in Ihrem Batch-Job

Wenn Sie während eines Batch-Transkriptionsvorgangs personenbezogene Daten (PII) aus einem Transkript redigieren, wird jede identifizierte Instanz von PII durch den Haupttext Ihres [PII] Transkripts Amazon Transcribe ersetzt. Sie können auch die Art der PII anzeigen, die im Wort-für-Wort-Teil der Transkriptionsausgabe geschwärzt wurden. Ein Beispiel für die Ausgabe finden Sie unter [Beispiel für eine geschwärzte Ausgabe \(Batch\)](#).

Die Schwärzung mit Batch-Transkriptionen ist im englischen Dialekt verfügbar: US (en-US), im spanischen Dialekt: USA (es-US), in französischen Dialekten: Französisch (fr-FR), Kanada (), in deutschen Dialekten: Deutschland (fr-CA), Schweizerisch (de-DE), im italienischen Dialekt: Italien (de-CH) und in portugiesischen Dialekten: Portugal (it-IT), Brasilien (). pt-PT pt-BR Die Schwärzung ist nicht mit der [Sprachidentifikation](#) kompatibel.

Sowohl geschwärzte als auch unredigierte Transkripte werden im selben Ausgabe-Bucket gespeichert. Amazon S3 Amazon Transcribe speichert sie in einem von Ihnen angegebenen Bucket oder in dem vom Service verwalteten Amazon S3 Standard-Bucket.

Arten von personenbezogenen Daten Amazon Transcribe kann Batch-Transkriptionen erkennen

PII-Typ	Description
ADDRESS	Eine physische Adresse, z. B. 100 Main Street, Anytown, USA oder Suite #12, Gebäude 123. Eine Adresse kann eine Straße, ein Gebäude, einen Ort, eine Stadt, ein Bundesland, ein Land, eine Grafschaft, eine Postleitzahl, einen Bezirk, ein Stadtviertel und mehr enthalten.
AGE	Das Alter einer Person, einschließlich Menge und Zeiteinheit. In der Formulierung „Ich bin 40

PII-Typ	Description
	Jahre alt“ wird beispielsweise „40 Jahre“ als Alter Amazon Transcribe erkannt.
ALL	Schwärzen oder identifizieren Sie alle in dieser Tabelle aufgeführten PII-Typen.
AWS_ACCESS_KEY	Eine eindeutige Kennung, die mit einem geheimen Zugriffsschlüssel verknüpft ist. Sie verwenden die Zugriffsschlüssel-ID und den geheimen Zugriffsschlüssel, um programmatische AWS-Anfragen kryptografisch zu signieren.
AWS_SECRET_KEY	Eine eindeutige ID, die einem Zugriffsschlüssel zugeordnet ist. Sie verwenden die Zugriffsschlüssel-ID und den geheimen Zugriffsschlüssel, um programmatische AWS-Anfragen kryptografisch zu signieren.
BANK_ACCOUNT_NUMBER	Eine US-Bankkontonummer. Diese sind in der Regel zwischen 10 und 12 Ziffern lang, aber Amazon Transcribe erkennt auch Bankkontonummern, bei denen nur die letzten 4 Ziffern vorhanden sind.
BANK_ROUTING	Eine US-Bankleitzahl. Diese sind in der Regel 9 Ziffern lang, aber Amazon Transcribe erkennt auch Routing-Nummern, bei denen nur die letzten 4 Ziffern vorhanden sind.
CA_HEALTH_NUMBER	Eine kanadische Gesundheitsdienstnummer ist eine 10-stellige eindeutige Kennung, die für den Zugang zu Gesundheitsleistungen erforderlich ist.

PII-Typ	Description
CA_SOCIAL_INSURANCE_NUMBER	Eine kanadische Sozialversicherungsnummer (SIN) ist eine 9-stellige eindeutige Kennung, die Einzelpersonen benötigen, um staatliche Programme und Leistungen in Anspruch zu nehmen.
CREDIT_DEBIT_CVV	Ein dreistelliger Kartenprüfcode (CVV), der auf Kredit- und Debitkarten von VISA und Discover vorhanden ist. MasterCard Bei American Express-Kredit- oder Debitkarten handelt es sich um einen 4-stelligen Zahlencode.
CREDIT_DEBIT_EXPIRY	Das Ablaufdatum einer Kredit- oder Debitkarte. Diese Nummer ist normalerweise vierstellig und als oder formatiert. month/year MM/YY Amazon Transcribe Kann beispielsweise Ablaufdaten wie 01/2101/2021, und Januar 2021 erkennen.
CREDIT_DEBIT_NUMBER	Die Nummer einer Kredit- oder Debitkarte. Diese Zahlen können zwischen 13 und 16 Ziffern lang sein, erkennen aber Amazon Transcribe auch Kredit- oder Debitkart ennummern, wenn nur die letzten 4 Ziffern vorhanden sind.
DATE_TIME	Ein Datum kann ein Jahr, einen Monat, einen Tag, einen Wochentag oder eine Tageszeit enthalten. Amazon Transcribe Erkennt beispielsweise „19. Januar 2020“ oder „11 Uhr“ als Datum. Amazon Transcribe erkennt Teildaten, Datumsbereiche und Datumsintervalle. Es wird auch Jahrzehnte wie „die 1990er“ berücksichtigen.

PII-Typ	Description
DRIVER_ID	Die Nummer, die einem Führerschein zugewiesen ist. Dabei handelt es sich um ein offizielles Dokument, das es einer Person ermöglicht, ein oder mehrere motorisierte Fahrzeuge auf einer öffentlichen Straße zu fahren. Eine Führerscheinnummer besteht aus alphanumerischen Zeichen.
EMAIL	Eine E-Mail-Adresse, z. B. efua.owusu@email.com.
INTERNATIONAL_BANK_ACCOUNT_NUMBER	Eine internationale Bankkontonummer hat in jedem Land spezifische Formate. Weitere Informationen finden Sie unter www.iban.com/structure .
IP_ADDRESS	Eine IPv4-Adresse, z. B. 198.51.100.0.
LICENSE_PLATE	Ein Nummernschild für ein Fahrzeug wird von dem Staat oder Land ausgestellt, in dem das Fahrzeug registriert ist. Das Format für Personenkraftwagen besteht in der Regel aus fünf bis acht Zeichen, mit Großbuchstaben gefolgt von Ziffern. Das Format variiert je nach Standort des ausstellenden Staates oder Landes.
MAC_ADDRESS	Eine MAC-Adresse (Media Access Control) ist eine eindeutige Kennung, die einem Network Interface Controller (NIC) zugewiesen ist.

PII-Typ	Description
NAME	Der Name einer Person. Dieser Entitätstyp umfasst keine Titel wie Herr, Frau, Fräulein oder Dr.. Er wendet diesen Entitätstyp Amazon Transcribe nicht auf Namen an, die Teil von Organisationen oder Adressen sind. Amazon Transcribe Erkennt beispielsweise die John Doe Organization als Organisation und Jane Doe Street als Adresse.
PASSPORT_NUMBER	Eine eindeutige Kennung, die dem Reisepass einer Person zugewiesen wird. Das Format umfasst in der Regel eine Kombination aus Buchstaben und Zahlen und ist je nach Land unterschiedlich.
PASSWORD	Eine alphanumerische Zeichenfolge, die als Passwort verwendet wird, z. B. „*very20special#pass*“.
PHONE	Eine Telefonnummer. Dieser Entitätstyp umfasst auch Fax- und Pager-Nummern.
PIN	Eine 4-stellige persönliche Identifikationsnummer (PIN), mit der jemand Zugang zu seinen Kontodaten erhält.
SSN	Eine Sozialversicherungsnummer (SSN) ist eine 9-stellige Nummer, die an US-Bürger, Personen mit ständigem Wohnsitz und Personen mit vorübergehender Erwerbstätigkeit vergeben wird. Amazon Transcribe erkennt auch Sozialversicherungsnummern, wenn nur die letzten 4 Ziffern vorhanden sind.

PII-Typ	Description
SWIFT_CODE	Ein SWIFT-Code ist ein Standardformat für den Bank Identifier Code (BIC), der zur Angabe einer bestimmten Bank oder Filiale verwendet wird. Banken verwenden diese Codes für Geldbewegungen wie internationale Überweisungen. SWIFT-Codes bestehen aus 8 oder 11 Zeichen. Die 11-stelligen Codes beziehen sich auf bestimmte Filialen, während sich die 8-stelligen Codes (oder 11-stellige Codes, die auf „XXX“ enden) auf den Hauptsitz oder die Hauptniederlassung beziehen.
URL	Eine Webadresse, z. B. www.example.com .
US_INDIVIDUAL_TAX_IDENTIFICATION_NUMBER	Eine US-amerikanische Steueridentifikationsnummer (ITIN) ist eine 9-stellige Zahl, die mit einer „9“ beginnt und eine „7“ oder „8“ als vierte Ziffer enthält. Eine ITIN kann mit einem Leerzeichen oder einem Gedankenstrich nach der dritten und vierten Ziffer formatiert werden.
USERNAME	Ein Benutzername, der ein Konto identifiziert, wie etwa ein Login-Name, Bildschirmname, Spitzname oder Handle.
VEHICLE_IDENTIFICATION_NUMBER	Eine Fahrgestellnummern (VIN) identifiziert ein Fahrzeug eindeutig. Inhalt und Format der Fahrgestellnummer sind in der Spezifikation ISO 3779 definiert. Jedes Land hat spezifische Codes und Formate für VINs.

Sie können einen Batch-Transkriptionsauftrag mit dem SDK AWS-Managementkonsole AWS CLI, oder starten. AWS

AWS-Managementkonsole

1. Melden Sie sich an der [AWS-Managementkonsole](#) an.
2. Wählen Sie im Navigationsbereich Transkriptionsaufträge und dann Auftrag erstellen (oben rechts). Daraufhin wird die Seite Auftragsdetails angeben geöffnet.
3. Nachdem Sie die gewünschten Felder auf der Seite Auftragsdetails angeben ausgefüllt haben, wählen Sie Weiter, um zur Seite Auftrag konfigurieren – optional zu gelangen. Hier finden Sie den Bereich zum Entfernen von Inhalten mit dem Schalter für die Schwärzung von PII .

Configure job - optional [Info](#)

Audio settings

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

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

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

☐ **PII redaction** [Info](#)
Label the type of PII and also mask the content with the PII entity type in the transcription output. For example, (123) 456-7890 will be masked as [PHONE].

4. Sobald Sie die Schwärzung von PII ausgewählt haben, haben Sie die Möglichkeit, alle PII-Typen auszuwählen, die Sie schwärzen möchten. Sie können auch festlegen, dass ein ungeschwärztes Transkript angezeigt wird, wenn Sie Ungeschwärztes Transkript in das Auftragsausgabefeld einschließen auswählen.

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)**

☒ 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

☐ **Vocabulary filtering** [Info](#)

Vocabulary filtering can remove, mask or tag specified words in the final transcript.

5. Wählen Sie Auftrag erstellen, um Ihren Transkriptionsauftrag auszuführen.

AWS CLI

In diesem Beispiel werden der Befehl [start-Transkription-job](#) und der Parameter content-redaction verwendet. Weitere Informationen erhalten Sie unter [StartTranscriptionJob](#) und [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
```

Hier ein weiteres Beispiel, bei dem die Methode [start-Transkription-job](#) verwendet wird und der Anforderungstext die PII für diesen Auftrag unkenntlich macht.

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

Die Datei *my-first-redaction-job.json* enthält den folgenden Anforderungstext.

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

AWS SDK für Python (Boto3)

[In diesem Beispiel wird der verwendet AWS SDK für Python \(Boto3\) , um Inhalte mithilfe des ContentRedaction Arguments für die Methode start_transcription_job zu redigieren.](#) Weitere Informationen erhalten Sie unter [StartTranscriptionJob](#) und [ContentRedaction](#).

Weitere Beispiele für die Verwendung der AWS SDKs, einschließlich funktionspezifischer, szenarienspezifischer und serviceübergreifender Beispiele, finden Sie im Kapitel. [Codebeispiele für Amazon Transcribe mit AWS SDKs](#)

```
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

Die Bearbeitung personenbezogener Daten für Batch-Jobs wird nur in folgenden Fällen unterstützt AWS-Regionen: Asien-Pazifik (Hongkong), Asien-Pazifik (Mumbai), Asien-Pazifik (Seoul), Asien-Pazifik (Singapur), Asien-Pazifik (Sydney), Asien-Pazifik GovCloud (TokioUS-West), (), Kanada (Zentral), EU (Frankfurt), EU (Irland), EU (London), EU (Paris), Naher Osten (Bahrain), Südamerika (Sao Paulo), USA Ost (N. Virginia), USA Ost (Ohio), USA West (Oregon) und USA West (Nordkalifornien).

Schwärzen oder Identifizieren von PII in einem Echtzeit-Datenstrom

Amazon Transcribe Ersetzt bei der Schwärzung von persönlich identifizierbaren Informationen (PII) aus einer Streaming-Transkription jede identifizierte Instanz von PII durch [PII] in Ihrem Transkript.

Eine zusätzliche Option für Streaming-Transkriptionen ist die Identifizierung von personenbezogenen Daten. Wenn Sie PII Identification aktivieren, Amazon Transcribe kennzeichnet die PII in Ihren Transkriptionsergebnissen unter einem Objekt. **Entities** Ein Beispiel für die Ausgabe finden Sie unter [Beispiel für eine geschwärzte Streaming-Ausgaben](#) und [Beispiel für die Ausgabe der PII-Identifikation](#).

Die Bearbeitung und Identifizierung personenbezogener Daten mit Streaming-Transkriptionen ist in den folgenden englischen Dialekten möglich: Schottisch (en-AB), Australien (en-AU), Kanada (en-CA), Britisch (en-GB), Irland (en-IE), Indien (en-IN), Neuseeland (), Vereinigte Staaten (en-NZ), Wales (en-US) und Südafrika (en-WL); spanische Dialekte: USA (en-ZA), Spanien (); Französische Dialekte: Französisch (es-USes-ES), Kanada (); Portugiesische Dialekte: Portugal (fr-FRfr-CA), Brasilien (); italienischer Dialekt: Italien pt-PT pt-BR (it-IT); und deutsche Dialekte: Deutschland (de-DE), Schweizerisch (de-CH).

Die Identifizierung und Schwärzung von personenbezogenen Daten bei Streaming-Aufträgen erfolgt erst nach vollständiger Transkription der Audiosegmente.

Arten von personenbezogenen Daten Amazon Transcribe kann Streaming-Transkriptionen erkennen

PII-Typ	Description
ADDRESS	Eine physische Adresse, z. B. 100 Main Street, Anytown, USA oder Suite #12, Gebäude 123. Eine Adresse kann eine Straße, ein Gebäude, einen Ort, eine Stadt, ein Bundesland, ein Land, eine Grafschaft, eine Postleitzahl, einen Bezirk, ein Stadtviertel und mehr enthalten.
ALL	Schwärzen oder identifizieren Sie alle in dieser Tabelle aufgeführten PII-Typen.
BANK_ACCOUNT_NUMBER	Eine US-Bankkontonummer. Diese sind in der Regel zwischen 10 und 12 Ziffern lang, aber Amazon Transcribe erkennt auch Bankkontonummern, bei denen nur die letzten 4 Ziffern vorhanden sind.
BANK_ROUTING	Eine US-Bankleitzahl. Diese sind in der Regel 9 Ziffern lang, aber Amazon Transcribe erkennt

PII-Typ	Description
	auch Routing-Nummern, bei denen nur die letzten 4 Ziffern vorhanden sind.
CREDIT_DEBIT_CVV	Ein dreistelliger Kartenbestätigungscode (CVV), der auf Kredit- und Debitkarten von VISA und Discover vorhanden ist. MasterCard Bei American Express-Kredit- oder Debitkarten handelt es sich um einen 4-stelligen Zahlencode.
CREDIT_DEBIT_EXPIRY	Das Ablaufdatum einer Kredit- oder Debitkarte. Diese Nummer ist normalerweise vierstellig und als oder formatiert. month/year MM/YY Amazon Transcribe Kann beispielsweise Ablaufdaten wie 01/2101/2021, und Januar 2021 erkennen.
CREDIT_DEBIT_NUMBER	Die Nummer einer Kredit- oder Debitkarte. Diese Zahlen können zwischen 13 und 16 Ziffern lang sein, erkennen aber Amazon Transcribe auch Kredit- oder Debitkart ennummern, wenn nur die letzten 4 Ziffern vorhanden sind.
EMAIL	Eine E-Mail-Adresse, z. B. efua.owusu@email.com.
NAME	Der Name einer Person. Dieser Entitätstyp umfasst keine Titel wie Herr, Frau, Fräulein oder Dr.. Er wendet diesen Entitätstyp Amazon Transcribe nicht auf Namen an, die Teil von Organisationen oder Adressen sind. Amazon Transcribe Erkennt beispielsweise die John Doe Organization als Organisation und Jane Doe Street als Adresse.

PII-Typ	Description
PHONE	Eine Telefonnummer. Dieser Entitätstyp umfasst auch Fax- und Pager-Nummern.
PIN	Eine 4-stellige persönliche Identifikationsnummer (PIN), mit der jemand Zugang zu seinen Kontodaten erhält.
SSN	Eine Sozialversicherungsnummer (SSN) ist eine 9-stellige Nummer, die an US-Bürger, Personen mit ständigem Wohnsitz und Personen mit vorübergehender Erwerbstätigkeit vergeben wird. Amazon Transcribe erkennt auch Sozialversicherungsnummern, wenn nur die letzten 4 Ziffern vorhanden sind.
AGE	Das Alter einer Person, einschließlich Menge und Zeiteinheit. In der Phrase „Ich bin 40 Jahre alt“ wird beispielsweise „40 Jahre“ als Alter Amazon Transcribe erkannt.
DATE_TIME	Ein Datum kann ein Jahr, einen Monat, einen Tag, einen Wochentag oder eine Tageszeit enthalten. Amazon Transcribe Erkennt beispielsweise „19. Januar 2020“ oder „11 Uhr“ als Datum. Amazon Transcribe erkennt Teildaten, Datumsbereiche und Datumsintervalle. Es wird auch Jahrzehnte wie „die 1990er“ berücksichtigen.

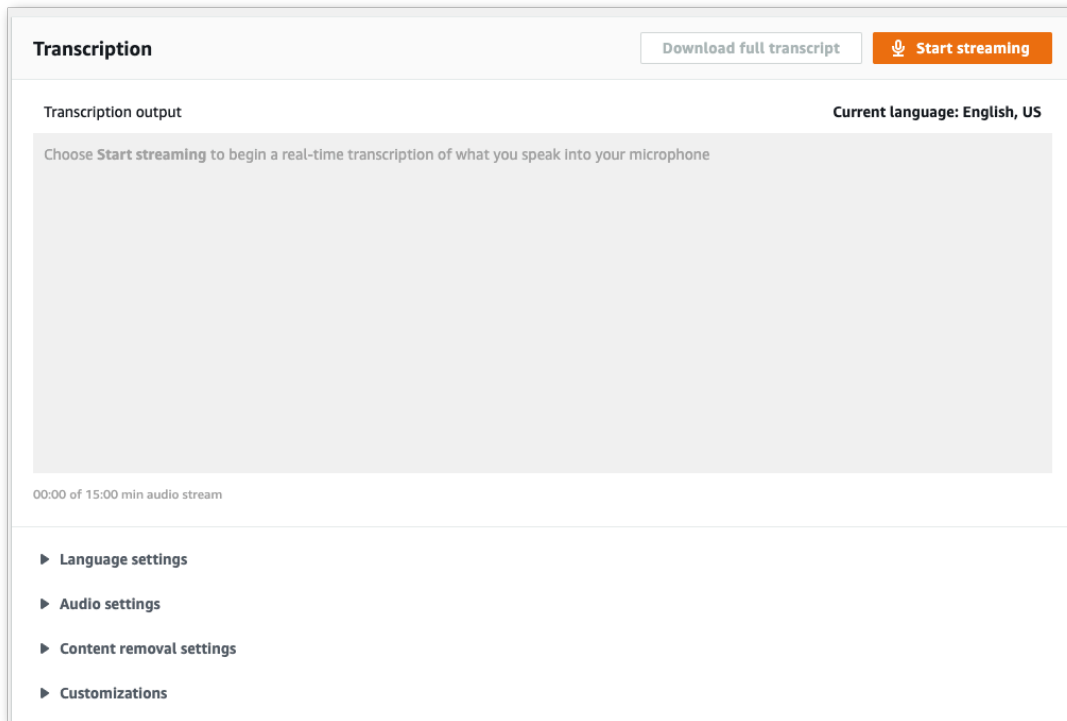
PII-Typ	Description
LICENSE_PLATE	Ein Nummernschild für ein Fahrzeug wird von dem Staat oder Land ausgestellt, in dem das Fahrzeug registriert ist. Das Format für Personenkraftwagen besteht in der Regel aus fünf bis acht Zeichen, mit Großbuchstaben gefolgt von Ziffern. Das Format variiert je nach Standort des ausstellenden Staates oder Landes.
PASSPORT_NUMBER	Eine eindeutige Kennung, die dem Reisepass einer Person zugewiesen wird. Das Format umfasst in der Regel eine Kombination aus Buchstaben und Zahlen und ist je nach Land unterschiedlich.
PASSWORD	Eine alphanumerische Zeichenfolge, die als Passwort verwendet wird, z. B. „*very20special#pass*“.
USERNAME	Ein Benutzername, der ein Konto identifiziert, wie etwa ein Login-Name, Bildschirmname, Spitzname oder Handle.
VEHICLE_IDENTIFICATION_NUMBER	Eine Fahrgestellnummern (VIN) identifiziert ein Fahrzeug eindeutig. Inhalt und Format der Fahrgestellnummer sind in der Spezifikation ISO 3779 definiert. Jedes Land hat spezifische Codes und Formate für VINs.

Sie können eine Streaming-Transkription mit AWS-Managementkonsole WebSocket, oder HTTP/2 starten.

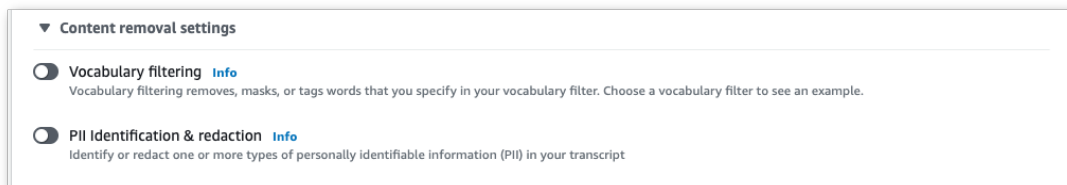
AWS-Managementkonsole

1. Melden Sie sich beim [AWS-Managementkonsole](#) an.

2. Wählen Sie im Navigationsbereich Real-time Transkription aus. Scrollen Sie nach unten zu Einstellungen für das Entfernen von Inhalten und erweitern Sie dieses Feld, falls es minimiert ist.



3. Umschalten auf PII-Identifizierung und Schwärzung.



4. Wählen Sie Nur Identifizierung oder Identifizierung und Schwärzung und wählen Sie dann die Arten von personenbezogenen Daten, die Sie in Ihrem Transkript identifizieren oder schwärzen möchten.

▼ 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)

☒ BANK_ACCOUNT_NUMBER	☒ BANK_ROUTING	☒ CREDIT_DEBIT_NUMBER
☒ CREDIT_DEBIT_CVV	☒ CREDIT_DEBIT_EXPIRY	☒ PIN

☒ Personal (8 of 8 selected)

☒ NAME	☒ ADDRESS	☒ PHONE
☒ EMAIL	☒ SSN	☒ PASSPORT_NUMBER
☒ DRIVER_ID	☒ AGE	

☒ Digital footprint (7 of 7 selected)

☒ URL	☒ USERNAME	☒ PASSWORD
☒ AWS_ACCESS_KEY	☒ AWS_SECRET_KEY	☒ IP_ADDRESS
☒ MAC_ADDRESS		

☒ Other (1 of 1 selected)

☒ DATE_TIME

► Customizations

5. Jetzt können Sie Ihren Stream transkribieren. Wählen Sie Streaming starten und beginnen Sie zu sprechen. Um Ihr Diktat zu beenden, wählen Sie Streaming beenden.

WebSocket streamen

In diesem Beispiel wird eine vorsignierte URL erstellt, die PII-Schwärzung (oder PII-Identifizierung) in einem Stream verwendet. WebSocket Für eine bessere Lesbarkeit werden Zeilenumbrüche hinzugefügt. Weitere Informationen zur Verwendung von WebSocket Streams mit finden Sie unter [Amazon Transcribe Einen WebSocket Stream einrichten](#) Weitere Einzelheiten zu den Parametern finden Sie unter [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)
```

Sie können nicht sowohl `content-identification-type` als auch `content-redaction-type` in derselben Anfrage verwenden.

Parameterdefinitionen finden Sie in der [API-Referenz](#). Parameter, die allen AWS API-Vorgängen gemeinsam sind, sind im Abschnitt [Allgemeine Parameter](#) aufgeführt.

HTTP/2 streamen

In diesem Beispiel wird eine HTTP/2 Anfrage mit aktivierter PII-Identifizierung oder PII-Schwärzung erstellt. Weitere Informationen zur Verwendung von HTTP/2 Streaming mit Amazon Transcribe finden Sie unter [Einen HTTP/2 Stream einrichten](#). Weitere Informationen zu spezifischen Parametern und Headern finden Sie Amazon Transcribe unter [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
```

Sie können nicht sowohl `content-identification-type` als auch `content-redaction-type` in derselben Anfrage verwenden.

Parameterdefinitionen finden Sie in der [API-Referenz](#). Parameter, die allen AWS API-Vorgängen gemeinsam sind, sind im Abschnitt [Allgemeine Parameter](#) aufgeführt.

 Note

Die Bearbeitung personenbezogener Daten für das Streaming wird nur in folgenden Ländern unterstützt AWS-Regionen: Asien-Pazifik (Seoul), Asien-Pazifik (Sydney), Asien-Pazifik (Tokio), Kanada (Zentral), EU (Frankfurt), EU (Irland), EU (London), USA Ost (Nord-Virginia), USA Ost (Ohio) und USA West (Oregon).

Beispiel für die Schwärzung und Identifizierung von PII

Die folgenden Beispiele zeigen geschwärzte Transkripte von Batch- und Streaming-Aufträgen sowie die PII-Identifizierung bei einem Streaming-Auftrag.

Transkriptionsaufträge, die die Inhaltsschwärzung verwenden, erzeugen zwei Arten von confidence-Werten. Bei der automatischen Spracherkennungszuverlässigkeit (Automatic Speech Recognition, ASR) werden die Elemente angegeben, die den type der pronunciation aufweisen oder bei denen die punctuation eine bestimmte Äußerung ist. In der folgenden Transkriptaussgabe hat das Wort Good einen confidence von 1.0. Dieser Konfidenzwert gibt an, dass zu 100 Prozent sicher Amazon Transcribe ist, dass das in diesem Protokoll gesprochene Wort „Gut“ ist. Der confidence-Wert für ein [PII]-Tag stellt die Zuverlässigkeit dar, dass es sich bei dem Gesprochenen, das von Amazon Transcribe zur Schwärzung gekennzeichnet wurde, wirklich um personenbezogene Daten handelt. In der folgenden Transkriptaussgabe 0.9999 bedeutet der Wert confidence von, dass zu 99,99 Prozent sicher Amazon Transcribe ist, dass es sich bei der Entität, die im Protokoll redigiert wurde, um personenbezogene Daten handelt.

Beispiel für eine geschwärzte Ausgabe (Batch)

```
{
  "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"

```

```
}
```

Hier ist das ungekürzte Transkript zum Vergleich:

```
{
  "jobName": "job id",
  "accountId": "111122223333",
  "isRedacted": false,
  "results": {
    "transcripts": [
      {
        "transcript": "Good morning, everybody. My name is Mike, and today I
feel like
my Social
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
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": [
          {
            "confidence": "0.9999",
            "content": "Mike",
```

```

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

```

Beispiel für eine geschwärzte Streaming-Ausgaben

```

{
  "TranscriptResultStream": {
    "TranscriptEvent": {
      "Transcript": {
        "Results": [
          {
            "Alternatives": [
              {
                "Transcript": "my name is [NAME]",
                "Items": [
                  {
                    "Content": "my",
                    "EndTime": 0.3799375,
                    "StartTime": 0.0299375,
                    "Type": "pronunciation"
                  },
                  {
                    "Content": "name",
                    "EndTime": 0.5899375,
                    "StartTime": 0.3899375,
                    "Type": "pronunciation"
                  },
                  {
                    "Content": "is",
                    "EndTime": 0.7899375,
                    "StartTime": 0.5999375,
                    "Type": "pronunciation"
                  },
                  {
                    "Content": "[NAME]",
                    "EndTime": 1.0199375,

```

```

        "StartTime": 0.7999375,
        "Type": "pronunciation"
    },
],
"Entities": [
    {
        "Content": "[NAME]",
        "Category": "PII",
        "Type": "NAME",
        "StartTime" : 0.7999375,
        "EndTime" : 1.0199375,
        "Confidence": 0.9989
    }
]
}
},
"EndTime": 1.02,
"IsPartial": false,
"ResultId": "12345a67-8bc9-0de1-2f34-a5b678c90d12",
"StartTime": 0.0199375
}
]
}
}
}
```

Beispiel für die Ausgabe der PII-Identifikation

Die PII-Identifikation ist eine zusätzliche Feature, die Sie bei Ihrem Streaming-Transkriptionsauftrag nutzen können. Die identifizierten PII sind im **Entities**-Abschnitt des jeweiligen Segments aufgeführt.

```
{
  "TranscriptResultStream": {
    "TranscriptEvent": {
      "Transcript": {
        "Results": [
          {
            "Alternatives": [
              {
                "Transcript": "my name is mike",
                "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": "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
}
]

```



```
}  
  }  
    }  
      }
```

Erstellen von Videountertiteln

Amazon Transcribe unterstützt die Ausgabe von WebVTT (*.vtt) und SubRip (*.srt) zur Verwendung als Videountertitel. Sie können einen oder beide Dateitypen auswählen, wenn Sie Ihren Batch-Videotranskriptionsauftrag einrichten. Bei Verwendung des Features „Untertitel“ werden die von Ihnen ausgewählte(n) Untertiteldatei(en) und eine reguläre Transkriptdatei (mit zusätzlichen Informationen) erstellt. Untertitel- und Transkriptionsdateien werden an dasselbe Ziel ausgegeben.

Die Untertitel werden gleichzeitig mit dem gesprochenen Text angezeigt und bleiben sichtbar, bis eine natürliche Pause entsteht oder der Sprecher zu Ende spricht. Beachten Sie, dass keine Untertiteldatei erstellt wird, wenn Sie Untertitel in Ihrer Transkriptionsanforderung aktivieren und Ihr Audio keine Sprache enthält.

Important

Amazon Transcribe verwendet einen Standard-Startindex von 0 für die Untertitelausgabe, der sich vom häufiger verwendeten Wert von 1 unterscheidet. Wenn Sie einen Startindex von 1 benötigen, können Sie dies in der AWS-Managementkonsole oder in Ihrer API-Anfrage mithilfe des [OutputStartIndex](#) Parameters angeben.

Die Verwendung des falschen Startindexes kann zu Kompatibilitätsfehlern mit anderen Services führen. Stellen Sie daher sicher, dass Sie vor der Erstellung Ihrer Untertitel überprüfen, welchen Startindex Sie benötigen. Wenn Sie sich nicht sicher sind, welchen Wert Sie verwenden sollen, empfehlen wir Ihnen, 1 zu wählen. Weitere Informationen finden Sie unter [Subtitles](#).

Unterstützte Features mit Untertiteln:

- Schwärzung von Inhalten – Alle geschwärzten Inhalte werden sowohl in den Untertitel- als auch in den regulären Transkriptausgabedateien als „PII“ angezeigt. Audio wird nicht verändert.
- Wortschatzfilter – Untertiteldateien werden aus der Transkriptionsdatei generiert, sodass alle Wörter, die Sie in Ihrer Standardtranskriptionsausgabe filtern, auch in Ihren Untertiteln gefiltert werden. Gefilterte Inhalte werden als Leerraum oder *** in Ihren Transkript- und Untertiteldateien angezeigt. Audio wird nicht verändert.
- Sprecherdiarisierung – Wenn es in einem Untertitelsegment mehrere Sprecher gibt, werden Bindestriche zur Unterscheidung der einzelnen Sprecher verwendet. Dies gilt sowohl für WebVTT als auch für SubRip Formate; zum Beispiel:

- -- Text gesprochen von Person 1
- -- Text gesprochen von Person 2

Untertiteldateien werden am selben Amazon S3 Ort wie Ihre Transkriptionsausgabe gespeichert.

Siehe [Amazon Transcribe Video Snacks: Erstellen von Videountertiteln, ohne Code zu schreiben](#), für eine Videoanleitung zur Erstellung von Untertiteln.

Erzeugen von Untertiteldateien

Sie können Untertiteldateien mit den SDKs AWS-Managementkonsole, AWS CLI oder AWS erstellen. Siehe die folgenden Beispiele:

AWS-Managementkonsole

1. Melden Sie sich an der [AWS-Managementkonsole](#) an.
2. Wählen Sie im Navigationsbereich Transkriptionsaufträge und dann Auftrag erstellen (oben rechts). Dies öffnet die Seite Auftragsdetails angeben. Die Optionen für Untertitel befinden sich im Bereich Ausgabedaten.
3. Wählen Sie die Formate aus, die Sie für Ihre Untertiteldateien wünschen, und wählen Sie dann einen Wert für den Startindex. Beachten Sie, dass dies die Amazon Transcribe Standardeinstellung ist 0, aber häufiger verwendet 1 wird. Wenn Sie sich nicht sicher sind, welchen Wert Sie verwenden sollen, empfehlen wir Ihnen, 1 zu wählen, da dies die Kompatibilität mit anderen Services verbessern kann.

The screenshot shows the 'Output data' configuration section in the AWS Management Console. It contains three main settings:

- Output data location type info** (with an 'Info' link):
 - ☒ **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** (with an 'Info' link):
 - ☒ **SRT (SubRip)**
 - ☐ **VTT (WebVTT)**
- Specify the start index**: A dropdown menu with the value '0' selected.

4. Füllen Sie alle anderen Felder aus, die Sie auf der Seite Auftragsdetails angeben möchten, und wählen Sie dann Weiter. Hier gelangen Sie zur Seite Auftrag konfigurieren – optional .
5. Wählen Sie Auftrag erstellen, um Ihren Transkriptionsauftrag auszuführen.

AWS CLI

In diesem Beispiel werden der Befehl [start-Transkription-job](#) und der Parameter Subtitles verwendet. Weitere Informationen erhalten Sie unter [StartTranscriptionJob](#) und [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
```

Hier ein weiteres Beispiel mit dem Befehl [start-transcription-job](#) und einem Anforderungstext, der Untertitel zu diesem Auftrag hinzufügt.

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

Die Datei *my-first-subtitle-job.json* enthält den folgenden Anforderungstext.

```
{  
  "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  
  }  
}
```

```
}
}
```

AWS SDK für Python (Boto3)

In diesem Beispiel werden mithilfe des `Subtitles` Arguments für AWS SDK für Python (Boto3) die Methode `start_transcription_job` Untertitel hinzugefügt. Weitere Informationen erhalten Sie unter [StartTranscriptionJob](#) und [Subtitles](#).

Weitere Beispiele für die Verwendung der AWS SDKs, einschließlich funktionspezifischer, szenarienspezifischer und serviceübergreifender Beispiele, finden Sie im Kapitel. [Codebeispiele für Amazon Transcribe mit AWS SDKs](#)

```
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)
```

Analyse von Callcenter-Audio mit Call Analytics

Verwenden Sie Amazon Transcribe Call Analytics, um Einblicke in die Interaktionen zwischen Kunden und Mitarbeitern zu erhalten. Call Analytics ist speziell für Callcenter-Audio konzipiert und liefert Ihnen automatisch wertvolle Daten zu jedem Anruf und jedem Teilnehmer. Sie können auch Daten zu bestimmten Zeitpunkten während des Anrufs einschränken. Sie können zum Beispiel die Stimmung der Kunden in den ersten Sekunden eines Anrufs mit dem letzten Viertel des Anrufs vergleichen, um festzustellen, ob Ihr Kundendienstmitarbeiter eine positive Erfahrung vermittelt hat. Weitere Beispiele für Anwendungsfälle sind im [folgenden Abschnitt](#) aufgeführt.

Call Analytics ist für Transkriptionen nach dem Gespräch und Echtzeit-Transkriptionen verfügbar. Wenn Sie eine Datei transkribieren, die sich in einem Amazon S3 Bucket befindet, führen Sie eine Transkription nach dem Anruf durch. Wenn Sie einen Audiostream transkribieren, führen Sie eine Echtzeittranskription durch. Diese beiden Transkriptionsmethoden bieten unterschiedliche Einblicke und Features von Call Analytics. Weitere Einzelheiten zu den einzelnen Methoden finden Sie unter [Post-call Analytik](#) und [Real-time Rufen Sie Analytics auf](#).

Mit den Echtzeit-Transkriptionen von Call Analytics können Sie auch [Analyse nach Anrufen](#) in Ihre Anfrage aufnehmen. Ihr Analytics-Transkript nach dem Anruf wird in dem Amazon S3 Bucket gespeichert, den Sie in Ihrer Anfrage angeben. Weitere Informationen finden Sie unter [Post-call Analytik mit Transkriptionen in Echtzeit](#).

API-Vorgänge speziell für Call Analytics

Post-call: [CreateCallAnalyticsCategory](#), [DeleteCallAnalyticsCategory](#), [DeleteCallAnalyticsJob](#), [GetCallAnalyticsCategory](#), [GetCallAnalyticsJob](#), [ListCallAnalyticsCategories](#), [ListCallAnalyticsJobs](#), [StartCallAnalyticsJob](#), [UpdateCallAnalyticsCategory](#)

Real-time: [StartCallAnalyticsStreamTranscription](#),
[StartCallAnalyticsStreamTranscriptionWebSocket](#)

Häufige Anwendungsfälle

Post-call Transkriptionen:

- Überwachen Sie die Häufigkeit von Problemen im Laufe der Zeit: Verwenden [Sie die](#) Anrufr kategorisierung, um wiederkehrende Schlüsselwörter in Ihren Transkripten zu identifizieren.

- Gewinnen Sie Einblicke in Ihren Kundenservice: Nutzen Sie [Anrufmerkmale](#) (Nichtgesprächszeit, Gesprächszeit, Unterbrechungen, Sprachlautstärke, Gesprächsgeschwindigkeit) und Stimmungsanalysen, um festzustellen, ob Kundenprobleme während des Anrufs angemessen gelöst werden.
- Gewährleisten Sie die Einhaltung gesetzlicher Vorschriften oder von Unternehmensrichtlinien: Legen Sie [Schlüsselwörter und Formulierungen](#) für unternehmensspezifische Begrüßungen oder Haftungsausschlüsse fest, um zu überprüfen, ob Ihre Kundendienstmitarbeiter die gesetzlichen Vorschriften einhalten.
- Verbessern Sie den Umgang mit den personenbezogenen Daten Ihrer Kunden: Verwenden Sie die [PII-Schwärzung](#) in Ihrer Transkriptionsausgabe oder Audiodatei, um die Privatsphäre der Kunden zu schützen.
- Verbesserung der Mitarbeiterschulung: Verwenden Sie Kriterien (Stimmung, Nicht-Gesprächszeit, Unterbrechungen, Gesprächsgeschwindigkeit), um Transkripte zu kennzeichnen, die als Beispiele für positive oder negative Kundeninteraktionen verwendet werden können.
- Messen Sie die Effektivität Ihrer Kundendienstmitarbeiter bei der Schaffung eines positiven Kundenerlebnisses: Verwenden Sie [Stimmungsanalysen](#), um zu messen, ob Ihre Kundendienstmitarbeiter in der Lage sind, eine negative Kundenstimmung im Laufe des Gesprächs in eine positive zu verwandeln.
- Verbessern Sie die Datenorganisation: Beschriften und sortieren Sie Anrufe nach [benutzerdefinierten Kategorien](#) (einschließlich Schlüsselwörtern und Formulierungen, Stimmung, Gesprächszeit und Unterbrechungen).
- Fassen Sie die wichtigen Aspekte eines Anrufs mithilfe von generativer KI zusammen: Verwenden Sie die [generative Anrufzusammenfassung](#), um eine kurze Zusammenfassung des Transkripts zu erhalten, die wichtige Komponenten wie die bei dem Anruf besprochenen Probleme, Maßnahmen und Ergebnisse enthält.

Real-time Transkriptionen:

- Eskalationen in Echtzeit abschwächen: Richten Sie [Echtzeitwarnungen](#) für wichtige Formulierungen ein – z. B. wenn ein Kunde sagt: „Mit einem Vorgesetzten sprechen“ –, um Anrufe zu kennzeichnen, wenn sie zu eskalieren beginnen. Sie können Echtzeit-Warnungen mit Echtzeit-Kategorieübereinstimmungen erstellen.
- Verbessern Sie den Umgang mit Kundendaten: Verwenden Sie die [PII-Kennzeichnung](#) oder [PII-Schwärzung](#) in Ihrer Transkriptionsausgabe, um die Privatsphäre der Kunden zu schützen.

- Identifizieren Sie benutzerdefinierte Schlüsselwörter und Formulierungen: Verwenden Sie [benutzerdefinierte Kategorien](#), um bestimmte Schlüsselwörter in einem Anruf zu kennzeichnen.
- Probleme automatisch erkennen: Nutzen Sie die automatische [Erkennung von Problemen](#), um eine kurze Zusammenfassung aller in einem Anruf identifizierten Probleme zu erhalten.
- Messen Sie die Effektivität Ihrer Kundendienstmitarbeiter bei der Schaffung eines positiven Kundenerlebnisses: Verwenden Sie [Stimmungsanalysen](#), um zu messen, ob Ihre Kundendienstmitarbeiter in der Lage sind, eine negative Kundenstimmung im Laufe des Gesprächs in eine positive zu verwandeln.
- Agent-Assist einrichten: Nutzen Sie die Erkenntnisse Ihrer Wahl, um Ihren Kundendienstmitarbeitern proaktive Unterstützung bei der Lösung von Kundenanrufen zu bieten. Weitere Informationen finden Sie unter [Analyse von Live-Anrufen und Unterstützung für Kundendienstmitarbeiter für Ihr Contact Center mit Amazon Language AI Services](#).

Informationen zum Vergleich der in Call Analytics verfügbaren Funktionen mit denen von Amazon Transcribe und Amazon Transcribe Medical finden Sie in der [Funktionstabelle](#).

Die ersten Schritte finden Sie unter [Beginn einer analytischen Nach Anrufen-Transkription](#) und [Start einer Call Analytics in Echtzeit](#). Die Ausgabe von Call Analytics ähnelt der eines normalen Transkriptionsauftrags, enthält jedoch zusätzliche Analysedaten. Beispiele für die Ausgabe finden Sie unter [Post-call Ergebnisse aus Analysen](#) und [Real-time Rufen Sie die Analytics-Ausgabe auf](#).

Überlegungen und zusätzliche Informationen

Bevor Sie Call Analytics verwenden, sollten Sie Folgendes beachten:

- Call Analytics unterstützt nur Zwei-Kanal-Audio, bei dem ein Kundendienstmitarbeiter auf einem Kanal und ein Kunde auf einem zweiten Kanal präsent ist.
- [Auftragswarteschlangen](#) ist immer für Aufträge für Analyse nach Anrufen aktiviert, sodass Sie auf 100 gleichzeitige Call-Analytics-Aufträge beschränkt sind. Wenn Sie eine Quotaerhöhung beantragen möchten, siehe [AWS service quotas](#).
- Die Eingabedateien für Aufträge für Analyse nach Anrufen dürfen nicht größer als 500 MB sein und müssen weniger als 4 Stunden umfassen. Bei Audiodateien mit einer Frequenz von mehr als 8 kHz muss die Audiodauer weniger als oder gleich 2 Stunden sein. Beachten Sie, dass die Dateigrößenbeschränkung für bestimmte komprimierte Audiodateiformate, die nicht WAV sind, geringer sein kann.

- Wenn Sie Kategorien verwenden, müssen Sie alle gewünschten Kategorien erstellen, bevor Sie eine Call-Analytics-Übertragung starten. Neue Kategorien können nicht auf bestehende Transkriptionen angewendet werden. Wie Sie eine neue Kategorie erstellen können, erfahren Sie unter [Erstellen von Kategorien für Transkription nach Anrufen](#) und [Kategorien für Echtzeit-Transkriptionen erstellen](#).
- Einige Call Analytics-Kontingente unterscheiden sich von Amazon Transcribe denen für Amazon Transcribe medizinische Daten. Einzelheiten finden Sie in der [AWS allgemeinen Referenz](#).

 Tauchen Sie tiefer ein mit AWS Blog zum Machine Learning

Weitere Informationen zu den Call-Analytics-Optionen finden Sie unter:

- [Analyse nach Anrufen für Ihr Contact Center mit Amazon Language AI Services](#)
- [Analyse von Live-Anrufen und Unterstützung für Kundendienstmitarbeiter für Ihr Contact Center mit Amazon Language AI Services](#)

Beispiele für die Ausgabe und Funktionen von Call Analytics finden Sie in unserer [GitHubDemo](#). Wir bieten auch eine [JSON-zu-Word-Dokumentenwendung](#), mit der Sie Ihr Transkript in ein leicht lesbares Format konvertieren können.

Regionale Verfügbarkeit und Quotas

Call Analytics wird in den folgenden Bereichen unterstützt AWS-Regionen:

Region	Transkriptionstyp
ap-northeast-1 (Tokyo)	post-call, real-time
ap-northeast-2 (Seoul)	post-call, real-time
ap-south-1 (Mumbai)	post-call
ap-southeast-1 (Singapur)	post-call
ap-southeast-2 (Sydney)	post-call, real-time
ca-central-1 (Kanada, Zentral)	post-call, real-time

Region	Transkriptionstyp
eu-central-1 (Frankfurt)	post-call, real-time
eu-west-2 (London)	post-call, real-time
us-east-1 (N. Virginia)	post-call, real-time
us-west-2 (Oregon)	post-call, real-time

Beachten Sie, dass die Unterstützung der Regionen unterschiedlich ist für [Amazon Transcribe](#), [Amazon Transcribe Medical](#) und Call Analytics.

Um die Endpunkte für jede unterstützte Region zu erhalten, siehe [Dienstendpunkte](#) in der Allgemeinen AWS -Referenz.

Eine Liste der Quotas, die sich auf Ihre Transkriptionen beziehen, finden Sie in der Allgemeinen AWS -Referenz unter [Service Quotas](#). Einige Quotas können auf Anfrage geändert werden. Steht in der Spalte Anpassbar Ja, können Sie eine Erhöhung beantragen. Wählen Sie dazu den angegebenen Link.

Post-call Analytik

Call Analytics bietet Analysen nach Anrufen, die für die Überwachung von Trends im Kundenservice nützlich sind.

Post-call Transkriptionen bieten die folgenden Einblicke:

- [Gesprächsmerkmale](#), einschließlich Gesprächszeit, Nicht-Gesprächszeit, Lautstärke des Sprechers, Unterbrechungen, Gesprächsgeschwindigkeit, Problemen, Ergebnissen und Maßnahmen
- [Generative Anrufzusammenfassung](#), die eine kurze Zusammenfassung des gesamten Anrufs erstellt
- [Benutzerdefinierte Kategorisierung](#) mit Regeln, die Sie verwenden können, um auf bestimmte Schlüsselwörter und Kriterien einzugehen
- [PII-Schwärzung](#) Ihres Texttranskripts und Ihrer Audiodatei
- [Sprecherstimmung](#) für jeden Anrufer zu verschiedenen Zeitpunkten eines Anrufs

Post-call Einblicke

In diesem Abschnitt werden die für die Transkription von Analyse nach Anrufen verfügbaren Erkenntnisse erläutert.

Anrufmerkmale

Das Feature „Anrufmerkmale“ misst die Qualität der Interaktion zwischen Kundendienstmitarbeiter und Kunde anhand dieser Kriterien:

- **Unterbrechung:** Misst, ob und wann ein Teilnehmer den anderen Teilnehmer mitten im Satz unterbricht. Häufige Unterbrechungen können mit Unhöflichkeit oder Verärgerung verbunden sein und könnten mit einer negativen Stimmung für einen oder beide Teilnehmer korrelieren.
- **Lautstärke:** Misst die Lautstärke, mit der jeder Teilnehmer spricht. Anhand dieser Kennzahl können Sie feststellen, ob der Anrufer oder der Kundendienstmitarbeiter laut spricht oder schreit, was oft ein Zeichen für Verärgerung ist. Diese Metrik wird als normalisierter Wert (Sprachpegel pro Sekunde der Sprache in einem bestimmten Segment) auf einer Skala von 0 bis 100 dargestellt, wobei ein höherer Wert eine lautere Stimme anzeigt.
- **Non-talk Zeit:** Misst Zeiträume, in denen keine Sprache vorkommt. Verwenden Sie diese Kennzahl, um zu sehen, ob es lange Schweigephasen gibt, z. B. wenn ein Kundendienstmitarbeiter einen Kunden übermäßig lange in der Warteschleife hält.
- **Sprechgeschwindigkeit:** Misst die Geschwindigkeit, mit der beide Teilnehmer sprechen. Die Verständlichkeit kann beeinträchtigt werden, wenn ein Teilnehmer zu schnell spricht. Diese Kennzahl wird in Wörtern pro Minute gemessen.
- **Gesprächszeit:** Misst die Zeit (in Millisekunden), die jeder Teilnehmer während des Gesprächs gesprochen hat. Anhand dieser Metrik können Sie feststellen, ob ein Teilnehmer das Gespräch dominiert oder ob der Dialog ausgewogen ist.
- **Probleme, Ergebnisse und Maßnahmen:** Identifiziert Probleme, Ergebnisse und Maßnahmen aus dem Anruftranskript

Hier ist ein [Beispiel für die Ausgabe](#).

Generative Anrufzusammenfassung

Bei der generativen Anrufzusammenfassung wird eine präzise Zusammenfassung des gesamten Anrufs erstellt, in der wichtige Komponenten wie der Grund für den Anruf, die zur Problemlösung unternommenen Schritte und die nächsten Schritte erfasst werden.

Die generative Anruferzusammenfassung bietet Ihnen folgende Vorteile:

- Geringere Notwendigkeit von manuellen Notizen während und nach Anrufen
- Verbesserung der Effizienz der Kundendienstmitarbeiter, da diesen mehr Zeit für Gespräche mit den in der Warteschlange wartenden Anrufern bleibt und sie sich weniger um die Nachbearbeitung des Anrufs kümmern müssen
- Schnellere Überprüfungen durch Supervisoren, da Zusammenfassungen von Anrufen viel schneller geprüft werden können als komplette Transkripte

Informationen zur Verwendung der generativen Anruferzusammenfassung mit einem Auftrag zur Analyse nach dem Anruf finden Sie unter [Aktivieren der generativen Anruferzusammenfassung](#). Ein Beispiel für die Ausgabe finden Sie unter [Generative Anruferzusammenfassung – Beispielausgabe](#). Die generative Anruferzusammenfassung wird separat in Rechnung gestellt. (Weitere Informationen finden Sie auf der [Seite mit den Preisen](#).)

Note

Die generative Anruferzusammenfassung ist derzeit in us-east-1 und us-west-2 verfügbar. Diese Funktion wird in den folgenden englischen Dialekten unterstützt: Australisch (en-AU), Britisch (en-GB), Indisch (en-IN), Irisch (en-IE), Schottisch (en-AB), Amerikanisch (en-US) und Walisisch (en-WL).

Benutzerdefinierte Kategorisierung

Verwenden Sie die Anruferkategorisierung, um Schlüsselwörter, Formulierungen, Stimmungen oder Aktionen innerhalb eines Anrufs zu markieren. Unsere Kategorisierungsoptionen helfen Ihnen bei der Triage von Eskalationen, z. B. bei Anrufen mit negativer Stimmung und vielen Unterbrechungen, oder bei der Einteilung von Anrufen in bestimmte Kategorien, z. B. nach Unternehmensabteilungen.

Zu den Kriterien, die Sie einer Kategorie hinzufügen können, gehören:

- Non-talk Zeit: Zeiträume, in denen weder der Kunde noch der Mitarbeiter sprechen.
- Unterbrechungen: Wenn der Kunde oder der Kundendienstmitarbeiter die andere Person unterbricht.
- Kunden- oder Kundendienstmitarbeiterstimmung: Wie sich der Kunde oder der Kundendienstmitarbeiter während eines bestimmten Zeitraums fühlt. Wenn mindestens 50

Prozent der Gesprächsrunden (das Hin und Her zwischen zwei Sprechern) in einem bestimmten Zeitraum der angegebenen Stimmung entsprechen, wird die Stimmung als Amazon Transcribe übereinstimmend betrachtet.

- Schlüsselwörter oder Formulierungen: Entspricht einem Teil der Transkription auf der Grundlage einer exakten Formulierung. Wenn Sie z. B. einen Filter für den Satz „Ich möchte mit einem Vorgesetzten sprechen“ setzen, filtert Amazon Transcribe genau nach dieser Formulierung.

Sie können auch die Umkehrung der vorherigen Kriterien kennzeichnen (Gesprächszeit, fehlende Unterbrechungen, fehlende Stimmung und das Fehlen eines bestimmten Satzes).

Hier ist ein [Beispiel für die Ausgabe](#).

Weitere Informationen zu Kategorien oder wie Sie eine neue Kategorie erstellen können, finden Sie unter [Erstellen von Kategorien für Transkription nach Anrufen](#).

Schwärzung sensibler Daten

Durch die Schwärzung sensibler Daten werden persönlich identifizierbare Informationen (PII) im Texttranskript und der Audiodatei ersetzt. Ein geschwärztes Transkript ersetzt den Originaltext durch [PII]; eine geschwärzte Audiodatei ersetzt gesprochene persönliche Informationen durch Stille. Dieser Parameter ist nützlich für den Schutz von Kundeninformationen.

Note

Post-call Die Bearbeitung personenbezogener Daten wird in US-Englisch (en-US) und US-Spanisch (es-US) unterstützt.

Um die Liste der mit diesem Feature geschwärzten personenbezogenen Daten anzuzeigen oder mehr über die Schwärzung mit Amazon Transcribe zu erfahren, besuchen Sie [Schwärzung oder Identifizierung persönlich identifizierbarer Informationen](#).

Hier ist ein [Beispiel für eine Ausgabe](#).

Stimmungsanalyse

Die Stimmungsanalyse schätzt ein, wie sich der Kunde und der Kundendienstmitarbeiter während des Gesprächs fühlen. Diese Metrik wird sowohl als quantitativer Wert (mit einem Bereich von 5

bis -5) als auch als qualitativer Wert (positive, neutral, mixed oder negative) dargestellt. Quantitative Werte werden pro Quartal und pro Anruf angegeben; qualitative Werte werden pro Runde angegeben.

Anhand dieser Kennzahl lässt sich feststellen, ob Ihr Kundendienstmitarbeiter in der Lage ist, einen verärgerten Kunden bis zum Ende des Anrufs zufrieden zu stellen.

Die Stimmungsanalyse ist sofort verfügbar und unterstützt daher keine Anpassung, z. B. Modelltraining oder benutzerdefinierte Kategorien.

Hier ist ein [Beispiel für die Ausgabe](#).

Erstellen von Kategorien für Transkription nach Anrufen

Post-call Analytics unterstützt die Erstellung benutzerdefinierter Kategorien, sodass Sie Ihre Transkriptanalysen optimal an Ihre spezifischen Geschäftsanforderungen anpassen können.

Sie können so viele Kategorien erstellen, wie Sie möchten, um eine Reihe verschiedener Szenarien abzudecken. Für jede Kategorie, die Sie anlegen, müssen Sie zwischen 1 und 20 Regeln erstellen. Jede Regel basiert auf einem von vier Kriterien: Unterbrechungen, Schlüsselwörter, Nicht-Gesprächszeit oder Gefühlslage. Weitere Informationen zur Verwendung dieser Kriterien für den [CreateCallAnalyticsCategory](#)-Vorgang finden Sie im [Regelkriterien für Analysekatgorien nach Anrufen](#)-Abschnitt.

Wenn der Inhalt Ihrer Medien allen Regeln entspricht, die Sie in einer bestimmten Kategorie angegeben haben, kennzeichnet Amazon Transcribe Ihre Ausgabe mit dieser Kategorie. Ein Beispiel für eine Kategorieübereinstimmung in der JSON-Ausgabe finden Sie unter [Ausgabe der Anrufrkategorisierung](#).

Hier sind einige Beispiele dafür, was Sie mit benutzerdefinierten Kategorien machen können:

- Isolierung von Anrufen mit bestimmten Merkmalen, z. B. von Anrufen, die mit einem negativen Kundenfeedback enden
- Erkennen Sie Trends bei Kundenproblemen, indem Sie bestimmte Schlüsselwörter markieren und verfolgen
- Überwachung der Einhaltung von Vorschriften, z. B. wenn ein Kundendienstmitarbeiter in den ersten Sekunden eines Anrufs einen bestimmten Satz spricht (oder auslöst)
- Gewinnen Sie Einblicke in die Kundenerfahrung, indem Sie Anrufe mit vielen Unterbrechungen durch Kundendienstmitarbeiter und negativer Kundenstimmung markieren

- Vergleichen Sie mehrere Kategorien, um Korrelationen zu messen, z. B. um zu analysieren, ob ein Kundendienstmitarbeiter, der eine Begrüßungsfloskel verwendet, mit einer positiven Kundenstimmung korreliert

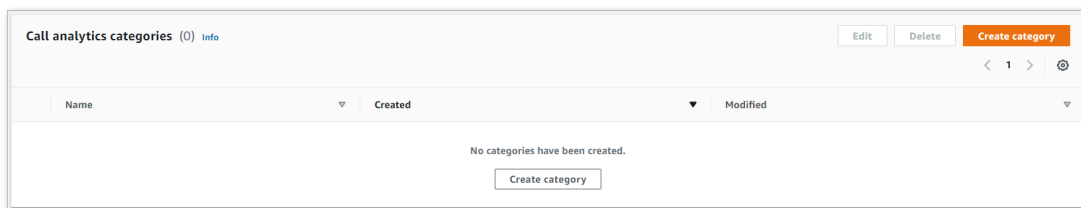
Post-call im Vergleich zu Kategorien in Echtzeit

Bei der Erstellung einer neuen Kategorie können Sie angeben, ob sie als Nach Anrufen-Analyse-Kategorie (POST_CALL) oder als Echtzeit-Call-Analytics-Kategorie (REAL_TIME) erstellt werden soll. Wenn Sie keine Option angeben, wird Ihre Kategorie standardmäßig als Kategorie nach dem Anruf erstellt. Post-call Übereinstimmungen mit den Analytics-Kategorien sind in Ihrer Ausgabe verfügbar, sobald Ihre Analytics-Transkription nach dem Anruf abgeschlossen ist.

Um eine neue Kategorie für die Analyse nach Anrufen zu erstellen, können Sie die SDKs AWS AWS-Managementkonsole oder AWS CLI verwenden. Beispiele finden Sie im Folgenden:

AWS-Managementkonsole

1. Wählen Sie im Navigationsbereich unter Amazon Transcribe Call Amazon Transcribe Analytics aus.
2. Wählen Sie Anruf-Analytics-Kategorien, um zur Seite Anruf-Analytics-Kategorien zu gelangen. Wählen Sie Kategorie erstellen.



3. Sie befinden sich jetzt auf der Seite „Kategorie erstellen“. Geben Sie einen Namen für Ihre Kategorie ein und wählen Sie dann im Dropdown-Menü Kategorietyp die Option „Batch-Call-Analytics“.

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. Sie können eine Vorlage wählen, um Ihre Kategorie zu erstellen, oder Sie können eine neue erstellen.

Wenn Sie eine Vorlage verwenden: Wählen Sie Vorlage verwenden (empfohlen), wählen Sie die gewünschte Vorlage und wählen Sie dann Kategorie erstellen.

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. Wenn Sie eine benutzerdefinierte Kategorie erstellen: Wählen Sie Von Grund auf neu erstellen.

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. Fügen Sie über das Dropdown-Menü Regeln zu Ihrer Kategorie hinzu. Sie können bis zu 20 Regeln pro Kategorie hinzufügen.

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 ▲

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. Hier ein Beispiel für eine Kategorie mit zwei Regeln: ein Kundendienstmitarbeiter, der einen Kunden während des Gesprächs länger als 15 Sekunden unterbricht, und eine negative Stimmung, die der Kunde oder der Kundendienstmitarbeiter in den letzten zwei Minuten des Gesprächs empfindet.

Rules

All the rule conditions must be met for a transcription job to be classified in this category.

▼ Rule 1

When the duration of the interruption was more than 15 second(s) during the entire call when the speaker was agent.

Delete rule

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 15 second(s) ▼

during the entire call ▼

when the speaker was agent ▼

AND

▼ Rule 2

When the sentiment is negative during the last 2 minute(s) when the speaker was either.

Delete rule

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 negative ▼

during the last 2 minute(s) ▼

when the speaker was either ▼

Add rule

You can add up to 18 more rules.

8. Wenn Sie alle Regeln zu Ihrer Kategorie hinzugefügt haben, wählen Sie Kategorie erstellen.

AWS CLI

In diesem Beispiel wird der Befehl [create-call-analytics-category](#) verwendet. Weitere Informationen finden Sie unter [CreateCallAnalyticsCategory](#), [CategoryProperties](#) und [Rule](#).

Im folgenden Beispiel wird eine Kategorie mit den Regeln erstellt:

- Der Kunde wurde in den ersten 60.000 Millisekunden unterbrochen. Die Dauer dieser Unterbrechungen betrug mindestens 10.000 Millisekunden.
- Zwischen 10 % und 80 % des Anrufs herrschte eine Stille von mindestens 20.000 Millisekunden.
- Der Kundendienstmitarbeiter hatte zu irgendeinem Zeitpunkt des Gesprächs eine negative Stimmung.

- Die Worte „Willkommen“ oder „Hallo“ wurden in den ersten 10.000 Millisekunden des Anrufs nicht verwendet.

In diesem Beispiel wird der Befehl [create-call-analytics-category](#) verwendet und ein Anforderungskörper, der mehrere Regeln zu Ihrer Kategorie hinzufügt.

```
aws transcribe create-call-analytics-category \  
--cli-input-json file://filepath/my-first-analytics-category.json
```

Die Datei `my-first-analytics-category.json` enthält den folgenden Anforderungstext.

```
{  
  "CategoryName": "my-new-category",  
  "InputType": "POST_CALL",  
  "Rules": [  
    {  
      "InterruptionFilter": {  
        "AbsoluteTimeRange": {  
          "First": 60000  
        },  
        "Negate": false,  
        "ParticipantRole": "CUSTOMER",  
        "Threshold": 10000  
      }  
    },  
    {  
      "NonTalkTimeFilter": {  
        "Negate": false,  
        "RelativeTimeRange": {  
          "EndPercentage": 80,  
          "StartPercentage": 10  
        },  
        "Threshold": 20000  
      }  
    },  
    {  
      "SentimentFilter": {  
        "ParticipantRole": "AGENT",  
        "Sentiments": [  
          "NEGATIVE"  
        ]  
      }  
    }  
  ]  
}
```

```

    },
    {
        "TranscriptFilter": {
            "Negate": true,
            "AbsoluteTimeRange": {
                "First": 10000
            },
            "Targets": [
                "welcome",
                "hello"
            ],
            "TranscriptFilterType": "EXACT"
        }
    }
]
}

```

AWS SDK für Python (Boto3)

In diesem Beispiel wird mithilfe der AWS SDK für Python (Boto3) Rules Argumente `CategoryName` und für die Methode [create_call_analytics_category](#) eine Kategorie erstellt. Weitere Informationen finden Sie unter [CreateCallAnalyticsCategory](#), [CategoryProperties](#) und [Rule](#).

Weitere Beispiele für die Verwendung der AWS SDKs, einschließlich funktionspezifischer, szenarienspezifischer und serviceübergreifender Beispiele, finden Sie im Kapitel. [Codebeispiele für Amazon Transcribe mit AWS SDKs](#)

Im folgenden Beispiel wird eine Kategorie mit den Regeln erstellt:

- Der Kunde wurde in den ersten 60.000 Millisekunden unterbrochen. Die Dauer dieser Unterbrechungen betrug mindestens 10.000 Millisekunden.
- Zwischen 10 % und 80 % des Anrufs herrschte eine Stille von mindestens 20.000 Millisekunden.
- Der Kundendienstmitarbeiter hatte zu irgendeinem Zeitpunkt des Gesprächs eine negative Stimmung.
- Die Worte „Willkommen“ oder „Hallo“ wurden in den ersten 10.000 Millisekunden des Anrufs nicht verwendet.

```

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)

```

Regelkriterien für Analysekategorien nach Anrufen

In diesem Abschnitt werden die Arten von benutzerdefinierten POST_CALL Regeln beschrieben, die Sie mit der [CreateCallAnalyticsCategory](#)-API-Operation erstellen können.

Unterbrechungsabgleich

Regeln mit Unterbrechungen ([InterruptionFilter](#)-Datentyp) sind auf Übereinstimmung ausgelegt:

- Fälle, in denen ein Kundendienstmitarbeiter einen Kunden unterbricht
- Fälle, in denen ein Kunde einen Kundendienstmitarbeiter unterbricht
- Jeder Teilnehmer, der den anderen unterbricht
- Ein Mangel an Unterbrechungen

Hier ist ein Beispiel für die verfügbaren Parameter mit [InterruptionFilter](#):

```

"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
},

```

Weitere Informationen zu diesen Parametern und den damit verbundenen gültigen Werten finden Sie unter [CreateCallAnalyticsCategory](#) und [InterruptionFilter](#).

Schlüsselwort-Übereinstimmung

Regeln mit Schlüsselwörtern ([TranscriptFilter](#)-Datentyp) sind auf Übereinstimmung ausgelegt:

- Benutzerdefinierte Wörter oder Phrasen, die vom Kundendienstmitarbeiter, dem Kunden oder beiden gesprochen werden
- Eigene Wörter oder Formulierungen, die nicht vom Kundendienstmitarbeiter, dem Kunden oder beiden gesprochen werden
- Benutzerdefinierte Wörter oder Formulierungen, die in einem bestimmten Zeitrahmen vorkommen

Hier ist ein Beispiel für die verfügbaren Parameter mit [TranscriptFilter](#):

```
"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  
}
```

Weitere Informationen zu diesen Parametern und den damit verbundenen gültigen Werten finden Sie unter [CreateCallAnalyticsCategory](#) und [TranscriptFilter](#).

Non-talk Zeit, Spiel

Regeln mit Nicht-Gesprächszeit ([NonTalkTimeFilter](#)-Datentyp) sind auf Übereinstimmung ausgelegt:

- Das Vorhandensein von Stille zu bestimmten Zeiten während des Gesprächs

- Das Vorhandensein von Sprache zu bestimmten Zeiten während des Gesprächs

Hier ist ein Beispiel für die verfügbaren Parameter mit [NonTalkTimeFilter](#):

```
"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
},
```

Weitere Informationen zu diesen Parametern und den damit verbundenen gültigen Werten finden Sie unter [CreateCallAnalyticsCategory](#) und [NonTalkTimeFilter](#).

Stimmungsübereinstimmung

Regeln mit Stimmung ([SentimentFilter](#)-Datentyp), sind auf Übereinstimmung:

- Vorhandensein oder Nichtvorhandensein einer positiven Stimmung, die vom Kunden, vom Kundendienstmitarbeiter oder von beiden an bestimmten Punkten des Anrufs ausgedrückt wird
- Vorhandensein oder Nichtvorhandensein eines negativen Gefühls, das vom Kunden, vom Kundendienstmitarbeiter oder von beiden an bestimmten Punkten des Anrufs geäußert wurde
- Vorhandensein oder Nichtvorhandensein einer neutralen Stimmung, die vom Kunden, vom Kundendienstmitarbeiter oder von beiden an bestimmten Punkten des Anrufs ausgedrückt wird
- Vorhandensein oder Nichtvorhandensein einer gemischten Stimmung, die vom Kunden, vom Kundendienstmitarbeiter oder von beiden an bestimmten Punkten des Anrufs ausgedrückt wird

Hier ist ein Beispiel für die verfügbaren Parameter mit [SentimentFilter](#):

```
"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 ]
},
```

Weitere Informationen zu diesen Parametern und den damit verbundenen gültigen Werten finden Sie unter [CreateCallAnalyticsCategory](#) und [SentimentFilter](#).

Beginn einer analytischen Nach Anrufen-Transkription

Bevor Sie mit einer Analytics-Transkription nach dem Anruf beginnen, müssen Sie alle [Kategorien](#) erstellen, denen Sie in Ihrem Audio entsprechen Amazon Transcribe möchten.

Note

Call-Analytics-Transkripte können nicht rückwirkend neuen Kategorien zugeordnet werden. Nur die Kategorien, die Sie vor Beginn einer Call-Analytics-Transkription erstellen, können auf diese Transkriptionsausgabe angewendet werden.

Wenn Sie eine oder mehrere Kategorien erstellt haben und Ihr Audiomaterial allen Regeln in mindestens einer Ihrer Kategorien entspricht, kennzeichnet Amazon Transcribe Ihre Ausgabe mit der entsprechenden Kategorie. Wenn Sie sich dafür entscheiden, keine Kategorien zu verwenden, oder wenn Ihr Audio nicht den in Ihren Kategorien angegebenen Regeln entspricht, wird Ihr Transkript nicht markiert.

Um eine analytische Transkription nach Anrufen zu starten, können Sie die SDKs AWS , AWS-Managementkonsole oder AWS CLI verwenden. Beispiele finden Sie im Folgenden:

AWS-Managementkonsole

Gehen Sie wie folgt vor, um einen Auftrag für eine Analyse nach Anrufen zu starten. Die Anrufe, die allen von einer Kategorie definierten Merkmalen entsprechen, werden mit dieser Kategorie gekennzeichnet.

1. Wählen Sie im Navigationsbereich unter Amazon Transcribe Call Analytics die Option Call Analytics-Jobs aus.

2. Wählen Sie Job erstellen aus.

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
☐	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.](#) 

Cancel

Previous

Create job

3. Geben Sie auf der Seite Auftragsdetails angeben Informationen zu Ihrem Call-Analytics-Auftrag an, einschließlich des Speicherorts Ihrer Eingabedaten.

Specify job details [Info](#)

Job settings

Name

MyCallAnalyticsJob

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.

English, US (en-US) ▼

Input data [Info](#)

Input file location on S3

Choose an input audio or video file in Amazon S3.

s3://bucket/prefix/file.mp3

[Browse S3](#)

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

Agent audio channel identification [Info](#)

Choose the channel that has the speech from the agent. The other channel is used for the customer's speech.

Channel 1 ▼

Geben Sie den gewünschten Amazon S3 Speicherort Ihrer Ausgabedaten und die zu verwendende IAM Rolle an.

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

Cancel **Next**

4. Wählen Sie Weiter aus.
5. Aktivieren Sie unter Auftrag konfigurieren alle optionalen Features, die Sie in Ihren Call-Analytics-Auftrag aufnehmen möchten. Wenn Sie zuvor Kategorien erstellt haben, werden diese im Bereich Kategorien angezeigt und automatisch auf Ihren Call-Analytics-Auftrag angewendet.

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 ▾
☐	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.](#) 

Cancel

Previous

Create job

6. Wählen Sie Job erstellen aus.

AWS CLI

In diesem Beispiel werden der Befehl [start-call-analytics-job](#) und der Parameter `channel-definitions` verwendet. Weitere Informationen erhalten Sie unter [StartCallAnalyticsJob](#) und [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
```

Hier ein weiteres Beispiel mit dem Befehl [start-call-analytics-job](#) und einem Anforderungstext, der Call Analytics für diesen Auftrag aktiviert.

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

Die Datei `my-call-analytics-job.json` enthält den folgenden Anforderungstext.

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

AWS SDK für Python (Boto3)

In diesem Beispiel wird der verwendet AWS SDK für Python (Boto3) , um einen Call Analytics-Job mit der Methode [start_call_analytics_job](#) zu starten. Weitere Informationen erhalten Sie unter [StartCallAnalyticsJob](#) und [ChannelDefinition](#).

Weitere Beispiele für die Verwendung der AWS SDKs, einschließlich funktionspezifischer, szenarienspezifischer und serviceübergreifender Beispiele, finden Sie im Kapitel. [Codebeispiele für Amazon Transcribe mit AWS SDKs](#)

```
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)
```

Post-call Ergebnisse aus Analysen

Post-call Analytics-Transkripte werden Schritt für Schritt nach Segmenten angezeigt. Dazu gehören Anrufkategorisierung, Anrufmerkmale (Lautstärkewerte, Unterbrechungen, Nicht-Gesprächszeit, Gesprächsgeschwindigkeit), Anrufzusammenfassung (Probleme, Ergebnisse und Aktionselemente), Schwärzung und Stimmung. Zusätzlich wird am Ende des Transkripts eine Zusammenfassung der Gesprächsmerkmale bereitgestellt.

Um die Genauigkeit zu erhöhen und Ihre Transkripte weiter an Ihren Anwendungsfall anzupassen, z. B. durch Einbeziehung branchenspezifischer Begriffe, fügen Sie Ihrer Call-Analytics-Anfrage [benutzerdefinierte Vokabulare](#) oder [benutzerdefinierte Sprachmodelle](#) hinzu. Um Wörter zu maskieren, zu entfernen oder zu markieren, die Sie in Ihren Transkriptionsergebnissen nicht möchten, z. B. Obszönitäten, fügen Sie [eine Wortschatzfilterung](#) hinzu. Wenn Sie sich nicht sicher sind, welcher Sprachcode an die Mediendatei übergeben werden soll, können Sie die [Batch-Sprachenidentifizierung aktivieren](#), um die Sprache in Ihrer Mediendatei automatisch zu identifizieren.

Die folgenden Abschnitte zeigen Beispiele für die JSON-Ausgabe auf Einblicksebene. Kompilierte Ausgaben finden Sie unter [Kompilierte Analyse nach Anrufen-Ausgaben](#).

Anrufkategorisierung

So sieht eine Kategorieübereinstimmung in Ihrer Transkriptionsausgabe aus. Dieses Beispiel zeigt, dass das Audio vom Zeitstempel von 40040 Millisekunden bis zum Zeitstempel von 42460 Millisekunden mit der Kategorie „positive Auflösung“ übereinstimmt. In diesem Fall erforderte die benutzerdefinierte Kategorie „positive Auflösung“ eine positive Stimmung in den letzten Sekunden der Rede.

```
"Categories": {
  "MatchedDetails": {
    "positive-resolution": {
      "PointsOfInterest": [
        {
          "BeginOffsetMillis": 40040,
          "EndOffsetMillis": 42460
        }
      ]
    }
  },
}
```



```

    "MatchedCategories": [
      " positive-resolution"
    ]
  },

```

Anrufmerkmale

So sehen die Anrufmerkmale in Ihrer Transkriptionsausgabe aus. Beachten Sie, dass für jeden Gesprächsschritt die Lautstärkewerte angegeben werden, während alle anderen Merkmale am Ende des Transkripts angegeben werden.

```

"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
}
},

```

Probleme, Maßnahmen und nächste Schritte

- Im folgenden Beispiel werden Probleme identifiziert, die bei Zeichen 7 beginnen und bei Zeichen 51 enden, was sich auf diesen Abschnitt des Textes bezieht: „Ich möchte mein Rezeptabonnement kündigen“.

```

"Content": "Well, I would like to cancel my recipe subscription.",
"IssuesDetected": [

```

```
{
  "CharacterOffsets": {
    "Begin": 7,
    "End": 51
  }
},
```

- Im folgenden Beispiel werden die Ergebnisse als beginnend bei Zeichen 12 und endend bei Zeichen 78 identifiziert, was sich auf diesen Abschnitt des Textes bezieht: „Ich habe alle Änderungen an Ihrem Konto vorgenommen und jetzt wird dieser Rabatt angewendet“.

```
"Content": "Wonderful. I made all changes to your account and now this discount is applied, please check.",
"OutcomesDetected": [
  {
    "CharacterOffsets": {
      "Begin": 12,
      "End": 78
    }
  }
],
```

- Im folgenden Beispiel werden Aktionspunkte als beginnend mit Zeichen 0 und endend mit Zeichen 103 identifiziert, was sich auf diesen Abschnitt des Textes bezieht: „Ich werde Ihnen heute eine E-Mail mit allen Details senden und Sie nächste Woche zur Nachverfolgung zurückrufen“.

```
"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
    }
  }
],
```

Generative Anrufzusammenfassung

So sieht die generative Anrufzusammenfassung in Ihrer Transkriptionsausgabe aus:

```
"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."
    }
  }
}
```

In den folgenden Fällen wird der Analyseauftrag ohne Generierung einer Zusammenfassung abgeschlossen:

- Ungenügender Gesprächsinhalt: Die Konversation muss mindestens eine Runde sowohl vom Agenten als auch vom Kunden beinhalten. Wenn der Konversationsinhalt nicht ausreicht, gibt der Service den Fehlercode `INSUFFICIENT_CONVERSATION_CONTENT` zurück.
- Sicherheitsleitplanken: Die Konversation muss den vorhandenen Sicherheitsvorkehrungen entsprechen, um sicherzustellen, dass eine angemessene Zusammenfassung generiert wird. Wenn diese Leitplanken nicht eingehalten werden, gibt der Service den Fehlercode `FAILED_SAFETY_GUIDELINES` zurück.

Der Fehlercode befindet sich im Abschnitt innerhalb der Ausgabe. Skipped AnalyticsJobDetails Möglicherweise finden Sie den Fehlergrund auch CallAnalyticsJobDetails in der [GetCallAnalyticsJob](#) API-Antwort.

Beispiel für eine Fehlerausgabe

```
{
  "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",
...
}

```

Stimmungsanalyse

So sieht die Stimmungsanalyse in Ihrer Transkriptionsausgabe aus.

- Qualitative Turn-by-Turn-Stimmungswerte:

```

"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"

```

- Quantitative Stimmungswerte für den gesamten Anruf:

```

"Sentiment": {
  "OverallSentiment": {
    "AGENT": 2.5,
    "CUSTOMER": 2.1
  }
}

```

```
},
```

- Quantitative Stimmungswerte pro Teilnehmer und pro Anrufviertel:

```
"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-Schwärzung

So sieht die PII-Schwärzung in Ihrer Transkriptionsausgabe aus.

```
"Content": "[PII], my name is [PII], how can I help?",
"Redaction": [{
  "Confidence": "0.9998",
  "Type": "NAME",
  "Category": "PII"
}]
```

Weitere Informationen finden Sie unter [Schwärzen von PII in Ihrem Batch-Job](#).

Sprachidentifizierung

So sieht die Sprachidentifizierung in Ihrer Transkriptionsausgabe aus, wenn das Feature aktiviert ist.

```
"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"
}]
```

```
}]
```

Im obigen Ausgabebeispiel füllt die Sprachidentifizierung die Sprachcodes mit Konfidenzwerten. Das Ergebnis mit dem höchsten Konfidenzwert wird als Sprachcode für die Transkription ausgewählt. Weitere Informationen finden Sie unter [Identifizierung der vorherrschenden Sprachen in Ihren Medien](#).

Kompilierte Analyse nach Anrufen-Ausgaben

Der Kürze halber werden einige Inhalte in der folgenden Transkriptionsausgabe durch Ellipsen ersetzt.

Dieses Beispiel beinhaltet die optionale Funktion Generative Anrufzusammenfassung.

```
{
  "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": []
    },
    ...
}

```

Aktivieren der generativen Anrufzusammenfassung

Note

Bereitgestellt von Amazon Bedrock: AWS implementiert [automatisierte Missbrauchserkennung](#). Da die durch generative KI gestützte Zusammenfassung nach erfolgreichem Kontakt auf Amazon Bedrock basiert, können Benutzer die in Amazon Bedrock

implementierten Kontrollen zur Durchsetzung von Sicherheit und dem verantwortungsvollen Einsatz von künstlicher Intelligenz (KI) in vollem Umfang nutzen.

Im Folgenden finden Sie Beispiele für die Verwendung der generativen Anrufzusammenfassung mit einem Auftrag zur Analyse nach dem Anruf:

AWS-Managementkonsole

Aktivieren Sie im Bereich „Zusammenfassung“ die Option „Generative Anrufzusammenfassung“, um eine Zusammenfassung in der Ausgabe zu erhalten.

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 ▾
☐	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.](#) 

[Cancel](#)[Previous](#)[Create job](#)

AWS CLI

In diesem Beispiel werden der Befehl [start-call-analytics-job](#) und der Parameter Settings mit den Subparametern Summarization verwendet. Weitere Informationen finden Sie unter [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}}'
```

Hier ein weiteres Beispiel mit dem Befehl [start-call-analytics-job](#) und einem Anforderungstext, der eine Zusammenfassung für diesen Auftrag aktiviert.

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

Die Datei my-call-analytics-job.json enthält den folgenden Anforderungstext.

```
{  
  "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
    }
}
}
}

```

AWS SDK für Python (Boto3)

[In diesem Beispiel wird mithilfe der Methode AWS SDK für Python \(Boto3\) `start\_call\_analytics\_job` ein Call Analytics mit aktivierter Zusammenfassung gestartet.](#) Weitere Informationen finden Sie unter [StartCallAnalyticsJob](#).

Weitere Beispiele für die Verwendung der AWS SDKs, einschließlich funktionspezifischer, szenarienspezifischer und serviceübergreifender Beispiele, finden Sie im Kapitel. [Codebeispiele für Amazon Transcribe mit AWS SDKs](#)

```

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)
```

Real-time Rufen Sie Analytics auf

Real-time Call Analytics bietet Einblicke in Echtzeit, die zur Behebung von Problemen und zur Abmilderung von Eskalationen verwendet werden können, sobald sie auftreten.

Die folgenden Erkenntnisse sind mit Call Analytics in Echtzeit verfügbar:

- [Kategorieereignisse](#), die mithilfe von Regeln bestimmte Schlüsselwörter und Ausdrücke kennzeichnen; Kategorieereignisse können zur Erstellung von [Echtzeitwarnungen](#) verwendet werden
- [Erkennung von Problemen](#) identifiziert die in jedem Audiosegment angesprochenen Probleme
- [Identifizierung von PII \(sensiblen Daten\)](#) in Ihrem Texttranskript
- [Schwärzung von PII \(sensible Daten\)](#) in Ihrem Texttranskript
- [Stimmungsanalyse](#) für jedes Sprachsegment
- Die [Sprachenerkennung](#) erkennt die Hauptsprache, die in jedem Audiokanal gesprochen wird

Zusätzlich zur Anrufanalyse in Echtzeit Amazon Transcribe können Sie auch Analysen [nach dem Anruf](#) für Ihren Medienstream durchführen. Mithilfe des [PostCallAnalyticsSettings](#)-Parameters können Sie Analysen nach Anrufen in Ihre Echtzeit-Call Analytics-Anfrage einbeziehen.

Real-time Einblicke

In diesem Abschnitt werden die Erkenntnisse beschrieben, die für Call-Analytics-Transkriptionen in Echtzeit verfügbar sind.

Kategorieereignisse

Mithilfe von Kategorieereignissen können Sie Ihre Transkription anhand eines genauen Schlüsselworts oder einer Formulierung abgleichen. Wenn Sie beispielsweise einen Filter für den Ausdruck „Ich möchte mit dem Manager sprechen“ festlegen, wird nach genau diesem Satz Amazon Transcribe gefiltert.

Hier ist ein [Beispiel für die Ausgabe](#).

Weitere Informationen zum Erstellen von Call-Analytics-Kategorien in Echtzeit finden Sie unter [Kategorien für Echtzeit-Transkriptionen erstellen](#).

Tip

Mit Kategorieereignissen können Sie Echtzeitwarnungen einstellen; weitere Informationen finden Sie unter [Erstellung von Echtzeitwarnungen für Kategorieübereinstimmungen](#).

Erkennung von Problemen

Die Erkennung von Problemen bietet kurze Zusammenfassungen der erkannten Probleme in jedem Audiosegment. Mit dem Feature „Erkennung von Problemen“ können Sie:

- Reduzieren Sie den Bedarf an manuellen Notizen während und nach Anrufen
- Verbesserung der Effizienz der Kundendienstmitarbeiter, damit sie schneller auf Kunden reagieren können

Note

Die Erkennung von Problemen wird mit den folgenden englischen Dialekten unterstützt: Australisch (en-AU), Britisch (en-GB) und US (en-US).

Das Feature zur Erkennung von Problemen ist in allen Branchen und Geschäftsbereichen und ist kontextabhängig. Sie ist sofort einsatzbereit und unterstützt daher keine Anpassungen wie Modelltraining oder benutzerdefinierte Kategorien.

Die Erkennung von Problemen mit Call Analytics in Echtzeit wird für jedes komplette Audiosegment durchgeführt.

Hier ist ein [Beispiel für die Ausgabe](#).

PII-Identifizierung (sensible Daten)

Die Kennzeichnung sensibler Daten kennzeichnet persönlich identifizierbare Informationen (PII) im Texttranskript. Dieser Parameter ist nützlich für den Schutz von Kundeninformationen.

Note

Real-time Die Identifizierung personenbezogener Daten wird in den folgenden englischen Dialekten unterstützt: Australisch (en-AU), Britisch (en-GB), USA (en-US) und spanischsprachiger Dialekt (es-US).

Die PII-Identifizierung mit Echtzeit-Call Analytics wird für jedes komplette Audiosegment durchgeführt.

Eine Liste der personenbezogenen Daten, die mithilfe dieser Funktion identifiziert wurden, oder weitere Informationen zur PII-Identifizierung mit finden Sie unter [Amazon Transcribe Schwärzung oder Identifizierung persönlich identifizierbarer Informationen](#)

Hier ist ein [Beispiel für eine Ausgabe](#).

Schwärzung von PII (sensible Daten)

Bei der Schwärzung sensibler Daten werden persönlich identifizierbare Informationen (PII) in Ihrem Texttranskript durch die Art der PII ersetzt (z. B. [NAME]). Dieser Parameter ist nützlich für den Schutz von Kundeninformationen.

 Note

Real-time Die Bearbeitung personenbezogener Daten wird in den folgenden englischen Dialekten unterstützt: Australisch (en-AU), Britisch (), USA (en-GB) und spanischsprachiger Dialekt (en-US). es-US

Die PII-Schwärzung mit Echtzeit-Call Analyticsn wird für jedes komplette Audiosegment durchgeführt.

Um die Liste der mit diesem Feature geschwärzten personenbezogenen Daten anzuzeigen oder mehr über die Schwärzung mit Amazon Transcribe zu erfahren, besuchen Sie [Schwärzung oder Identifizierung persönlich identifizierbarer Informationen](#).

Hier ist ein [Beispiel für eine Ausgabe](#).

Stimmungsanalyse

Die Stimmungsanalyse schätzt ein, wie sich der Kunde und der Kundendienstmitarbeiter während des Gesprächs fühlen. Diese Metrik wird für jedes Sprachsegment bereitgestellt und als qualitativer Wert (positive, neutral mixed oder negative) dargestellt.

Mit diesem Parameter können Sie die Gesamtstimmung für jeden Gesprächsteilnehmer und die Stimmung für jeden Teilnehmer während jedes Sprachsegments qualitativ auswerten. Anhand dieser Kennzahl lässt sich feststellen, ob Ihr Kundendienstmitarbeiter in der Lage ist, einen verärgerten Kunden bis zum Ende des Anrufs zufrieden zu stellen.

Die Stimmungsanalyse mit Echtzeit-Call Analyticsn wird für jedes komplette Audiosegment durchgeführt.

Die Stimmungsanalyse ist sofort verfügbar und unterstützt daher keine Anpassung, z. B. Modelltraining oder benutzerdefinierte Kategorien.

Hier ist ein [Beispiel für die Ausgabe](#).

Sprachidentifizierung

Die Sprachenerkennung erkennt und bestimmt bei der Anrufanalyse in Echtzeit automatisch die Hauptsprache, die in jedem Kanal Ihrer Audiostreams gesprochen wird. Nach der Identifizierung verarbeitet Call Analytics die am besten geeignete Transkription auf der Grundlage der erkannten

Sprache und sendet sie zurück. Diese Informationen werden dann in Echtzeit über den Stream zurückgesendet.

Mit dieser Funktion können Sie automatisch die dominante Sprache erkennen und identifizieren, die in jedem Kanal Ihres Audiostreams gesprochen wird. Sobald die Sprache erkannt wurde, verarbeitet Call Analytics die entsprechende Transkription für die identifizierte Sprache und liefert sie in Echtzeit.

Die automatische Sprachenerkennung wird für alle von [Call Analytics Streaming unterstützten Sprachen unterstützt](#), die derzeit ohne zusätzliche Kosten für Streaming-Transkriptionen unterstützt werden. Sie ist in den von Call Analytics Streaming unterstützten [AWS-Regionen](#) verfügbar.

Important

Call Analytics unterstützt nur die einsprachige Identifizierung, wodurch die dominante Sprache identifiziert wird, die in Ihrem Audiokanal gesprochen wird. Multi-language Die Identifizierung wird nicht unterstützt, was bedeutet, dass jeder Kanal nur in einer Sprache transkribiert werden kann.

Um die Sprachenidentifikation verwenden zu können, müssen Sie mindestens zwei Sprachcodes und höchstens fünf Sprachcodes angeben. Außerdem können Sie pro Sprache und Stream nur einen Sprachdialekt aus den [unterstützten Call Analytics-Streaming-Sprachen](#) auswählen. Das bedeutet, dass Sie en-US und en-AU nicht als Sprachoptionen für dieselbe Transkription auswählen können. Wenn Sie diese Funktion verwenden, muss der LanguageCode Parameter in der Anfrage leer bleiben, da IdentifyLanguage sich beide LanguageCode Optionen gegenseitig ausschließen.

Warning

Wenn Ihre angegebenen Sprachcodes nicht mit der tatsächlich gesprochenen Sprache übereinstimmen, wählt das System aus Ihren Optionen die ähnlichste Sprache aus, was zu ungenauen Transkriptionen führen kann.

Mithilfe der Funktion zur Sprachenerkennung können Sie:

- Automatische Erkennung der dominierenden Sprache in Echtzeit
- Verarbeiten Sie verschiedene Sprachen über separate Kanäle
- Erhalten Sie Konfidenzwerte für die Spracherkennung

- Wenden Sie sprachspezifische benutzerdefinierte Vokabeln an

Um die Sprachenidentifikation zu verwenden, müssen Sie die folgenden Parameter konfigurieren:

Erforderliche Parameter:

- `identifyLanguage`- Auf `true` setzen, um die Sprachenidentifikation zu aktivieren.
- `languageOptions`- Eine Liste möglicher Sprachcodes, die verwendet werden können, wenn `IdentifyLanguage` auf `true` gesetzt ist. Sie müssen mindestens zwei Sprachen auswählen, da die Auswahl einer Sprache nicht unterstützt wird.

Optionale Parameter:

- `preferredLanguage`- Ihre erwartete Hauptsprache aus den bereitgestellten `LanguageOptions`. Durch das Hinzufügen einer bevorzugten Sprache kann Call Analytics die Sprache schneller identifizieren.
- `vocabularyNames`- Benutzerdefinierte Vokabelnamen für eine verbesserte Genauigkeit. Beachten Sie, dass bei Vokabelnamen Groß- und Kleinschreibung beachtet wird. Wenn die Sprache des benutzerdefinierten Vokabulars nicht mit der identifizierten Mediensprache übereinstimmt, wird sie nicht auf die Transkription angewendet.
- `vocabularyFilterNames`- Namen nach Vokabeln filtern, um die Transkriptaussage individuell anzupassen.

Hier ist ein [Beispiel für die Ausgabe](#).

Kategorien für Echtzeit-Transkriptionen erstellen

Real-time Call Analytics unterstützt die Erstellung benutzerdefinierter Kategorien, mit denen Sie Ihre Transkriptanalysen optimal an Ihre spezifischen Geschäftsanforderungen anpassen können.

Sie können so viele Kategorien erstellen, wie Sie möchten, um eine Reihe verschiedener Szenarien abzudecken. Für jede Kategorie, die Sie erstellen, müssen Sie zwischen 1 und 20 Regeln erstellen. Real-time Call Analytics-Transkriptionen unterstützen nur Regeln, die [TranscriptFilter](#) (Stichwortübereinstimmungen) verwenden. Weitere Informationen zur Verwendung von Regeln mit dem [CreateCallAnalyticsCategory](#)-Vorgang finden Sie im [Regelkriterien für Echtzeit-Call-Analytics-Kategorien](#)-Abschnitt.

Wenn der Inhalt Ihrer Medien allen Regeln entspricht, die Sie in einer bestimmten Kategorie angegeben haben, kennzeichnet Amazon Transcribe Ihre Ausgabe mit dieser Kategorie. Ein Beispiel für eine Kategorieübereinstimmung im JSON-Ausgabeformat finden Sie unter [Ausgabe von Kategorieereignissen](#).

Hier sind einige Beispiele dafür, was Sie mit benutzerdefinierten Kategorien machen können:

- Identifizieren von Problemen, die sofortige Aufmerksamkeit erfordern, indem bestimmte Gruppen von Schlüsselwörtern markiert und verfolgt werden
- Überwachung der Einhaltung von Vorschriften, z. B. wenn ein Kundendienstmitarbeiter einen bestimmten Satz spricht (oder auslöst)
- Kennzeichnen Sie bestimmte Wörter und Formulierungen in Echtzeit. Sie können dann Ihre Kategorieübereinstimmung festlegen, um eine sofortige Warnung auszulösen. Wenn Sie beispielsweise eine Echtzeit-Call-Analytics-Kategorie für einen Kunden mit der Aufforderung „Mit einem Vorgesetzten sprechen“ erstellen, können Sie eine [Ereigniswarnung](#) für diese Echtzeit-Kategorieübereinstimmung festlegen, die den diensthabenden Vorgesetzten benachrichtigt.

Post-call im Vergleich zu Echtzeit-Kategorien

Wenn Sie eine neue Kategorie erstellen, können Sie angeben, ob sie als Nach Anrufen-Kategorie (POST_CALL) oder als Echtzeitkategorie (REAL_TIME) erstellt werden soll. Wenn Sie keine Option angeben, wird Ihre Kategorie standardmäßig als Kategorie nach dem Anruf erstellt. Real-time Übereinstimmungen mit Kategorien können verwendet werden, um Benachrichtigungen in Echtzeit zu erstellen. Weitere Informationen finden Sie unter [Erstellung von Echtzeitwarnungen für Kategorieübereinstimmungen](#).

Um eine neue Kategorie für die Echtzeit-Call-Analytics zu erstellen, können Sie die SDKs AWS-Managementkonsole, AWS CLI oder AWS verwenden; Beispiele finden Sie im Folgenden:

AWS-Managementkonsole

1. Wählen Sie im Navigationsbereich Amazon Transcribe unter Amazon Transcribe Call Analytics aus.
2. Wählen Sie Anruf-Analytics-Kategorien, um zur Seite Anruf-Analytics-Kategorien zu gelangen. Wählen Sie die Schaltfläche Kategorie erstellen.

Call analytics categories (0) [Info](#)

[Edit](#) [Delete](#) [Create category](#)

< 1 > [ⓘ](#)

Name	Created	Modified
No categories have been created.		

[Create category](#)

- Sie befinden sich jetzt auf der Seite „Kategorie erstellen“. Geben Sie einen Namen für Ihre Kategorie ein und wählen Sie dann „Anruf-Analytics in Echtzeit“ im Dropdown-Menü Kategorietyp.

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.

- Sie können eine Vorlage wählen, um Ihre Kategorie zu erstellen, oder Sie können eine neue erstellen.

Wenn Sie eine Vorlage verwenden: Wählen Sie Vorlage verwenden (empfohlen), wählen Sie die gewünschte Vorlage und wählen Sie dann Kategorie erstellen.

Category settings

Category name

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

Category type [Info](#)

Category creation method [Info](#)

☒ Use a template (recommended)
Use a template to edit predefined rules.

☐ Create from scratch
If you know the rules that you want to define, choose this option.

Template type [Info](#)

Choose the template for the category that most closely matches the one you want to create.

Customer content is negative and mentioned manager

5. Wenn Sie eine benutzerdefinierte Kategorie erstellen: Wählen Sie Von Grund auf neu erstellen.

6. Fügen Sie über das Dropdown-Menü Regeln zu Ihrer Kategorie hinzu. Sie können bis zu 20 Regeln pro Kategorie hinzufügen. Bei Echtzeit-Transkriptionen von Call Analytics können Sie nur Regeln einbeziehen, die Übereinstimmungen mit dem Inhalt der Transkription beinhalten. Alle Übereinstimmungen werden in Echtzeit angezeigt.

7. Hier ein Beispiel für eine Kategorie mit einer Regel: ein Kunde, der zu irgendeinem Zeitpunkt des Anrufs sagt: „Mit einem Vorgesetzten sprechen“.

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.

speak to a manager Add a new word or phrase

The word or phrase can be up to 2,000 characters.

Add rule

You can add up to 19 more rules.

8. Wenn Sie alle Regeln zu Ihrer Kategorie hinzugefügt haben, wählen Sie Kategorie erstellen.

AWS CLI

In diesem Beispiel wird der Befehl [create-call-analytics-category](#) verwendet. Weitere Informationen finden Sie unter [CreateCallAnalyticsCategory](#), [CategoryProperties](#) und [Rule](#).

Im folgenden Beispiel wird eine Kategorie mit der Regel erstellt:

- Der Kunde hat zu einem beliebigen Zeitpunkt des Gesprächs den Satz „Mit einem Vorgesetzten Sprechen“ gesprochen.

In diesem Beispiel wird der Befehl [create-call-analytics-category](#) verwendet und ein Anforderungskörper, der eine Regel zu Ihrer Kategorie hinzufügt.

```
aws transcribe create-call-analytics-category \
--cli-input-json file:///filepath/my-first-analytics-category.json
```

Die Datei my-first-analytics-category.json enthält den folgenden Anforderungstext.

```
{
```

```

"CategoryName": "my-new-real-time-category",
"InputType": "REAL_TIME",
"Rules": [
    {
        "TranscriptFilter": {
            "Negate": false,
            "Targets": [
                "speak to the manager"
            ],
            "TranscriptFilterType": "EXACT"
        }
    }
]
}

```

AWS SDK für Python (Boto3)

In diesem Beispiel wird mithilfe der AWS SDK für Python (Boto3) Rules Argumente CategoryName und für die Methode [create_call_analytics_category](#) eine Kategorie erstellt. Weitere Informationen finden Sie unter [CreateCallAnalyticsCategory](#), [CategoryProperties](#) und [Rule](#).

Weitere Beispiele für die Verwendung der AWS SDKs, einschließlich funktionspezifischer, szenarienspezifischer und serviceübergreifender Beispiele, finden Sie im Kapitel. [Codebeispiele für Amazon Transcribe mit AWS SDKs](#)

Im folgenden Beispiel wird eine Kategorie mit der Regel erstellt:

- Der Kunde hat zu einem beliebigen Zeitpunkt des Gesprächs den Satz „Mit einem Vorgesetzten Sprechen“ gesprochen.

```

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)

```

Regelkriterien für Echtzeit-Call-Analytics-Kategorien

In diesem Abschnitt werden die Arten von benutzerdefinierten REAL_TIME Regeln beschrieben, die Sie mit der [CreateCallAnalyticsCategory](#)-API-Operation erstellen können.

Die Erkennung von Problemen erfolgt automatisch, sodass Sie keine Regeln oder Kategorien zur Kennzeichnung von Problemen erstellen müssen.

Beachten Sie, dass für Echtzeit-Transkriptionen von Call Analytics nur Schlüsselwortübereinstimmungen unterstützt werden. Wenn Sie Kategorien erstellen möchten, die Unterbrechungen, Stille oder Stimmungen beinhalten, lesen Sie bitte [Regelkriterien für Analysekatgorien nach Anrufen](#).

Schlüsselwort-Übereinstimmung

Regeln mit Schlüsselwörtern ([TranscriptFilter](#)-Datentyp) sind auf Übereinstimmung ausgelegt:

- Benutzerdefinierte Wörter oder Phrasen, die vom Kundendienstmitarbeiter, dem Kunden oder beiden gesprochen werden
- Eigene Wörter oder Formulierungen , die nicht vom Kundendienstmitarbeiter, dem Kunden oder beiden gesprochen werden
- Benutzerdefinierte Wörter oder Formulierungen, die in einem bestimmten Zeitrahmen vorkommen

Hier ist ein Beispiel für die verfügbaren Parameter mit [TranscriptFilter](#):

```

"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
}
```

Weitere Informationen zu diesen Parametern und den damit verbundenen gültigen Werten finden Sie unter [CreateCallAnalyticsCategory](#) und [TranscriptFilter](#).

Post-call Analytik mit Transkriptionen in Echtzeit

Post-call Analytics ist eine optionale Funktion, die für Call Analytics-Transkriptionen in Echtzeit verfügbar ist. Zusätzlich zu den standardmäßigen [Echtzeit-Analysen](#) bietet Ihnen die Analyse nach Anrufen folgende Informationen:

- **Aktionspunkte:** Auflistung aller in dem Anruf genannten Maßnahmen
- **Unterbrechungen:** Misst, ob und wann ein Teilnehmer den anderen Teilnehmer mitten im Satz unterbricht
- **Probleme:** Enthält die in dem Anruf genannten Probleme
- **Lautstärke:** Misst die Lautstärke, mit der jeder Teilnehmer spricht
- **Non-talk Zeit:** Misst Zeiträume, in denen keine Sprache vorkommt
- **Ergebnisse:** Liefert das Ergebnis oder die Lösung, die in dem Anruf genannt wurde
- **Sprechgeschwindigkeit:** Misst die Geschwindigkeit, mit der beide Teilnehmer sprechen
- **Gesprächszeit:** Misst die Zeit (in Millisekunden), die jeder Teilnehmer während des Gesprächs gesprochen hat

Wenn diese Option aktiviert ist, erstellt die Analyse nach dem Anruf anhand eines Audiostreams ein Transkript, das einer [Analyse nach einem Anruf aus einer Audiodatei](#) ähnelt, und speichert es in dem unter angegebenen Amazon S3 Bucket. OutputLocation Darüber hinaus zeichnet die Analyse

nach dem Anruf Ihren Audiostream auf und speichert ihn als Audiodatei (WAVFormat) im selben Bucket. Amazon S3 Wenn Sie die Schwärzung aktivieren, werden ein geschwärztes Transkript und eine geschwärzte Audiodatei ebenfalls im angegebenen Bucket gespeichert. Amazon S3 Wenn Sie die Analyse nach Anrufen für Ihren Audiostream aktivieren, werden zwischen zwei und vier Dateien erzeugt, wie hier beschrieben:

- Wenn die Schwärzung nicht aktiviert ist, werden Ihre Ausgabedateien geschwärzt:
 1. Ein ungeschwärztes Transkript
 2. Eine ungeschwärzte Audiodatei
- Wenn die Schwärzung ohne die Option ungeschwärzt (redacted) aktiviert ist, sind Ihre Ausgabedateien:
 1. Ein geschwärztes Transkript
 2. Eine geschwärzte Audiodatei
- Wenn die Schwärzung mit der Option ungeschwärzt (redacted_and_unredacted) aktiviert ist, sind Ihre Ausgabedateien:
 1. Ein geschwärztes Transkript
 2. Eine geschwärzte Audiodatei
 3. Ein ungeschwärztes Transkript
 4. Eine ungeschwärzte Audiodatei

Beachten Sie, dass Sie, wenn Sie die Analyse nach Anrufen ([PostCallAnalyticsSettings](#)) mit Ihrer Anfrage aktivieren und Sie FLAC oder OPUS-OGG verwenden, erhalten Sie keine `loudnessScore` in Ihrem Transkript und es werden keine Audioaufnahmen Ihres Streams erstellt. Transcribe ist möglicherweise auch nicht in der Lage, Analysen nach dem Anruf für lang andauernde Audiostreams, die länger als 90 Minuten dauern, bereitzustellen.

Weitere Informationen zu den Erkenntnissen, die mit der Analyse nach Anrufen für Audiostreams verfügbar sind, finden Sie im Abschnitt [Erkenntnisse der Analyse nach Anrufen](#).

 Tip

Wenn Sie die Analyse nach Anrufen mit Ihrer Echtzeit-Call-Analytics-Anfrage aktivieren, werden alle Ihre POST_CALL- und REAL-TIME- Kategorien auf Ihre Analyse nach Anrufen-Transkription angewendet.

Ermöglichen von Analyse nach Anrufen

Um Analyse nach Anrufen zu aktivieren, müssen Sie den Parameter [PostCallAnalyticsSettings](#) in Ihre Echtzeit-Call-Analytics-Anfrage aufnehmen. Die folgenden Parameter müssen angegeben werden, wenn PostCallAnalyticsSettings aktiviert ist:

- **OutputLocation:** Der Amazon S3 Bucket, in dem Ihr Protokoll nach dem Anruf gespeichert werden soll.
- **DataAccessRoleArn:** Der Amazon-Ressourcenname (ARN) der Amazon S3 -Rolle, die die Berechtigung hat, auf den angegebenen Amazon S3 -Bucket zuzugreifen. Beachten Sie, dass Sie die [Vertrauensrichtlinie](#) auch für Echtzeit-Analysen verwenden müssen.

Wenn Sie eine geschwärzte Version Ihres Transkripts wünschen, können Sie ContentRedactionOutput oder ContentRedactionType in Ihre Anfrage aufnehmen. Weitere Informationen zu diesen Parametern finden Sie unter [StartCallAnalyticsStreamTranscription](#) in der API-Referenz.

Um eine Call Analytics-Transkription in Echtzeit mit aktivierter Analyse nach dem Anruf zu starten, können Sie AWS-Managementkonsole(nur Demo), oder verwenden. HTTP/2WebSockets Beispiele finden Sie unter [Start einer Call Analytics in Echtzeit](#).

Important

Derzeit bietet The AWS-Managementkonsole Only eine Demo für Anrufanalysen in Echtzeit mit vorinstallierten Audiobeispielen. Wenn Sie Ihr eigenes Audio verwenden möchten, müssen Sie die API (HTTP/2, WebSockets, oder ein SDK) verwenden.

Beispiel für die Analyse nach Anrufen

Post-call Transkripte werden in einem Turn-by-Turn-Format segmentweise angezeigt. Dazu gehören Anrufmerkmale, Stimmung, Anrufzusammenfassung, Erkennung von Problemen und (optional) PII-Schwärzung. Wenn eine Ihrer Analyse nach Anrufen-Kategorien mit dem Audioinhalt übereinstimmt, sind diese auch in Ihrer Ausgabe enthalten.

Um die Genauigkeit zu erhöhen und Ihre Transkripte weiter an Ihren Anwendungsfall anzupassen, z. B. durch Einbeziehung branchenspezifischer Begriffe, fügen Sie Ihrer Call-Analytics-Anfrage [benutzerdefinierte Vokabulare](#) oder [benutzerdefinierte Sprachmodelle](#) hinzu. Um Wörter zu

maskieren, zu entfernen oder zu markieren, die Sie in Ihren Transkriptionsergebnissen nicht möchten, z. B. Obszönitäten, fügen Sie [Wortschatzfilterung](#) hinzu.

Hier ist ein Beispiel für eine kompilierte Analyse nach Anrufen:

```
{
  "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
}
},
...
}

```

Start einer Call Analytics in Echtzeit

Bevor Sie eine Call Analytics-Transkription in Echtzeit starten, müssen Sie alle [Kategorien erstellen, denen](#) Ihr Anruf entsprechen Amazon Transcribe soll.

 Note

Call-Analytics-Transkripte können nicht rückwirkend neuen Kategorien zugeordnet werden. Nur die Kategorien, die Sie vor Beginn einer Call-Analytics-Transkription erstellen, können auf diese Transkriptionsausgabe angewendet werden.

Wenn Sie eine oder mehrere Kategorien erstellt haben und Ihr Audiomaterial allen Regeln in mindestens einer Ihrer Kategorien entspricht, kennzeichnet Amazon Transcribe Ihre Ausgabe mit den entsprechenden Kategorien. Wenn Sie sich dafür entscheiden, keine Kategorien zu verwenden, oder wenn Ihr Audio nicht den in Ihren Kategorien angegebenen Regeln entspricht, wird Ihr Transkript nicht markiert.

Um Analyse nach Anrufen in Ihre Echtzeit-Anrufanalysen-Transkription aufzunehmen, müssen Sie in Ihrer Anfrage mit dem Parameter `OutputLocation` einen Amazon S3 -Bucket angeben. Sie müssen auch eine `DataAccessRoleArn` angeben, die über Schreibrechte für den angegebenen Bucket verfügt. Nach Abschluss Ihrer Echtzeit-Call-Analytics-Streaming-Sitzung wird ein separates Transkript erstellt und im angegebenen Bucket gespeichert.

Mit der Echtzeit-Call-Analytics haben Sie auch die Möglichkeit, Echtzeit-Kategoriewarnungen zu erstellen; Anweisungen dazu finden Sie unter [Erstellung von Echtzeitwarnungen für Kategorieübereinstimmungen](#) .

Um eine Call Analytics-Transkription in Echtzeit zu starten, können Sie das AWS-ManagementkonsoleHTTP/2WebSockets, oder verwenden. Im Folgenden finden Sie Beispiele:

 Important

Derzeit bietet The AWS-Managementkonsole Only eine Demo für Call Analytics in Echtzeit mit vorinstallierten Audiobeispielen. Wenn Sie Ihr eigenes Audio verwenden möchten, müssen Sie die API (HTTP/2, WebSockets, oder ein SDK) verwenden.

AWS-Managementkonsole

Gehen Sie wie folgt vor, um eine Call-Analytics-Anfrage zu starten. Die Anrufe, die allen von einer Kategorie definierten Merkmalen entsprechen, werden mit dieser Kategorie gekennzeichnet.

Note

Für AWS-Managementkonsole ist nur eine Demo verfügbar. Um eine benutzerdefinierte Echtzeit-Analyse-Transkription zu starten, müssen Sie die [API](#) verwenden.

1. Wählen Sie im Navigationsbereich unter Amazon Transcribe Call Analytics die Option Echtzeitanruf analysieren aus.

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. Wählen Sie für Schritt 1: Eingabe von Audiodaten eine Demo-Testdatei aus dem Dropdown-Menü.



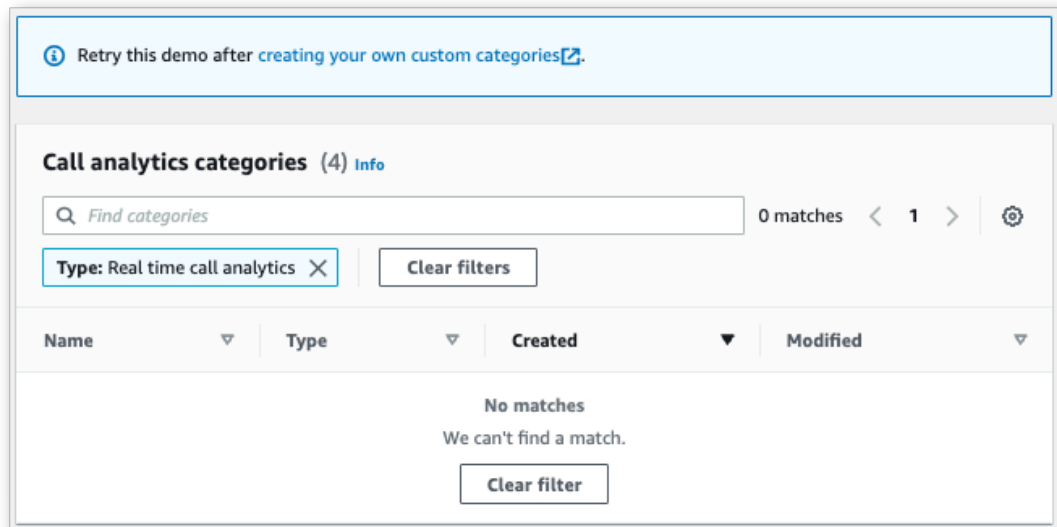
Step 1: Specify input audio

Input audio file

Insurance complaints (en-US)	▲
Insurance complaints (en-US)	✓
Hospitality complaints (en-US)	

3. Bei Schritt 2: Überprüfung der Anruferkategorien haben Sie die Möglichkeit, die zuvor erstellten Echtzeit-Call-Analytics-Kategorien zu überprüfen. Alle Echtzeit-Kategorien von Call Analytics werden auf Ihre Transkription angewendet.

Wenn Sie Kategorien anzeigen wählen, wird ein neuer Bereich geöffnet, der Ihre vorhandenen Echtzeit-Call-Analytics-Kategorien anzeigt und einen Link zum Erstellen neuer Kategorien enthält.



4. Bei Schritt 3: Konfigurieren der Ein- und Ausgabe haben Sie die Möglichkeit, zusätzliche Einstellungen vorzunehmen.

Wenn Sie die Option Erweiterte Einstellungen konfigurieren wählen, wird ein neues Fenster geöffnet, in dem Sie die Einstellungen für die Inhaltsschwärzung festlegen können.

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

Info The updates that you make here will only be applied when you start stream again.

Cancel Save

Wenn Sie alle Ihre Auswahl getroffen haben, wählen Sie Speichern, um zur Hauptseite zurückzukehren.

- Um zusätzliche Analysen anzuwenden, können Sie Post-call Analytics aktivieren. Auf diese Weise erhalten Sie dieselben Analysen wie bei einer Analysetranskription nach einem Anruf, einschließlich Unterbrechungen, Lautstärke, Nichtgesprächszeit, Gesprächsgeschwindigkeit, Gesprächszeit, Problemen, Aktionselementen und Ergebnissen. Post-call Die Analyseergebnisse werden in einer separaten Datei von Ihrem Call Analytics-Transkript in Echtzeit gespeichert.

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

Wenn Sie Analysen nach dem Anruf anwenden, müssen Sie ein Ziel für die Amazon S3 Ausgabedatei und eine IAM Rolle angeben. Optional können Sie Ihre Ausgabe auch verschlüsseln.

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
 [View](#) [Browse S3](#)

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

Choose a role ▼

[Start streaming](#)

6. Wählen Sie dann Start Streaming (Streamen starten).

HTTP/2 streamen

In diesem Beispiel wird eine HTTP/2 Anfrage erstellt, bei der Call Analytics aktiviert ist. Weitere Informationen zur Verwendung von HTTP/2 Streaming mit Amazon Transcribe finden Sie unter [Einen HTTP/2 Stream einrichten](#). Weitere Informationen zu spezifischen Parametern und Headern finden Sie Amazon Transcribe unter [StartCallAnalyticsStreamTranscription](#).

Dieses Beispiel enthält [Analyse nach Anrufen](#). Wenn Sie keine Analyse nach Anrufen wünschen, entfernen Sie den Abschnitt `PostCallAnalyticsSettings` aus der Anfrage.

Beachten Sie, dass das im folgenden Beispiel gezeigte Konfigurationsereignis als erstes Ereignis im Stream übergeben werden muss.

```
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"
    }
  }
}

```

Parameterdefinitionen finden Sie in der [API-Referenz](#). Parameter, die allen AWS API-Vorgängen gemeinsam sind, sind im Abschnitt [Allgemeine Parameter](#) aufgeführt.

WebSocket streamen

In diesem Beispiel wird eine vorsignierte URL erstellt, die Call Analytics in einem WebSocket Stream verwendet. Für eine bessere Lesbarkeit werden Zeilenumbrüche hinzugefügt. Weitere Informationen zur Verwendung von WebSocket Streams mit finden Sie Amazon Transcribe unter [Einen WebSocket Stream einrichten](#). Weitere Einzelheiten zu den Parametern finden Sie unter [StartCallAnalyticsStreamTranscription](#).

Dieses Beispiel enthält [Analyse nach Anrufen](#). Wenn Sie keine Analyse nach Anrufen wünschen, entfernen Sie den Abschnitt `PostCallAnalyticsSettings` aus der Anfrage.

Beachten Sie, dass das im folgenden Beispiel gezeigte Konfigurationsereignis als erstes Ereignis im Stream übergeben werden muss.

```

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"
      }
    }
  }
}

```

Parameterdefinitionen finden Sie in der [API-Referenz](#). Parameter, die allen AWS API-Vorgängen gemeinsam sind, sind im Abschnitt [Allgemeine Parameter](#) aufgeführt.

Tip

Zu den oben genannten HTTP/2 WebSocket Beispielen gehören Analysen nach dem Anruf. Wenn Sie keine Analyse nach Anrufen wünschen, entfernen Sie den Abschnitt `PostCallAnalyticsSettings` aus der Anfrage.

Wenn Sie `PostCallAnalyticsSettings` aktivieren, müssen Sie als erstes Ereignis ein Konfigurationsereignis senden. Ihr Konfigurationsereignis umfasst Einstellungen für `ChannelDefinitions` und `PostStreamAnalyticsSettings`, wie in den vorangegangenen Beispielen gezeigt.

Binäre Daten werden als binäre Nachricht mit `content-type application/octet-stream` und das Konfigurationsereignis als Textnachricht mit `content-type application/json` übergeben.

Weitere Informationen finden Sie unter [Einrichten einer Streaming-Transkription](#).

Erstellung von Echtzeitwarnungen für Kategorieübereinstimmungen

Um Echtzeitwarnungen einzurichten, müssen Sie zunächst eine [TranscriptFilterType](#)-Kategorie mit dem Kennzeichen `REAL_TIME` erstellen. Mit diesem Kennzeichen kann Ihre Kategorie auf Echtzeit-Transkriptionen von Call Analytics angewendet werden.

Eine Anleitung zum Erstellen einer neuen Kategorie finden Sie unter [Kategorien für Echtzeit-Transkriptionen erstellen](#).

Wenn Sie mit der Echtzeit-Transkription von Call Analytics beginnen, werden alle Kategorien, die das Kennzeichen `REAL_TIME` tragen, automatisch auf Ihre Transkriptionsausgabe auf Segmentebene angewendet. Wenn eine Übereinstimmung mit `TranscriptFilterType` vorliegt, erscheint sie unter `CategoryEvent` in Ihrem Transkript. Sie können dann diesen Parameter und seine Unterparameter `MatchedCategories` und `MatchedDetails` verwenden, um benutzerdefinierte Echtzeitwarnungen einzurichten.

Hier ist ein Beispiel für die Echtzeit-Transkriptionsausgabe von Call Analytics für eine `CategoryEvent`-Übereinstimmung:

```
"CategoryEvent": {
  "MatchedCategories": [ "shipping-complaint" ],
  "MatchedDetails": {
    "my package never arrived" : {
      "TimestampRanges": [
        {
          "BeginOffsetMillis": 19010,
          "EndOffsetMillis": 22690
        }
      ]
    }
  }
}
```

```
    ]  
  }  
}  
,
```

Das vorherige Beispiel stellt eine exakte Textübereinstimmung mit der Aussage „Mein Paket ist nie angekommen“ dar, die eine Regel innerhalb der Kategorie „Versandbeschwerde“ darstellt.

Sie können Ihre Echtzeitwarnung so einrichten, dass sie eine beliebige Kombination der aufgeführten Parameter enthält. Sie können zum Beispiel festlegen, dass Ihre Warnmeldung nur die übereinstimmende Formulierung (`MatchedDetails`) oder nur den Kategorienamen (`MatchedCategories`) enthält. Sie können auch alle Parameter in die Warnmeldung einbeziehen.

Wie Sie Ihre Echtzeitwarnungen einrichten, hängt von den Schnittstellen Ihres Unternehmens und der gewünschten Warnart ab. Sie können zum Beispiel eine `CategoryEvent`-Übereinstimmung so einstellen, dass eine Pop-up-Benachrichtigung, eine E-Mail, eine SMS oder eine andere von Ihrem System akzeptierte Warnmeldung gesendet wird.

Real-time Rufen Sie die Analytics-Ausgabe auf

Real-time Call Analytics-Transkripte werden in einem detaillierten Format nach Segmenten angezeigt. Dazu gehören Kategorie-Ereignisse, Erkennung von Problemen, Stimmungslage sowie die Identifizierung und PII-Schwärzung. Mit Kategorieereignissen können Sie Echtzeitwarnungen einstellen; weitere Informationen finden Sie unter [Erstellung von Echtzeitwarnungen für Kategorieübereinstimmungen](#).

Um die Genauigkeit zu erhöhen und Ihre Transkripte weiter an Ihren Anwendungsfall anzupassen, z. B. durch Einbeziehung branchenspezifischer Begriffe, fügen Sie Ihrer Call-Analytics-Anfrage [benutzerdefinierte Vokabulare](#) oder [benutzerdefinierte Sprachmodelle](#) hinzu. Um Wörter zu maskieren, zu entfernen oder zu markieren, die Sie in Ihren Transkriptionsergebnissen nicht möchten, z. B. Obszönitäten, fügen Sie [Wortschatzfilterung](#) hinzu.

In den folgenden Abschnitten finden Sie Beispiele für die JSON-Ausgabe für Echtzeit-Transkriptionen von Call Analytics.

Kategorieereignisse

So sieht eine Kategorieübereinstimmung in Ihrer Transkriptionsausgabe aus. Dieses Beispiel zeigt, dass die Audiodaten zwischen dem Zeitstempel 19010 Millisekunden und dem Zeitstempel 22690

Millisekunden der Kategorie „Netzwerkbeschwerde“ entsprechen. In diesem Fall erforderte die benutzerdefinierte Kategorie „Netzbeschwerde“, dass der Kunde „Netzprobleme“ sagte (genaue Wortübereinstimmung).

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

Erkennung von Problemen

So sieht eine Übereinstimmung bei der Fehlererkennung in Ihrer Transkriptionsausgabe aus. Dieses Beispiel zeigt, dass der Text von Zeichen 26 bis Zeichen 62 ein Problem beschreibt.

```
"UtteranceEvent": {
  ...
  "Transcript": "Wang Xiulan I'm tired of the network issues my phone is having.",
  ...
  "IssuesDetected": [
    {
      "CharacterOffsets": {
        "BeginOffsetChar": 26,
        "EndOffsetChar": 62
      }
    }
  ]
},
```

Stimmung

So sieht die Stimmungsanalyse in Ihrer Transkriptionsausgabe aus.

```
"UtteranceEvent": {
  ...
  "Sentiment": "NEGATIVE",
  "Items": [{
    ...
```

PII-Identifikation

So sieht die PII-Identifikation in Ihrer Transkriptionsausgabe aus.

```
"Entities": [
  {
    "Content": "Wang Xiulan",
    "Category": "PII",
    "Type": "NAME",
    "BeginOffsetMillis": 7999375,
    "EndOffsetMillis": 199375,
    "Confidence": 0.9989
  }
],
```

PII-Schwärzung

So sieht die PII-Schwärzung in Ihrer Transkriptionsausgabe aus.

```
"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
    }
  ]
},
```


Sprachidentifizierung

So sieht die Sprachenidentifikation in Ihrer Transkriptionsausgabe aus.

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

Kompilierte Echtzeit-Call-Analytics-Ausgabe

Der Kürze halber werden einige Inhalte in der folgenden Transkriptionsausgabe durch Ellipsen ersetzt.

```
{
  "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
                }
            ]
        }
    }
}
}

```

```
}  
}
```

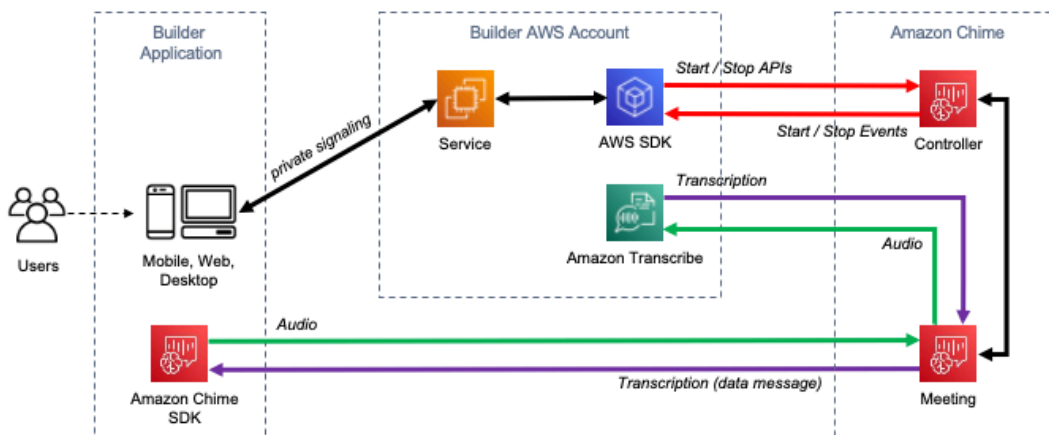
Transkribieren Sie Ihre Amazon Chime ruft in Echtzeit an

Amazon Transcribe ist in das Amazon Chime SDK integriert und ermöglicht die Transkription Ihrer Amazon Chime Anrufe in Echtzeit.

Wenn Sie über die Amazon Chime SDK-API eine Transkription anfordern, Amazon Chime beginnt das Streamen von Audio an Amazon Transcribe und setzt dies für die Dauer des Anrufs fort.

Das Amazon Chime SDK verwendet seinen „Active Talker“-Algorithmus, um die beiden aktivsten Sprecher auszuwählen, und sendet deren Audio dann über einen einzigen Stream Amazon Transcribe an zwei separate Kanäle. Besprechungsteilnehmer erhalten vom Benutzer zugeordnete Transkriptionen über SDK-Datennachrichten. Amazon Chime Beispiele für die Auslieferung finden Sie im [Amazon Chime -SDK-Entwicklerhandbuch](#).

Der Datenfluss einer Amazon Chime Transkription ist in der folgenden Abbildung dargestellt:



Weitere Informationen und detaillierte Anweisungen zur Einrichtung von Amazon Chime Echtzeit-Transkriptionen finden Sie unter [Verwenden der Amazon Chime SDK-Live-Transkription im SDK-Entwicklerhandbuch](#). Amazon Chime Informationen zu API-Vorgängen finden Sie in der [Amazon Chime -SDK-API-Referenz](#).

i Tauchen Sie tiefer ein mit dem AWS Blog zum Machine Learning

Weitere Informationen zur Verbesserung der Genauigkeit von Echtzeit-Transkriptionen finden Sie hier:

- [Amazon Chime SDK-Meetings unterstützen jetzt Live-Transkription mit und Amazon TranscribeAmazon Transcribe Medical](#)
- [Amazon Chime -SDK für telemedizinische Lösung](#)

Codebeispiele für Amazon Transcribe mit AWS SDKs

Die folgenden Codebeispiele zeigen, wie Amazon Transcribe mit einem AWS Software Development Kit (SDK) verwendet wird.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Während Aktionen Ihnen zeigen, wie Sie einzelne Service-Funktionen aufrufen, können Sie Aktionen im Kontext der zugehörigen Szenarien anzeigen.

Szenarien sind Codebeispiele, die Ihnen zeigen, wie Sie bestimmte Aufgaben ausführen, indem Sie mehrere Funktionen innerhalb eines Service aufrufen oder mit anderen AWS-Services kombinieren.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter.

[Verwenden Sie diesen Dienst mit einem AWS SDK](#) Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Codebeispiele

- [Grundlegende Beispiele für die Verwendung von Amazon Transcribe AWS SDKs](#)
 - [Aktionen für Amazon Transcribe mit AWS SDKs](#)
 - [Verwenden Sie CreateVocabulary mit einem AWS SDK oder CLI](#)
 - [Verwenden Sie DeleteMedicalTranscriptionJob mit einem AWS SDK oder CLI](#)
 - [Verwenden Sie DeleteTranscriptionJob mit einem AWS SDK oder CLI](#)
 - [Verwenden Sie DeleteVocabulary mit einem AWS SDK oder CLI](#)
 - [Verwenden Sie GetTranscriptionJob mit einem AWS SDK oder CLI](#)
 - [Verwenden Sie GetVocabulary mit einem AWS SDK oder CLI](#)
 - [Verwenden Sie ListMedicalTranscriptionJobs mit einem AWS SDK oder CLI](#)
 - [Verwenden Sie ListTranscriptionJobs mit einem AWS SDK oder CLI](#)
 - [Verwenden Sie ListVocabularies mit einem AWS SDK oder CLI](#)
 - [Verwenden Sie StartMedicalTranscriptionJob mit einem AWS SDK oder CLI](#)
 - [Verwenden Sie StartTranscriptionJob mit einem AWS SDK oder CLI](#)
 - [Verwenden Sie UpdateVocabulary mit einem AWS SDK oder CLI](#)
- [Szenarien für die Verwendung von Amazon Transcribe AWS SDKs](#)
 - [Erstellen einer Amazon-Transcribe-Streaming-App](#)
 - [Konvertieren Sie Text in Sprache und zurück in Text mithilfe eines AWS SDK](#)

- [Erstellen und verfeinern Sie ein benutzerdefiniertes Amazon Transcribe Transcribe-Vokabular mit einem AWS SDK](#)
- [Transkribieren Sie Audio und rufen Sie Auftragsdaten mit Amazon Transcribe mithilfe eines AWS SDK](#)

Grundlegende Beispiele für die Verwendung von Amazon Transcribe AWS SDKs

Die folgenden Codebeispiele zeigen, wie die Grundlagen von Amazon Transcribe mit AWS SDKs verwendet werden.

Beispiele

- [Aktionen für Amazon Transcribe mit AWS SDKs](#)
 - [Verwenden Sie CreateVocabulary mit einem AWS SDK oder CLI](#)
 - [Verwenden Sie DeleteMedicalTranscriptionJob mit einem AWS SDK oder CLI](#)
 - [Verwenden Sie DeleteTranscriptionJob mit einem AWS SDK oder CLI](#)
 - [Verwenden Sie DeleteVocabulary mit einem AWS SDK oder CLI](#)
 - [Verwenden Sie GetTranscriptionJob mit einem AWS SDK oder CLI](#)
 - [Verwenden Sie GetVocabulary mit einem AWS SDK oder CLI](#)
 - [Verwenden Sie ListMedicalTranscriptionJobs mit einem AWS SDK oder CLI](#)
 - [Verwenden Sie ListTranscriptionJobs mit einem AWS SDK oder CLI](#)
 - [Verwenden Sie ListVocabularies mit einem AWS SDK oder CLI](#)
 - [Verwenden Sie StartMedicalTranscriptionJob mit einem AWS SDK oder CLI](#)
 - [Verwenden Sie StartTranscriptionJob mit einem AWS SDK oder CLI](#)
 - [Verwenden Sie UpdateVocabulary mit einem AWS SDK oder CLI](#)

Aktionen für Amazon Transcribe mit AWS SDKs

Die folgenden Codebeispiele zeigen, wie einzelne Amazon Transcribe Transcribe-Aktionen mit AWS SDKs durchgeführt werden. Jedes Beispiel enthält einen Link zu GitHub, über den Sie Anweisungen zum Einrichten und Ausführen des Codes finden.

Diese Auszüge rufen die Amazon Transcribe API auf und sind Codeauszüge aus größeren Programmen, die im Kontext ausgeführt werden müssen. Sie können Aktionen im Kontext unter [Szenarien für die Verwendung von Amazon Transcribe AWS SDKs](#) anzeigen.

Die folgenden Beispiele enthalten nur die am häufigsten verwendeten Aktionen. Eine vollständige Liste finden Sie in der [Amazon Transcribe API-Referenz](#).

Beispiele

- [Verwenden Sie CreateVocabularymit einem AWS SDK oder CLI](#)
- [Verwenden Sie DeleteMedicalTranscriptionJobmit einem AWS SDK oder CLI](#)
- [Verwenden Sie DeleteTranscriptionJobmit einem AWS SDK oder CLI](#)
- [Verwenden Sie DeleteVocabularymit einem AWS SDK oder CLI](#)
- [Verwenden Sie GetTranscriptionJobmit einem AWS SDK oder CLI](#)
- [Verwenden Sie GetVocabularymit einem AWS SDK oder CLI](#)
- [Verwenden Sie ListMedicalTranscriptionJobsmit einem AWS SDK oder CLI](#)
- [Verwenden Sie ListTranscriptionJobsmit einem AWS SDK oder CLI](#)
- [Verwenden Sie ListVocabulariesmit einem AWS SDK oder CLI](#)
- [Verwenden Sie StartMedicalTranscriptionJobmit einem AWS SDK oder CLI](#)
- [Verwenden Sie StartTranscriptionJobmit einem AWS SDK oder CLI](#)
- [Verwenden Sie UpdateVocabularymit einem AWS SDK oder CLI](#)

Verwenden Sie **CreateVocabulary**mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie CreateVocabulary verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Erstellen und Verfeinern eines benutzerdefinierten Vokabulars](#)

.NET

SDK für .NET

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
/// <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;
}
```

- Einzelheiten zur API finden Sie [CreateVocabulary](#) in der AWS SDK für .NET API-Referenz.

CLI

AWS CLI

Erstellen eines benutzerdefinierten Vokabulars

Im folgenden Beispiel für `create-vocabulary` wird ein benutzerdefiniertes Vokabular erstellt. Um ein benutzerdefiniertes Vokabular zu erstellen, müssen Sie eine Textdatei mit allen Begriffen erstellt haben, die Sie genauer transkribieren möchten. Geben Sie für „`vocabulary-file-uri`“ den Amazon Simple Storage Service (Amazon S3)-URI der betreffenden Textdatei an. Geben Sie für „`language-code`“ den der Sprache Ihres benutzerdefinierten Vokabulars entsprechenden Sprachcode an. Geben Sie für „`vocabulary-name`“ die gewünschte Bezeichnung für Ihr benutzerdefiniertes Vokabular an.

```
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
```

Ausgabe:

```
{  
  "VocabularyName": "cli-vocab-example",  
  "LanguageCode": "language-code",  
  "VocabularyState": "PENDING"  
}
```

Weitere Informationen finden Sie unter [Benutzerdefinierte Vokabulare](#) im Amazon-Transcribe-Entwicklerhandbuch.

- Einzelheiten zur API finden Sie [CreateVocabulary](#) in der AWS CLI Befehlsreferenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
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

```

- Einzelheiten zur API finden Sie [CreateVocabulary](#) in AWS SDK for Python (Boto3) API Reference.

SAP ABAP

SDK für SAP ABAP

Note

Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
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.
```

- Einzelheiten zur API finden Sie [CreateVocabulary](#) in der API-Referenz zum AWS SDK für SAP ABAP.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem AWS SDK](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **DeleteMedicalTranscriptionJob** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie DeleteMedicalTranscriptionJob verwendet wird.

.NET

SDK für .NET

 Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
/// <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;
}
```

- Einzelheiten zur API finden Sie [DeleteMedicalTranscriptionJob](#) in der AWS SDK für .NET API-Referenz.

CLI

AWS CLI

Löschen eines medizinischen Transkriptionsauftrags

Im folgenden Beispiel für `delete-medical-transcription-job` wird ein medizinischer Transkriptionsauftrag gelöscht.

```
aws transcribe delete-medical-transcription-job \
  --medical-transcription-job-name medical-transcription-job-name
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [DeleteMedicalTranscriptionJob](#) im Amazon Transcribe Developer Guide.

- Einzelheiten zur API finden Sie [DeleteMedicalTranscriptionJob](#) in der AWS CLI Befehlsreferenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

Erstellen Sie den Client.

```
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 };
```

Einen Auftrag für medizinische Transkription löschen.

```
// 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();
```

- Weitere Informationen finden Sie im [AWS SDK für JavaScript -Entwicklerhandbuch](#).
- Einzelheiten zur API finden Sie [DeleteMedicalTranscriptionJob](#) in der AWS SDK für JavaScript API-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem AWS SDK](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **DeleteTranscriptionJob** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie DeleteTranscriptionJob verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Erstellen und Verfeinern eines benutzerdefinierten Vokabulars](#)

.NET

SDK für .NET

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
/// <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;
}
```

- Einzelheiten zur API finden Sie [DeleteTranscriptionJob](#) in der AWS SDK für .NET API-Referenz.

CLI

AWS CLI

Löschen eines Ihrer Transkriptionsaufträge

Im folgenden Beispiel für `delete-transcription-job` wird einer Ihrer Transkriptionsaufträge gelöscht.

```
aws transcribe delete-transcription-job \  
  --transcription-job-name your-transcription-job
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie [DeleteTranscriptionJob](#) im Amazon Transcribe Developer Guide.

- Einzelheiten zur API finden Sie [DeleteTranscriptionJob](#) in der AWS CLI Befehlsreferenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

Löschen eines Transkriptionsauftrags.

```
// 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();
```

Erstellen Sie den Client.

```
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 };
```

- Weitere Informationen finden Sie im [AWS SDK für JavaScript -Entwicklerhandbuch](#).
- Einzelheiten zur API finden Sie [DeleteTranscriptionJob](#) in der AWS SDK für JavaScript API-Referenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
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

```

- Einzelheiten zur API finden Sie [DeleteTranscriptionJob](#) in AWS SDK for Python (Boto3) API Reference.

SAP ABAP

SDK für SAP ABAP

Note

Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```

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_tnblimitexceedex 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.

```

- Einzelheiten zur API finden Sie [DeleteTranscriptionJob](#) in der API-Referenz zum AWS SDK für SAP ABAP.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem AWS SDK](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **DeleteVocabulary** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie DeleteVocabulary verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Erstellen und Verfeinern eines benutzerdefinierten Vokabulars](#)

.NET

SDK für .NET

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
/// <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;
}
```

- Einzelheiten zur API finden Sie [DeleteVocabulary](#) in der AWS SDK für .NET API-Referenz.

CLI

AWS CLI

So löschen Sie ein benutzerdefiniertes Vokabular

Im folgenden Beispiel für `delete-vocabulary` wird ein benutzerdefiniertes Vokabular gelöscht.

```
aws transcribe delete-vocabulary \
  --vocabulary-name vocabulary-name
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Benutzerdefinierte Vokabulare](#) im Amazon-Transcribe-Entwicklerhandbuch.

- Einzelheiten zur API finden Sie [DeleteVocabulary](#) in der AWS CLI Befehlsreferenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
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

```

- Einzelheiten zur API finden Sie [DeleteVocabulary](#) in AWS SDK for Python (Boto3) API Reference.

SAP ABAP

SDK für SAP ABAP

Note

Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```

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_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.
ENDTRY.

```

- Einzelheiten zur API finden Sie [DeleteVocabulary](#) in der API-Referenz zum AWS SDK für SAP ABAP.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem AWS SDK](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **GetTranscriptionJob** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie GetTranscriptionJob verwendet wird.

Aktionsbeispiele sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Sie können diese Aktion in den folgenden Codebeispielen im Kontext sehen:

- [Erstellen und Verfeinern eines benutzerdefinierten Vokabulars](#)
- [Audio transkribieren und Auftragsdaten abrufen](#)

.NET

SDK für .NET

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
/// <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;
}
```


- Einzelheiten zur API finden Sie [GetTranscriptionJob](#) in der AWS SDK für .NET API-Referenz.

CLI

AWS CLI

Abrufen von Informationen zu einem bestimmten Transkriptionsauftrag

Im folgenden Beispiel für `get-transcription-job` werden Informationen zu einem bestimmten Transkriptionsauftrag abgerufen. Verwenden Sie den `TranscriptFileUri` Parameter, um auf die Transkriptionsergebnisse zuzugreifen. Verwenden Sie den `MediaFileUri` Parameter, um zu sehen, welche Audiodatei Sie mit diesem Job transkribiert haben. Sie können das Objekt „Settings“ verwenden, um die optionalen Features zu sehen, die Sie im Transkriptionsauftrag aktiviert haben.

```
aws transcribe get-transcription-job \
  --transcription-job-name your-transcription-job
```

Ausgabe:

```
{
  "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
}
}
```

Weitere Informationen finden Sie unter [Erste Schritte \(AWS Befehlszeilenschnittstelle\)](#) im Amazon Transcribe Developer Guide.

- Einzelheiten zur API finden Sie unter [GetTranscriptionJob AWS CLI](#) Befehlsreferenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
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
```

- Einzelheiten zur API finden Sie [GetTranscriptionJob](#) in AWS SDK for Python (Boto3) API Reference.

SAP ABAP

SDK für SAP ABAP

Note

Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
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.
```

- Einzelheiten zur API finden Sie [GetTranscriptionJob](#) in der API-Referenz zum AWS SDK für SAP ABAP.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem AWS SDK](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **GetVocabulary** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `GetVocabulary` verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Erstellen und Verfeinern eines benutzerdefinierten Vokabulars](#)

.NET

SDK für .NET

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
/// <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;
}
```

- Einzelheiten zur API finden Sie [GetVocabulary](#) in der AWS SDK für .NET API-Referenz.

CLI

AWS CLI

Abrufen von Informationen zu einem benutzerdefinierten Vokabular

Im folgenden Beispiel für `get-vocabulary` werden Informationen zu einem zuvor erstellten benutzerdefinierten Vokabular abgerufen.

```
aws transcribe get-vocabulary \
  --vocabulary-name cli-vocab-1
```

Ausgabe:

```
{
  "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"
}
```

Weitere Informationen finden Sie unter [Benutzerdefinierte Vokabulare](#) im Amazon-Transcribe-Entwicklerhandbuch.

- Einzelheiten zur API finden Sie [GetVocabulary](#) in der AWS CLI Befehlsreferenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
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

```

- Einzelheiten zur API finden Sie [GetVocabulary](#) in AWS SDK for Python (Boto3) API Reference.

SAP ABAP

SDK für SAP ABAP

Note

Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```

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.
```

- Einzelheiten zur API finden Sie [GetVocabulary](#) in der API-Referenz zum AWS SDK für SAP ABAP.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem AWS SDK](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **ListMedicalTranscriptionJobs** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie ListMedicalTranscriptionJobs verwendet wird.

.NET

SDK für .NET

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
/// <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;
}

```

- Einzelheiten zur API finden Sie [ListMedicalTranscriptionJobs](#) in der AWS SDK für .NET API-Referenz.

CLI

AWS CLI

Auflisten von medizinischen Transkriptionsaufträgen

Im folgenden `list-medical-transcription-jobs` Beispiel werden die medizinischen Transkriptionsaufträge aufgeführt, die Ihrem AWS Konto und Ihrer Region zugeordnet sind. Um weitere Informationen zu einem bestimmten Transkriptionsauftrag zu erhalten, kopieren Sie den Wert eines `MedicalTranscriptionJobName` Parameters in die Transkriptionsausgabe und geben Sie diesen Wert für die `MedicalTranscriptionJobName` Option des Befehls an. `get-medical-transcription-job` Um mehr Ihrer Transkriptionsaufträge zu sehen, kopieren Sie den Wert des `NextToken` Parameters, führen Sie den `list-medical-transcription-jobs` Befehl erneut aus und geben Sie diesen Wert in der Option an. `--next-token`

```
aws transcribe list-medical-transcription-jobs
```

Ausgabe:

```

{
  "NextToken": "3/PblzkiGhzjER3KHuQt2fmbPLF7cDYafjFMEoGn440N/
gsuUSTIkGyanvRE6WMXfd/ZTEc2EZj+P9eii/
z102FDYli6RLI0WoRX4RwMisVrh9G0Kie0Y8ikBCdtqLZB10Wa9McC+eb0l
+LaDtZPC4u6ttoHLRL1EfzqstHXSgapXg3tEBtm9piIaPB6M0M5BB6t86+qtmocTR/
qrteHZBBudhTfbCwhsxaqujHiiUvFdm3BQbKKWIW06yV9b+4f38oD2lVIan
+vfUs3gBYA15VTDmXXzQPBQ0HPjtwmFI+IWX15nSUjWuN3TUylHgPWzDaYT8qBtu0Z+3UG4V6b
+K2CC0XszXg5rBq9hYgNzy4XoFh/6s5DoSnzq49Q9xHgHdT2yBADFmvFK7myZBsJ75+2vQZ0SVpWUPy3WT/32zFAc
+mFYfUjtTZ8n/jq7aQEjQ42A
+X/7K6Jg0cdVPtEg8P1Dr5kgYYG3q30mYXX37U3FZuJmnTI63VtIXsNn0U5eGoY0btpk00Nq9UkzgSJxqj84ZD5n
+S0EGy9ZUYBJRRcGeYUM3Q4DbSJfUwSAqcFdLIWZdp8qIREMQIBWy7BLwSdyqsQo2vRrd53hm5aWM7SVf6pPq6X/

```



```

IXR5+1eU00D8/coaTT4ES2DerbV6RkV4o0VT1d0SdVX/
MmtkNG8nYj8PqU07w7988quh1ZP6D80veJS1q73tUUR9MjnGernW2tAnvnLNhdefBcD
+sZvfYq3iBMFY7wTy1P1G6NqW9GrYDYox3tTPW1D7phpbVSyKrh/
PdYrps5UxnsGoA1b7L/FfAXDfUoGrGUB4N3JsPYXX9D++g+6gV1qBBs/
WffF934aKqfD6UTggm/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"
    }
]
}

```

Weitere Informationen finden Sie unter <https://docs.aws.amazon.com/transcribe/latest/dg/batch-med-transcription.html> > im Amazon Transcribe Developer Guide.

- Einzelheiten zur API finden Sie [ListMedicalTranscriptionJobs](#) in der AWS CLI Befehlsreferenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

Erstellen Sie den Client.

```
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 };
```

Auflisten medizinischer Transkriptionsjobs.

```
// 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();
```

- Weitere Informationen finden Sie im [AWS SDK für JavaScript -Entwicklerhandbuch](#).
- Einzelheiten zur API finden Sie [ListMedicalTranscriptionJobs](#) in der AWS SDK für JavaScript API-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem AWS SDK](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **ListTranscriptionJobs** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie ListTranscriptionJobs verwendet wird.

.NET

SDK für .NET

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
/// <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;
}
```

- Einzelheiten zur API finden Sie [ListTranscriptionJobs](#) in der AWS SDK für .NET API-Referenz.

CLI

AWS CLI

Auflisten Ihrer Transkriptionsaufträge

Das folgende `list-transcription-jobs` Beispiel listet die Transkriptionsaufträge auf, die Ihrem AWS Konto und Ihrer Region zugeordnet sind.

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

Ausgabe:

```
{
  "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"
    }
]
}

```

Weitere Informationen finden Sie unter [Erste Schritte \(AWS Befehlszeilenschnittstelle\)](#) im Amazon Transcribe Developer Guide.

- Einzelheiten zur API finden Sie unter [ListTranscriptionJobs AWS CLI](#) Befehlsreferenz.

Java

SDK für Java 2.x

 Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
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("-----");
    });
}
}

```

- Einzelheiten zur API finden Sie [ListTranscriptionJobs](#) in der AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

Auflisten von Transkriptionsaufträgen.

```

// 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();
```

Erstellen Sie den Client.

```
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 };
```

- Weitere Informationen finden Sie im [AWS SDK für JavaScript -Entwicklerhandbuch](#).
- Einzelheiten zur API finden Sie [ListTranscriptionJobs](#) in der AWS SDK für JavaScript API-Referenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
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

```

- Einzelheiten zur API finden Sie [ListTranscriptionJobs](#) in AWS SDK for Python (Boto3) API Reference.

SAP ABAP

SDK für SAP ABAP

Note

Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
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.
```

- Einzelheiten zur API finden Sie [ListTranscriptionJobs](#) in der API-Referenz zum AWS SDK für SAP ABAP.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem AWS SDK](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **ListVocabularies** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `ListVocabularies` verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Erstellen und Verfeinern eines benutzerdefinierten Vokabulars](#)

.NET

SDK für .NET

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
/// <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;
}
```

- Einzelheiten zur API finden Sie [ListVocabularies](#) in der AWS SDK für .NET API-Referenz.

CLI

AWS CLI

Auflisten Ihrer benutzerdefinierten Vokabulare

Das folgende `list-vocabularies` Beispiel listet die benutzerdefinierten Vokabulare auf, die mit Ihrem AWS Konto und Ihrer Region verknüpft sind.

```
aws transcribe list-vocabularies
```

Ausgabe:

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

Weitere Informationen finden Sie unter [Benutzerdefinierte Vokabulare](#) im Amazon-Transcribe-Entwicklerhandbuch.

- Einzelheiten zur API finden Sie [ListVocabularies](#) in der AWS CLI Befehlsreferenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
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

```

- Einzelheiten zur API finden Sie [ListVocabularies](#) in AWS SDK for Python (Boto3) API Reference.

SAP ABAP

SDK für SAP ABAP

Note

Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```

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.

```

- Einzelheiten zur API finden Sie [ListVocabularies](#) in der API-Referenz zum AWS SDK für SAP ABAP.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem AWS SDK](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **StartMedicalTranscriptionJob** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie StartMedicalTranscriptionJob verwendet wird.

.NET

SDK für .NET

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```

    /// <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;
}

```

- Einzelheiten zur API finden Sie [StartMedicalTranscriptionJob](#) in der AWS SDK für .NET API-Referenz.

CLI

AWS CLI

Beispiel 1: Transkribieren eines als Audiodatei gespeicherten medizinischen Diktats

Im folgenden Beispiel für `start-medical-transcription-job` wird eine Audiodatei transkribiert. Sie geben den Speicherort der Transkriptionsausgabe im Parameter `OutputBucketName` an.

```
aws transcribe start-medical-transcription-job \
    --cli-input-json file://myfile.json
```

Inhalt von `myfile.json`:

```
{
  "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"
  }
}
```

Ausgabe:

```
{
  "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"
  }
}
```

Weitere Informationen finden Sie unter [Übersicht über die Batch-Transkription](#) im Amazon-Transcribe-Entwicklerhandbuch.

Beispiel 2: Transkribieren eines als Audiodatei gespeicherten Dialogs zwischen Arzt und Patient

Im folgenden Beispiel für `start-medical-transcription-job` wird eine Audiodatei mit einem Dialog zwischen Arzt und Patient transkribiert. Sie geben den Speicherort der Transkriptionsausgabe im `OutputBucketName` Parameter an.

```
aws transcribe start-medical-transcription-job \
  --cli-input-json file://mysecondfile.json
```

Inhalt von `mysecondfile.json`:

```
{
  "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"
  }
}
```

Ausgabe:

```
{
  "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"
  }
}
```

Weitere Informationen finden Sie unter [Übersicht über die Batch-Transkription](#) im Amazon-Transcribe-Entwicklerhandbuch.

Beispiel 3: Transkribieren einer Mehrkanal-Audiodatei eines Dialogs zwischen Arzt und Patient

Im folgenden Beispiel für `start-medical-transcription-job` werden die Audiodaten aus jedem Kanal in der Audiodatei transkribiert und die einzelnen Transkriptionen von jedem Kanal zu einer einzigen Transkriptionsausgabe zusammengeführt. Sie geben den Speicherort der Transkriptionsausgabe im Parameter `OutputBucketName` an.

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

```
--cli-input-json file://mythirdfile.json
```

Inhalt von `mythirdfile.json`:

```
{
  "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
  }
}
```

Ausgabe:

```
{
  "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"
  }
}
```

Weitere Informationen finden Sie unter [Kanalidentifizierung](#) im Amazon-Transcribe-Entwicklerhandbuch.

Beispiel 4: Transkribieren einer Audiodatei eines Dialogs zwischen Arzt und Patient und Identifizieren der Sprecher in der Transkriptionsausgabe

Im folgenden Beispiel für `start-medical-transcription-job` wird eine Audiodatei transkribiert und die Sprache der einzelnen Sprecher wird in der Transkriptionsausgabe gekennzeichnet. Sie geben den Speicherort der Transkriptionsausgabe im Parameter `OutputBucketName` an.

```
aws transcribe start-medical-transcription-job \  
  --cli-input-json file://myfourthfile.json
```

Inhalt von `myfourthfile.json`:

```
{  
  "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  
  }  
}
```

Ausgabe:

```
{  
  "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"
  }
}
```

Weitere Informationen finden Sie unter [Identifizieren von Sprechern](#) im Amazon-Transcribe-Entwicklerhandbuch.

Beispiel 5: Transkribieren eines als Audiodatei gespeicherten medizinischen Gesprächs mit bis zu zwei Transkriptionsalternativen

Im folgenden Beispiel für `start-medical-transcription-job` werden bis zu zwei alternative Transkriptionen aus einer einzigen Audiodatei erstellt. Jeder Transkription ist ein gewisses Konfidenzniveau zugeordnet. Standardmäßig gibt Amazon Transcribe die Transkription mit dem höchsten Konfidenzniveau zurück. Sie können angeben, dass Amazon Transcribe zusätzliche Transkriptionen mit niedrigerem Konfidenzniveau zurückgeben soll. Sie geben den Speicherort der Transkriptionsausgabe im Parameter `OutputBucketName` an.

```
aws transcribe start-medical-transcription-job \
  --cli-input-json file://myfifthfile.json
```

Inhalt von `myfifthfile.json`:

```
{
  "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
  }
}
```

```
}
```

Ausgabe:

```
{
  "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"
  }
}
```

Weitere Informationen finden Sie unter [Alternative Transkriptionen](#) im Amazon-Transcribe-Entwicklerhandbuch.

Beispiel 6: Transkribieren einer Audiodatei eines medizinischen Diktats mit bis zu zwei alternativen Transkriptionen

Im folgenden Beispiel für `start-medical-transcription-job` wird eine Audiodatei transkribiert und zum Maskieren von unerwünschten Wörtern wird ein Vokabularfilter verwendet. Sie geben den Ort der Transkriptionsausgabe im Parameter an `OutputBucketName`.

```
aws transcribe start-medical-transcription-job \
  --cli-input-json file://mysixthfile.json
```

Inhalt von `mysixthfile.json`:

```
{
```

```

    "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
    }
}

```

Ausgabe:

```

{
    "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"
    }
}

```

Weitere Informationen finden Sie unter [Alternative Transkriptionen](#) im Amazon-Transcribe-Entwicklerhandbuch.

Beispiel 7: Transkribieren einer Audiodatei eines medizinischen Diktats mit höherer Genauigkeit durch Verwendung eines benutzerdefinierten Vokabulars

Im folgenden Beispiel für `start-medical-transcription-job` wird eine Audiodatei transkribiert und zur Verbesserung der Transkriptionsgenauigkeit wird ein zuvor von Ihnen erstelltes benutzerdefiniertes medizinisches Vokabular verwendet. Sie geben den Speicherort der Transkriptionsausgabe im Parameter `OutputBucketName` an.

```
aws transcribe start-transcription-job \  
  --cli-input-json file://myseventhfile.json
```

Inhalt von `mysixthfile.json`:

```
{  
  "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"  
  }  
}
```

Ausgabe:

```
{  
  "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"
  }
}
```

Weitere Informationen finden Sie unter [Benutzerdefinierte medizinische Vokabulare](#) im Amazon-Transcribe-Entwicklerhandbuch.

- Einzelheiten zur API finden Sie unter [StartMedicalTranscriptionJob AWS CLIBefehlsreferenz](#).

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

Erstellen Sie den Client.

```
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 };
```

Einen Auftrag für medizinische Transkription starten.

```
// 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();

```

- Weitere Informationen finden Sie im [AWS SDK für JavaScript -Entwicklerhandbuch](#).
- Einzelheiten zur API finden Sie [StartMedicalTranscriptionJob](#) in der AWS SDK für JavaScript API-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem AWS SDK](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **StartTranscriptionJob** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie StartTranscriptionJob verwendet wird.

Aktionsbeispiele sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Sie können diese Aktion in den folgenden Codebeispielen im Kontext sehen:

- [Erstellen und Verfeinern eines benutzerdefinierten Vokabulars](#)
- [Audio transkribieren und Auftragsdaten abrufen](#)

.NET

SDK für .NET

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
/// <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;
}

```

- Einzelheiten zur API finden Sie [StartTranscriptionJob](#) in der AWS SDK für .NET API-Referenz.

CLI

AWS CLI

Beispiel 1: Transkribieren einer Audiodatei

Im folgenden Beispiel für `start-transcription-job` wird Ihre Audiodatei transkribiert.

```
aws transcribe start-transcription-job \
  --cli-input-json file://myfile.json
```

Inhalt von `myfile.json`:

```
{
  "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"
  }
}
```

Weitere Informationen finden Sie unter [Erste Schritte \(AWS Befehlszeilenschnittstelle\)](#) im Amazon Transcribe Developer Guide.

Beispiel 2: Transkribieren einer Mehrkanal-Audiodatei

Im folgenden Beispiel für `start-transcription-job` wird Ihre Mehrkanal-Audiodatei transkribiert.

```
aws transcribe start-transcription-job \  
  --cli-input-json file://mysecondfile.json
```

Inhalt von `mysecondfile.json`:

```
{  
  "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  
  }  
}
```

Ausgabe:

```
{  
  "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  
    }  
  }  
}
```

Weitere Informationen finden Sie unter [Multi-Channel Audio transkribieren](#) im Amazon Transcribe Developer Guide.

Beispiel 3: Transkribieren einer Audiodatei und Identifizieren der verschiedenen Sprecher

Im folgenden Beispiel für `start-transcription-job` wird Ihre Audiodatei transkribiert und die Sprecher werden in der Transkriptionsausgabe identifiziert.

```
aws transcribe start-transcription-job \  
  --cli-input-json file:///mythirdfile.json
```

Inhalt von `mythirdfile.json`:

```
{  
  "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  
  }  
}
```

Ausgabe:

```
{  
  "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  
    }  
  }  
}
```

Weitere Informationen finden Sie unter [Identifizieren von Sprechern](#) im Amazon-Transcribe-Entwicklerhandbuch.

Beispiel 4: Transkribieren einer Audiodatei und Maskieren aller unerwünschten Wörter in der Transkriptionsausgabe

Im folgenden Beispiel für `start-transcription-job` wird Ihrer Audiodatei transkribiert und zum Maskieren von unerwünschten Wörtern wird ein zuvor von Ihnen erstellter Vokabularfilter verwendet.

```
aws transcribe start-transcription-job \  
  --cli-input-json file://myfourthfile.json
```

Inhalt von `myfourthfile.json`:

```
{  
  "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"  
  }  
}
```

Ausgabe:

```
{  
  "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"
    }
}

```

Weitere Informationen finden Sie unter [Filtern von Transkriptionen](#) im Amazon-Transcribe-Entwicklerhandbuch.

Beispiel 5: Transkribieren einer Audiodatei und Entfernen aller unerwünschten Wörter aus der Transkriptionsausgabe

Im folgenden Beispiel für `start-transcription-job` wird Ihrer Audiodatei transkribiert und zum Maskieren von unerwünschten Wörtern wird ein zuvor von Ihnen erstellter Vokabularfilter verwendet.

```

aws transcribe start-transcription-job \
  --cli-input-json file://myfifthfile.json

```

Inhalt von `myfifthfile.json`:

```

{
  "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"
  }
}

```

Ausgabe:

```

{
  "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"
    }
}
}

```

Weitere Informationen finden Sie unter [Filtern von Transkriptionen](#) im Amazon-Transcribe-Entwicklerhandbuch.

Beispiel 6: Transkribieren einer Audiodatei mit höherer Genauigkeit durch Verwendung eines benutzerdefinierten Vokabulars

Im folgenden Beispiel für `start-transcription-job` wird Ihrer Audiodatei transkribiert und zum Maskieren von unerwünschten Wörtern wird ein zuvor von Ihnen erstellter Vokabularfilter verwendet.

```

aws transcribe start-transcription-job \
  --cli-input-json file://mysixthfile.json

```

Inhalt von `mysixthfile.json`:

```

{
    "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"
    }
}

```

Ausgabe:

```

{

```

```

"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"
  }
}
}

```

Weitere Informationen finden Sie unter [Filtern von Transkriptionen](#) im Amazon-Transcribe-Entwicklerhandbuch.

Beispiel 7: Identifizieren der Sprache einer Audiodatei und Transkribieren der Datei

Im folgenden Beispiel für `start-transcription-job` wird Ihrer Audiodatei transkribiert und zum Maskieren von unerwünschten Wörtern wird ein zuvor von Ihnen erstellter Vokabularfilter verwendet.

```

aws transcribe start-transcription-job \
  --cli-input-json file://myseventhfile.json

```

Inhalt von `myseventhfile.json`:

```

{
  "TranscriptionJobName": "cli-identify-language-transcription-job",
  "IdentifyLanguage": true,
  "Media": {
    "MediaFileUri": "s3://amzn-s3-demo-bucket/Amazon-S3-prefix/your-media-
file-name.file-extension"
  }
}

```

Ausgabe:

```

{

```

```

    "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
    }
  }
}

```

Weitere Informationen finden Sie unter [Identifizieren der Sprache](#) im Amazon-Transcribe-Entwicklerhandbuch.

Beispiel 8: Transkribieren einer Audiodatei mit unkenntlich gemachten persönlich identifizierbaren Informationen

Im folgenden Beispiel für `start-transcription-job` wird Ihre Audiodatei transkribiert und die persönlich identifizierbaren Informationen werden in der Transkriptionsausgabe unkenntlich gemacht.

```

aws transcribe start-transcription-job \
  --cli-input-json file://myeighthfile.json

```

Inhalt von `myeighthfile.json`:

```

{
  "TranscriptionJobName": "cli-redaction-job",
  "LanguageCode": "language-code",
  "Media": {
    "MediaFileUri": "s3://Amazon-S3-Prefix/your-media-file.file-extension"
  },
  "ContentRedaction": {
    "RedactionOutput": "redacted",
    "RedactionType": "PII"
  }
}

```

Ausgabe:

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

Weitere Informationen finden Sie unter [Automatische Inhaltsschwärzung](#) im Amazon-Transcribe-Entwicklerhandbuch.

Beispiel 9: Generieren eines Transkripts mit unkenntlich gemachten persönlich identifizierbaren Informationen (PII) und eines ungeschwärzten Transkripts

Im folgenden Beispiel für `start-transcription-job` werden zwei Transkriptionen Ihrer Audiodatei generiert, eine mit unkenntlich gemachten persönlich identifizierbaren Informationen und die andere ohne Schwärzungen.

```
aws transcribe start-transcription-job \
  --cli-input-json file://myninthfile.json
```

Inhalt von `myninthfile.json`:

```
{
  "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"
  }
}

```

Ausgabe:

```

{
  "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"
    }
  }
}

```

Weitere Informationen finden Sie unter [Automatische Inhaltsschwärzung](#) im Amazon-Transcribe-Entwicklerhandbuch.

Beispiel 10: Verwenden eines benutzerdefinierten Sprachmodells, das Sie zuvor erstellt haben, um eine Audiodatei zu transkribieren

Im folgenden Beispiel für `start-transcription-job` wird Ihre Audiodatei mit einem benutzerdefinierten Sprachmodell transkribiert, das Sie zuvor erstellt haben.

```

aws transcribe start-transcription-job \
  --cli-input-json file://mytenthfile.json

```

Inhalt von `mytenthfile.json`:

```

{
  "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"
}
}
```

Ausgabe:

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

Weitere Informationen finden Sie unter [Verbessern der Domain-Specific Transkriptionsgenauigkeit mit benutzerdefinierten Sprachmodellen](#) im Amazon Transcribe Developer Guide.

- Einzelheiten zur API finden Sie [StartTranscriptionJob](#) in AWS CLI der Befehlsreferenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

Einen Transkriptionsauftrag starten.

```
// 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();
```

Erstellen Sie den Client.

```
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 };
```


- Weitere Informationen finden Sie im [AWS SDK für JavaScript -Entwicklerhandbuch](#).
- Einzelheiten zur API finden Sie [StartTranscriptionJob](#) in der AWS SDK für JavaScript API-Referenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
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

```

- Einzelheiten zur API finden Sie [StartTranscriptionJob](#) in AWS SDK for Python (Boto3) API Reference.

SAP ABAP

SDK für SAP ABAP

Note

Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```

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.

```

- Einzelheiten zur API finden Sie [StartTranscriptionJob](#) in der API-Referenz zum AWS SDK für SAP ABAP.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem AWS SDK](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **UpdateVocabulary** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie UpdateVocabulary verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Erstellen und Verfeinern eines benutzerdefinierten Vokabulars](#)

.NET

SDK für .NET

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
/// <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;
}
```

- Einzelheiten zur API finden Sie [UpdateVocabulary](#) in der AWS SDK für .NET API-Referenz.

CLI

AWS CLI

Aktualisieren eines benutzerdefinierten Vokabular mit neuen Begriffen

Im folgenden Beispiel für `update-vocabulary` werden die Begriffe, die zur Erstellung eines benutzerdefinierten Vokabulars verwendet wurden, mit den von Ihnen angegebenen neuen Begriffen überschrieben. Voraussetzung: Um die Begriffe in einem benutzerdefinierten Wortschatz zu ersetzen, benötigen Sie eine Datei mit neuen Begriffen.

```
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
```

Ausgabe:

```
{  
  "VocabularyName": "custom-vocabulary",  
  "LanguageCode": "language",  
  "VocabularyState": "PENDING"  
}
```

Weitere Informationen finden Sie unter [Benutzerdefinierte Vokabulare](#) im Amazon-Transcribe-Entwicklerhandbuch.

- Einzelheiten zur API finden Sie [UpdateVocabulary](#) in der AWS CLI Befehlsreferenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
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

```

- Einzelheiten zur API finden Sie [UpdateVocabulary](#) in AWS SDK for Python (Boto3) API Reference.

SAP ABAP

SDK für SAP ABAP

Note

Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```

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.

```

- Einzelheiten zur API finden Sie [UpdateVocabulary](#) in der API-Referenz zum AWS SDK für SAP ABAP.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem AWS SDK](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Szenarien für die Verwendung von Amazon Transcribe AWS SDKs

Die folgenden Codebeispiele zeigen Ihnen, wie Sie gängige Szenarien in Amazon Transcribe mit AWS SDKs implementieren. Diese Szenarien zeigen Ihnen, wie Sie bestimmte Aufgaben durch den Aufruf mehrerer Funktionen innerhalb von Amazon Transcribe oder in Kombination mit anderen

AWS-Services ausführen können. Jedes Szenario enthält einen Link zum vollständigen Quell-Code, wo Sie Anleitungen zum Einrichten und Ausführen des Codes finden.

Szenarien zielen auf eine mittlere Erfahrungsebene ab, um Ihnen zu helfen, Service-Aktionen im Kontext zu verstehen.

Beispiele

- [Erstellen einer Amazon-Transcribe-Streaming-App](#)
- [Konvertieren Sie Text in Sprache und zurück in Text mithilfe eines AWS SDK](#)
- [Erstellen und verfeinern Sie ein benutzerdefiniertes Amazon Transcribe Transcribe-Vokabular mit einem AWS SDK](#)
- [Transkribieren Sie Audio und rufen Sie Auftragsdaten mit Amazon Transcribe mithilfe eines AWS SDK](#)

Erstellen einer Amazon-Transcribe-Streaming-App

Das folgende Code-Beispiel zeigt, wie Sie eine App erstellen, die Live-Audio in Echtzeit aufzeichnet, transkribiert und übersetzt und die Ergebnisse per E-Mail sendet.

JavaScript

SDK für JavaScript (v3)

Zeigt, wie Amazon Transcribe verwendet wird, um eine App zu erstellen, die Live-Audio in Echtzeit aufzeichnet, transkribiert und übersetzt und die Ergebnisse mit Amazon Simple Email Service (Amazon SES) per E-Mail sendet.

Den vollständigen Quellcode und Anweisungen zur Einrichtung und Ausführung finden Sie im vollständigen Beispiel unter [GitHub](#).

In diesem Beispiel verwendete Dienste

- Amazon Comprehend
- Amazon SES
- Amazon Transcribe
- Amazon Translate

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem AWS SDK](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Konvertieren Sie Text in Sprache und zurück in Text mithilfe eines AWS SDK

Wie das aussehen kann, sehen Sie am nachfolgenden Beispielcode:

- Verwenden Sie Amazon Polly, um eine Klartext-Eingabedatei (UTF-8) zu einer Audiodatei zu synthetisieren.
- Laden Sie die Audiodatei in einen Amazon-S3-Bucket hoch.
- Konvertieren Sie die Audiodatei mit Amazon Transcribe in Text.
- Zeigen Sie den Text an.

Rust

SDK für Rust

Verwenden Sie Amazon Polly, um eine Nur-Text-Eingabedatei (UTF-8) in eine Audiodatei zu synthetisieren, die Audiodatei in einen Amazon S3 S3-Bucket hochzuladen, Amazon Transcribe zu verwenden, um diese Audiodatei in Text zu konvertieren und den Text anzuzeigen.

Den vollständigen Quellcode und Anweisungen zur Einrichtung und Ausführung finden Sie im vollständigen Beispiel unter. [GitHub](#)

In diesem Beispiel verwendete Dienste

- Amazon Polly
- Amazon S3
- Amazon Transcribe

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem AWS SDK](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Erstellen und verfeinern Sie ein benutzerdefiniertes Amazon Transcribe Transcribe-Vokabular mit einem AWS SDK

Wie das aussehen kann, sehen Sie am nachfolgenden Beispielcode:

- Laden Sie eine Audiodatei auf Amazon S3 hoch.
- Führen Sie einen Amazon Transcribe-Auftrag aus, um die Datei zu transkribieren und die Ergebnisse zu erhalten.
- Erstellen und verfeinern Sie ein benutzerdefiniertes Vokabular, um die Transkriptionsgenauigkeit zu verbessern.
- Führen Sie Aufträge mit benutzerdefinierten Vokabularen aus und erhalten Sie die Ergebnisse.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

Transkribieren Sie eine Audiodatei, die eine Lesung von Jabberwocky von Lewis Carroll enthält. Beginnen Sie damit, Funktionen zu erstellen, die Amazon Transcribe-Aktionen wrappen.

```
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_vocables(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
```

Rufen Sie die Wrapper-Funktionen auf, um Audio ohne ein benutzerdefiniertes Vokabular und anschließend mit verschiedenen Versionen eines benutzerdefinierten Vokabulars zu transkribieren, um bessere Ergebnisse zu erzielen.

```
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!")

```

- Weitere API-Informationen finden Sie in den folgenden Themen der API-Referenz zum AWS SDK für Python (Boto3).
 - [CreateVocabulary](#)
 - [DeleteTranscriptionJob](#)
 - [DeleteVocabulary](#)
 - [GetTranscriptionJob](#)

- [GetVocabulary](#)
- [ListVocabularies](#)
- [StartTranscriptionJob](#)
- [UpdateVocabulary](#)

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem AWS SDK](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Transkribieren Sie Audio und rufen Sie Auftragsdaten mit Amazon Transcribe mithilfe eines AWS SDK

Die folgenden Code-Beispiele veranschaulichen Folgendes:

- Starten Sie einen Transkriptionsauftrag mit Amazon Transcribe.
- Warten Sie, bis der -Auftrag abgeschlossen wurde.
- Ermitteln Sie die URI, unter der das Transkript gespeichert ist.

Weitere Informationen finden Sie unter [Erste Schritte mit Amazon Transcribe](#).

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

Transkribiert eine PCM-Datei.

```
/**
 * 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();  
    }  
}  
}
```

Transkribiert Streaming-Audio vom Mikrofon Ihres Computers.

```
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();
}
}
```

- Weitere API-Informationen finden Sie in den folgenden Themen der AWS SDK for Java 2.x - API-Referenz.
 - [GetTranscriptionJob](#)
 - [StartTranscriptionJob](#)

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel](#) einrichten und ausführen.

```
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()

```

- Weitere API-Informationen finden Sie in den folgenden Themen der API-Referenz zum AWS -SDK für Python (Boto3).
 - [GetTranscriptionJob](#)
 - [StartTranscriptionJob](#)

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem AWS SDK](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Sicherheit bei Amazon Transcribe

Cloud-Sicherheit AWS hat höchste Priorität. Als AWS Kunde profitieren Sie von einer Rechenzentrums- und Netzwerkarchitektur, die auf die Anforderungen der sicherheitssensibelsten Unternehmen zugeschnitten sind.

Sicherheit ist eine gemeinsame Verantwortung von Ihnen AWS und Ihnen. Das [Modell der geteilten Verantwortung](#) beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud:

- **Sicherheit der Cloud:** AWS ist verantwortlich für den Schutz der Infrastruktur, auf der AWS Dienste in der ausgeführt AWS Cloud werden. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Third-party Prüfer testen und verifizieren regelmäßig die Wirksamkeit unserer Sicherheitsmaßnahmen im Rahmen der [AWS](#) . Weitere Informationen zu den Compliance-Programmen, die für gelten Amazon Transcribe, finden Sie unter [AWS Services im Umfang nach Compliance-Programmen AWS](#) .
- **Sicherheit in der Cloud:** Ihre Verantwortung hängt von dem AWS Dienst ab, den Sie nutzen. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der geteilten Verantwortung bei der Verwendung von Amazon Transcribe einsetzen können. In den folgenden Themen erfahren Sie, wie Sie die Konfiguration vornehmen Amazon Transcribe , um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Sie erfahren auch, wie Sie andere AWS Dienste zur Überwachung und Sicherung Ihrer Amazon Transcribe Ressourcen verwenden können.

Topics

- [Identity and Access Management für Amazon Transcribe](#)
- [Datenschutz in Amazon Transcribe](#)
- [Überwachen Amazon Transcribe](#)
- [Konformitätsvalidierung für Amazon Transcribe](#)
- [Resilienz in Amazon Transcribe](#)
- [Infrastruktursicherheit in Amazon Transcribe](#)
- [Schwachstellenanalyse und -management in Amazon Transcribe](#)
- [Bewährte Sicherheitsmethoden für Amazon Transcribe](#)

Identity and Access Management für Amazon Transcribe

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service, den Zugriff auf Ressourcen sicher zu AWS kontrollieren. IAM-Administratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um Ressourcen zu verwenden. Amazon Transcribe IAM ist ein Programm AWS-Service, das Sie ohne zusätzliche Kosten nutzen können.

Themen

- [Zielgruppe](#)
- [Authentifizierung mit Identitäten](#)
- [Verwalten des Zugriffs mit Richtlinien](#)
- [Wie Amazon Transcribe funktioniert mit IAM](#)
- [Cross-service Prävention verwirrter Abgeordneter](#)
- [Amazon Transcribe Beispiele für identitätsbasierte Politik](#)
- [Fehlerbehebung Amazon Transcribe Identität und Zugriff](#)

Zielgruppe

Wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von Ihrer Rolle ab:

- Servicebenutzer – Fordern Sie von Ihrem Administrator Berechtigungen an, wenn Sie nicht auf Features zugreifen können (siehe [Fehlerbehebung Amazon Transcribe Identität und Zugriff](#)).
- Serviceadministrator – Bestimmen Sie den Benutzerzugriff und stellen Sie Berechtigungsanfragen (siehe [Wie Amazon Transcribe funktioniert mit IAM](#)).
- IAM-Administrator – Schreiben Sie Richtlinien zur Zugriffsverwaltung (siehe [Amazon Transcribe Beispiele für identitätsbasierte Politik](#)).

Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich AWS mit Ihren Identitätsdaten anmelden. Sie müssen sich als IAM-Benutzer authentifizieren oder eine IAM-Rolle annehmen. Root-Benutzer des AWS-Kontos

Sie können sich als föderierte Identität anmelden, indem Sie Anmeldeinformationen aus einer Identitätsquelle wie AWS IAM Identity Center (IAM Identity Center), Single Sign-On-Authentifizierung oder Anmeldeinformationen verwenden. Google/Facebook Weitere Informationen zum Anmelden finden Sie unter [So melden Sie sich bei Ihrem AWS-Konto an](#) im Benutzerhandbuch für AWS-Anmeldung .

AWS Bietet für den programmatischen Zugriff ein SDK und eine CLI zum kryptografischen Signieren von Anfragen. Weitere Informationen finden Sie unter [AWS Signature Version 4 for API requests](#) im IAM-Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie einen erstellen AWS-Konto, beginnen Sie mit einer Anmeldeidentität, dem sogenannten AWS-Konto Root-Benutzer, der vollständigen Zugriff auf alle AWS-Services Ressourcen hat. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Eine Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter [Tasks that require root user credentials](#) im IAM-Benutzerhandbuch.

Verbundidentität

Als bewährte Methode sollten menschliche Benutzer für den Zugriff AWS-Services mithilfe temporärer Anmeldeinformationen einen Verbund mit einem Identitätsanbieter verwenden.

Eine föderierte Identität ist ein Benutzer aus Ihrem Unternehmensverzeichnis, Ihrem Directory Service Web-Identitätsanbieter oder der AWS-Services mithilfe von Anmeldeinformationen aus einer Identitätsquelle zugreift. Verbundene Identitäten übernehmen Rollen, die temporäre Anmeldeinformationen bereitstellen.

Für die zentrale Zugriffsverwaltung empfehlen wir AWS IAM Identity Center. Weitere Informationen finden Sie unter [Was ist IAM Identity Center?](#) im AWS IAM Identity Center -Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein [IAM-Benutzer](#) ist eine Identität mit bestimmten Berechtigungen für eine einzelne Person oder Anwendung. Wir empfehlen die Verwendung temporärer Anmeldeinformationen anstelle von IAM-Benutzern mit langfristigen Anmeldeinformationen. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter [Erfordern, dass menschliche Benutzer den Verbund mit einem Identitätsanbieter verwenden müssen, um AWS mithilfe temporärer Anmeldeinformationen darauf zugreifen zu](#) können.

Eine [IAM-Gruppe](#) spezifiziert eine Sammlung von IAM-Benutzern und erleichtert die Verwaltung von Berechtigungen für große Gruppen von Benutzern. Weitere Informationen finden Sie unter [Anwendungsfälle für IAM-Benutzer](#) im IAM-Benutzerhandbuch.

IAM-Rollen

Eine [IAM-Rolle](#) ist eine Identität mit spezifischen Berechtigungen, die temporäre Anmeldeinformationen bereitstellt. Sie können eine Rolle übernehmen, indem Sie [von einer Benutzer- zu einer IAM-Rolle \(Konsole\) wechseln](#) AWS CLI oder einen AWS API-Vorgang aufrufen. Weitere Informationen finden Sie unter [Methoden, um eine Rolle zu übernehmen](#) im IAM-Benutzerhandbuch.

IAM-Rollen sind nützlich für den Verbundbenutzer-Zugriff, temporäre IAM-Benutzerberechtigungen, kontoübergreifenden Zugriff, serviceübergreifenden Zugriff und Anwendungen, die auf Amazon EC2 laufen. Weitere Informationen finden Sie unter [Kontoübergreifender Ressourcenzugriff in IAM](#) im IAM-Benutzerhandbuch.

Verwalten des Zugriffs mit Richtlinien

Sie kontrollieren den Zugriff, AWS indem Sie Richtlinien erstellen und diese an AWS Identitäten oder Ressourcen anhängen. Eine Richtlinie definiert Berechtigungen, wenn sie mit einer Identität oder Ressource verknüpft sind. AWS bewertet diese Richtlinien, wenn ein Principal eine Anfrage stellt. Die meisten Richtlinien werden AWS als JSON-Dokumente gespeichert. Weitere Informationen zu JSON-Richtliniendokumenten finden Sie unter [Übersicht über JSON-Richtlinien](#) im IAM-Benutzerhandbuch.

Mit Hilfe von Richtlinien legen Administratoren fest, wer Zugriff auf was hat, indem sie definieren, welches Prinzipal welche Aktionen auf welchen Ressourcen und unter welchen Bedingungendurchführen darf.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator erstellt IAM-Richtlinien und fügt sie zu Rollen hinzu, die die Benutzer dann übernehmen können. IAM-Richtlinien definieren Berechtigungen unabhängig von der Methode, die zur Ausführung der Operation verwendet wird.

Identity-based Richtlinien

Identity-based Richtlinien sind Richtliniendokumente für JSON-Berechtigungen, die Sie an eine Identität (Benutzer, Gruppe oder Rolle) anhängen. Diese Richtlinien steuern, welche Aktionen Identitäten für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter [Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien](#) im IAM-Benutzerhandbuch.

Identity-based Richtlinien können Inline-Richtlinien (direkt in eine einzelne Identität eingebettet) oder verwaltete Richtlinien (eigenständige Richtlinien, die mehreren Identitäten zugeordnet sind) sein. Informationen dazu, wie Sie zwischen verwalteten und Inline-Richtlinien wählen, finden Sie unter [Choose between managed policies and inline policies](#) im IAM-Benutzerhandbuch.

Resource-based Richtlinien

Resource-based Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anhängen. Beispiele hierfür sind Vertrauensrichtlinien für IAM-Rollen und Amazon S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Sie müssen in einer ressourcenbasierten Richtlinie [einen Prinzipal angeben](#).

Resource-based Richtlinien sind Inline-Richtlinien, die sich in diesem Dienst befinden. Sie können AWS verwaltete Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

Weitere Richtlinientypen

AWS unterstützt zusätzliche Richtlinientypen, mit denen die maximalen Berechtigungen festgelegt werden können, die durch gängigere Richtlinientypen gewährt werden:

- **Berechtigungsgrenzen** – Eine Berechtigungsgrenze legt die maximalen Berechtigungen fest, die eine identitätsbasierte Richtlinie einer IAM-Entität erteilen kann. Weitere Informationen finden Sie unter [Berechtigungsgrenzen für IAM-Entitäten](#) im -IAM-Benutzerhandbuch.
- **Service-Kontrollrichtlinien (SCPs)** – SCPs legen die maximalen Berechtigungen für eine Organisation oder Organisationseinheit in AWS Organizations fest. Weitere Informationen finden Sie unter [Service-Kontrollrichtlinien](#) im AWS Organizations -Benutzerhandbuch.
- **Ressourcen-Kontrollrichtlinien (RCPs)** – RCPs definieren die maximale Anzahl an Berechtigungen, die Ressourcen in Ihren Konten zur Verfügung stehen. Weitere Informationen finden Sie unter [Ressourcen-Kontrollrichtlinien](#) im AWS Organizations -Benutzerhandbuch.
- **Sitzungsrichtlinien** – Sitzungsrichtlinien sind erweiterte Richtlinien, die als Parameter übergeben werden, wenn Sie eine temporäre Sitzung für eine Rolle oder einen Verbundbenutzer erstellen. Weitere Informationen finden Sie unter [Sitzungsrichtlinien](#) im IAM-Benutzerhandbuch.

Mehrere Richtlinientypen

Wenn mehrere Arten von Richtlinien für eine Anfrage gelten, sind die daraus resultierenden Berechtigungen schwieriger zu verstehen. Informationen darüber, wie AWS bestimmt wird, ob eine

Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie unter [Bewertungslogik für Richtlinien](#) im IAM-Benutzerhandbuch.

Wie Amazon Transcribe funktioniert mit IAM

Bevor Sie IAM verwenden, um den Zugriff auf zu verwalten Amazon Transcribe, sollten Sie sich darüber informieren, mit welchen IAM-Funktionen Sie arbeiten können. Amazon Transcribe

IAM Funktionen, die Sie mit verwenden können Amazon Transcribe

IAM Merkmal	Amazon Transcribe Unterstützung
Identity-based Richtlinien	Ja
Resource-based Richtlinien	Nein
Richtlinienaktionen	Ja
Richtlinienressourcen	Ja
Richtlinienbedingungsschlüssel (servicespezifisch)	Ja
ACLs	Nein
ABAC (Tags in Richtlinien)	Teilweise
Temporäre Anmeldeinformationen	Ja
Prinzipalberechtigungen	Ja
Servicerollen	Ja
Service-linked -Rollen	Nein

Einen allgemeinen Überblick darüber, wie Amazon Transcribe und wie andere AWS Dienste mit den meisten IAM Funktionen funktionieren, finden Sie IAM im IAM Benutzerhandbuch unter [AWS Dienste, die mit funktionieren](#).

Identity-based Richtlinien für Amazon Transcribe

Unterstützt Richtlinien auf Identitätsbasis: Ja

Identity-based Richtlinien sind Richtliniendokumente für JSON-Berechtigungen, die Sie an eine Identität anhängen können, z. B. an einen IAM-Benutzer, eine Benutzergruppe oder eine Rolle. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter [Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien](#) im IAM-Benutzerhandbuch.

Mit identitätsbasierten IAM-Richtlinien können Sie angeben, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der [IAM-Referenz für JSON-Richtlinienelemente](#) im IAM-Benutzerhandbuch.

Identity-based Richtlinienbeispiele für Amazon Transcribe

Beispiele für Amazon Transcribe identitätsbasierte Richtlinien finden Sie unter [Amazon Transcribe Beispiele für identitätsbasierte Politik](#)

Resource-based Richtlinien innerhalb Amazon Transcribe

Unterstützt ressourcenbasierte Richtlinien: Nein

Resource-based Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anhängen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie [einen Prinzipal angeben](#). Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Um kontoübergreifenden Zugriff zu ermöglichen, können Sie ein gesamtes Konto oder IAM-Entitäten in einem anderen Konto als Prinzipal in einer ressourcenbasierten Richtlinie angeben. Weitere Informationen finden Sie unter [Kontoübergreifender Ressourcenzugriff in IAM](#) im IAM-Benutzerhandbuch.

Richtlinienaktionen für Amazon Transcribe

Unterstützt Richtlinienaktionen: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Nehmen Sie Aktionen in eine Richtlinie auf, um Berechtigungen zur Ausführung des zugehörigen Vorgangs zu erteilen.

Eine Liste der Amazon Transcribe Aktionen finden Sie unter [Aktionen definiert von Amazon Transcribe](#) in der Serviceautorisierungsreferenz.

Bei Richtlinienaktionen wird das `transcribe` Präfix vor der Aktion Amazon Transcribe verwendet. Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

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

Sie können auch Platzhalter verwenden, um mehrere Aktionen anzugeben. Beispielsweise können Sie alle Aktionen festlegen, die mit dem Wort `List` beginnen, einschließlich der folgenden Aktion:

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

Beispiele für Amazon Transcribe identitätsbasierte Richtlinien finden Sie unter [Amazon Transcribe Beispiele für identitätsbasierte Politik](#)

Politische Ressourcen für Amazon Transcribe

Unterstützt Richtlinienressourcen: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Als Best Practice geben Sie eine Ressource mit dem zugehörigen [Amazon-Ressourcennamen \(ARN\)](#) an. Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

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

Eine Liste der Amazon Transcribe Ressourcentypen und ihrer ARNs finden Sie unter [Ressourcen definiert von Amazon Transcribe](#) in der Service Authorization Reference. Informationen zu den Aktionen, mit denen Sie den ARN einzelner Ressourcen angeben können, finden Sie unter [Von Amazon Transcribe definierte Aktionen](#).

Beispiele für Amazon Transcribe identitätsbasierte Richtlinien finden Sie unter [Amazon Transcribe Beispiele für identitätsbasierte Politik](#)

Bedingungsschlüssel für Richtlinien für Amazon Transcribe

Unterstützt servicespezifische Richtlinienbedingungsschlüssel: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das Element `Condition` gibt an, wann Anweisungen auf der Grundlage definierter Kriterien ausgeführt werden. Sie können bedingte Ausdrücke erstellen, die [Bedingungsoperatoren](#) verwenden, z. B. ist gleich oder kleiner als, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter [Kontextschlüssel für AWS globale Bedingungen](#) im IAM-Benutzerhandbuch.

Eine Liste der Amazon Transcribe Bedingungsschlüssel finden Sie unter [Bedingungsschlüssel für Amazon Transcribe](#) in der Service Authorization Reference. Informationen zu den Aktionen und Ressourcen, mit denen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter [Aktionen definiert von Amazon Transcribe](#).

Beispiele für Amazon Transcribe identitätsbasierte Richtlinien finden Sie unter [Amazon Transcribe Beispiele für identitätsbasierte Politik](#)

ACLs in Amazon Transcribe

Unterstützt ACLs: Nein

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

ABAC mit Amazon Transcribe

Unterstützt ABAC (Tags in Richtlinien): Teilweise

Attribute-based Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, die Berechtigungen auf der Grundlage von Attributen definiert, die als Tags bezeichnet werden. Sie können Tags an IAM-Entitäten und AWS -Ressourcen anhängen und dann ABAC-Richtlinien entwerfen, die Operationen zulassen, wenn das Tag des Prinzipals mit dem Tag auf der Ressource übereinstimmt.

Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungelement einer [Richtlinie Tag-Informationen](#) an, indem Sie die Schlüssel `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, oder Bedingung `aws:TagKeys` verwenden.

Wenn ein Service alle drei Bedingungsschlüssel für jeden Ressourcentyp unterstützt, lautet der Wert für den Service Ja. Wenn ein Service alle drei Bedingungsschlüssel für nur einige Ressourcentypen unterstützt, lautet der Wert Teilweise.

Weitere Informationen zu ABAC finden Sie unter [Definieren von Berechtigungen mit ABAC-Autorisierung](#) im IAM-Benutzerhandbuch. Um ein Tutorial mit Schritten zur Einstellung von ABAC anzuzeigen, siehe [Attributbasierte Zugriffskontrolle \(ABAC\)](#) verwenden im IAM-Benutzerhandbuch.

Weitere Informationen zum Markieren von Amazon Transcribe Ressourcen finden Sie unter [Taggen von Ressourcen](#). Ausführlichere Informationen zur Tag-basierten Zugriffskontrolle finden Sie unter [Steuerung des Zugriffs auf AWS -Ressourcen mithilfe von Tags](#).

Verwenden temporärer Anmeldeinformationen mit Amazon Transcribe

Unterstützt temporäre Anmeldeinformationen: Ja

Temporäre Anmeldeinformationen ermöglichen kurzfristigen Zugriff auf AWS Ressourcen und werden automatisch erstellt, wenn Sie einen Verbund verwenden oder die Rollen wechseln. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter [Temporäre Anmeldeinformationen in IAM und AWS-Services , die mit IAM funktionieren](#) im IAM-Benutzerhandbuch.

Cross-service Hauptberechtigungen für Amazon Transcribe

Unterstützt Forward Access Sessions (FAS): Ja

Forward Access Sessions (FAS) verwenden die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, in Kombination mit der Anfrage, Anfragen AWS-Service an nachgelagerte Dienste zu stellen. Einzelheiten zu den Richtlinien für FAS-Anforderungen finden Sie unter [Zugriffssitzungen weiterleiten](#).

Servicerollen für Amazon Transcribe

Unterstützt Servicerollen: Ja

Eine Servicerolle ist eine [IAM-Rolle](#), die ein Service annimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service](#) im IAM-Benutzerhandbuch.

Warning

Durch das Ändern der Berechtigungen für eine Servicerolle kann die Amazon Transcribe Funktionalität beeinträchtigt werden. Bearbeiten Sie Servicerollen nur, Amazon Transcribe wenn Sie dazu eine Anleitung erhalten.

Service-linked Rollen für Amazon Transcribe

Unterstützt serviceverknüpfte Rollen: Ja

Eine dienstbezogene Rolle ist eine Art von Servicerolle, die mit einer AWS-Service verknüpft ist. Der Dienst kann die Rolle übernehmen, eine Aktion in Ihrem Namen auszuführen. Service-linked Rollen erscheinen in Ihrem Dienst AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.

Amazon Transcribe unterstützt keine dienstbezogenen Rollen.

Einzelheiten zum Erstellen oder Verwalten von dienstbezogenen Rollen für andere Dienste finden Sie unter [AWS Dienste, die mit funktionieren](#). IAM Suchen Sie in der Tabelle nach einem Dienst, der einen Yes in der Service-linked Rollenspalte enthält. Wählen Sie den Link Yes (Ja) aus, um die Dokumentation für die serviceverknüpfte Rolle für diesen Service anzuzeigen.

Cross-service Prävention verwirrter Abgeordneter

Ein confused Deputy ist eine Entität (ein Service oder ein Konto), die von einer anderen Entität gezwungen wird, eine Aktion durchzuführen. Diese Art des Identitätswechsels kann konto- und dienstübergreifend erfolgen.

Um zu verhindern, dass Stellvertreter verwirrt werden, AWS stellt Tools bereit, mit denen Sie Ihre Daten für alle Dienste schützen können, indem Sie Dienstprinzipale verwenden, denen Zugriff auf Ressourcen in Ihrem gewährt wurde. AWS-Kontoin diesem Abschnitt geht es speziell um die Vermeidung von Problemen mit verwirrten Amazon Transcribe Stellvertretern zwischen den einzelnen Diensten. Weitere Informationen zu diesem Thema finden Sie jedoch im Abschnitt „[Problem mit verwirrten Stellvertretern](#)“ des Benutzerhandbuchs.IAM

Um die Zugriffsrechte IAM auf Ihre Ressourcen einzuschränken, empfehlen wir, die Kontextschlüssel für globale Bedingungen [aws:SourceArn](#) und [aws:SourceAccount](#) in Ihren Ressourcenrichtlinien zu verwenden. Amazon Transcribe

Wenn Sie diese beiden globalen Bedingungskontextschlüssel verwenden und der `aws:SourceArn` Wert die AWS-Konto ID enthält, `aws:SourceArn` müssen der `aws:SourceAccount` Wert und die AWS-Konto Eingabe dieselbe AWS-Konto ID verwenden, wenn sie in derselben Richtlinienanweisung verwendet werden.

Wenn Sie nur eine Ressource mit dem betriebsübergreifenden Zugriff verknüpfen möchten, verwenden Sie `aws:SourceArn`. Wenn Sie eine der Ressourcen in diesem Bereich AWS-Konto mit dienstübergreifendem Zugriff verknüpfen möchten, verwenden Sie `aws:SourceAccount`.

Note

Der wirksamste Schutz gegen Confused-Deputy-Probleme ist die Verwendung des globalen Bedingungskontextschlüssels `aws:SourceArn` mit dem vollständigen ARN der Ressource. Wenn Sie den vollständigen ARN nicht kennen oder wenn Sie mehrere Ressourcen angeben, verwenden Sie den globalen Kontextbedingungsschlüssel `aws:SourceArn` mit Platzhaltern (*) für die unbekannten Teile des ARN. Beispiel, `arn:aws:transcribe::123456789012:*`.

Ein Beispiel für eine Rollenrichtlinie, die zeigt, wie Sie ein Problem mit „Confused Deputy“ verhindern können, finden Sie unter [Confused-Deputy-Prävention-Richtlinie](#).

Amazon Transcribe Beispiele für identitätsbasierte Politik

Benutzer und Rollen haben standardmäßig nicht die Berechtigung, Amazon Transcribe -Ressourcen zu erstellen oder zu ändern. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter [Erstellen von IAM-Richtlinien \(Konsole\)](#) im IAM-Benutzerhandbuch.

Details zu den von Amazon Transcribe definierten Aktionen und Ressourcentypen, einschließlich des Formats der ARNs für jeden der Ressourcentypen, finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für Amazon Transcribe](#) in der Service-Autorisierungs-Referenz.

Themen

- [Best Practices für Richtlinien](#)
- [Verwendung der AWS-Managementkonsole](#)
- [Berechtigungen erforderlich für IAM Rollen](#)
- [Berechtigungen erforderlich für Amazon S3 Verschlüsselungsschlüssel](#)
- [Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer](#)
- [AWS KMS Richtlinie für den Verschlüsselungskontext](#)
- [Confused-Deputy-Prävention-Richtlinie](#)
- [Anzeigen von Transkriptionsaufträgen basierend auf Tags](#)

Best Practices für Richtlinien

Identity-based Richtlinien legen fest, ob jemand Amazon Transcribe Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder sie löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Beachten Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Richtlinien und Empfehlungen:

- Erste Schritte mit AWS verwalteten Richtlinien und Umstellung auf Berechtigungen mit den geringsten Rechten — Verwenden Sie die AWS verwalteten Richtlinien, die Berechtigungen für viele gängige Anwendungsfälle gewähren, um damit zu beginnen, Ihren Benutzern und Workloads Berechtigungen zu gewähren. Sie sind in Ihrem verfügbar. AWS-Konto Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom AWS Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen

finden Sie unter [Von AWS verwaltete Richtlinien](#) oder [Von AWS verwaltete Richtlinien für Auftragsfunktionen](#) im IAM-Benutzerhandbuch.

- Anwendung von Berechtigungen mit den geringsten Rechten – Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung von IAM zum Anwenden von Berechtigungen finden Sie unter [Richtlinien und Berechtigungen in IAM](#) im IAM-Benutzerhandbuch.
- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs – Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Serviceaktionen zu gewähren, wenn diese für einen bestimmten Zweck verwendet werden AWS-Service, z. CloudFormation B. Weitere Informationen finden Sie unter [IAM-JSON-Richtlinienelemente: Bedingung](#) im IAM-Benutzerhandbuch.
- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten – IAM Access Analyzer validiert neue und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter [Richtlinienvvalidierung mit IAM Access Analyzer](#) im IAM-Benutzerhandbuch.
- Multi-Faktor-Authentifizierung (MFA) erforderlich — Wenn Sie ein Szenario haben, das IAM-Benutzer oder einen Root-Benutzer in Ihrem System erfordert AWS-Konto, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter [Sicherer API-Zugriff mit MFA](#) im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter [Best Practices für die Sicherheit in IAM](#) im IAM-Benutzerhandbuch.

Verwendung der AWS-Managementkonsole

Um auf die Amazon Transcribe-Konsole zugreifen zu können, müssen Sie über ein Mindestmaß an Berechtigungen verfügen. Diese Berechtigungen müssen Ihnen das Auflisten und Anzeigen

von Details zu den Amazon Transcribe Ressourcen in Ihrem ermöglichen. AWS-Konto Wenn Sie eine identitätsbasierte Richtlinie erstellen, die strenger ist als die mindestens erforderlichen Berechtigungen, funktioniert die Konsole nicht wie vorgesehen für Entitäten (Benutzer oder Rollen) mit dieser Richtlinie.

Sie müssen Benutzern, die nur die API AWS CLI oder die AWS API aufrufen, keine Mindestberechtigungen für die Konsole gewähren. Stattdessen sollten Sie nur Zugriff auf die Aktionen zulassen, die der API-Operation entsprechen, die die Benutzer ausführen möchten.

Um sicherzustellen, dass eine Entität (Benutzer und Rollen) die verwenden kann [AWS-Managementkonsole](#), fügen Sie ihnen eine der folgenden AWS verwalteten Richtlinien bei.

- `AmazonTranscribeFullAccess`: Gewährt vollen Zugriff zum Erstellen, Lesen, Aktualisieren, Löschen und Ausführen aller Amazon Transcribe Ressourcen. Es ermöglicht auch den Zugriff auf Amazon S3 -Buckets mit `transcribe` im Bucket-Namen.
- `AmazonTranscribeReadOnlyAccess`: Gewährt Lesezugriff auf die Ressourcen von Amazon Transcribe, sodass Sie Transkriptionsaufträge und benutzerdefinierte Vokabulare abrufen und auflisten können.

Note

Sie können die verwalteten Berechtigungsrichtlinien überprüfen, indem Sie sich bei IAM AWS-Managementkonsole anmelden und nach dem Namen der Richtlinie suchen. Bei der Suche nach „transcribe“ werden beide oben aufgeführten Richtlinien (`AmazonTranscribeReadOnly` und `AmazonTranscribeFullAccess`) zurückgegeben.

Sie können auch Ihre eigenen benutzerdefinierten IAM Richtlinien erstellen, um Berechtigungen für Amazon Transcribe API-Aktionen zuzulassen. Sie können diese benutzerdefinierten Richtlinien mit den Entitäten verknüpfen, die diese Berechtigungen benötigen.

Berechtigungen erforderlich für IAM Rollen

Wenn Sie eine IAM Rolle zum Aufrufen erstellen Amazon Transcribe, muss diese über die Zugriffsberechtigung für den Amazon S3 Bucket verfügen. Falls zutreffend, muss KMS key auch zur Verschlüsselung des Inhalts des Buckets verwendet werden. In den folgenden Abschnitten finden Sie Beispiele für Richtlinien.

Vertrauensrichtlinien

Die IAM Entität, die Sie für Ihre Transkriptionsanfrage verwenden, muss über eine Vertrauensrichtlinie verfügen, die es ermöglicht, diese Rolle Amazon Transcribe zu übernehmen. Verwenden Sie die folgende Amazon Transcribe Vertrauensrichtlinie. Beachten Sie, dass Sie bei einer Echtzeit-Call-Analytics-Anforderung mit aktivierter Analyse nach Anrufen die „Vertrauensrichtlinie für Echtzeit-Call-Analytics“ verwenden müssen.

Vertrauensrichtlinie für 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:*"
        }
      }
    }
  ]
}
```

Vertrauensrichtlinie für Echtzeit-Call-Analytics

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 Richtlinie für den Eingabe-Bucket

Die folgende Richtlinie erteilt einer IAM Rolle die Berechtigung, auf Dateien aus dem angegebenen Eingabe-Bucket zuzugreifen.

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 Richtlinie für den Ausgabe-Bucket

Die folgende Richtlinie erteilt einer IAM Rolle die Berechtigung, Dateien in den angegebenen Ausgabe-Bucket zu schreiben.

JSON

```

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

```

Berechtigungen erforderlich für Amazon S3 Verschlüsselungsschlüssel

Wenn Sie einen KMS key Amazon S3 Bucket verschlüsseln, schließen Sie Folgendes in die KMS key Richtlinie ein. Dadurch erhalten Amazon Transcribe Sie Zugriff auf den Inhalt des Buckets. Weitere Informationen zum Zulassen des Zugriffs auf KMS keys finden Sie unter [Zulassen des Zugriffs auf AWS-Konten an durch externe KMS key](#) Benutzer im AWS KMS Entwicklerhandbuch.

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"
  }
]
}

```

Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer Benutzeridentität angefügt sind. Diese Richtlinie umfasst Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe der AWS CLI AWS OR-API.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ],
      "Resource": ["arn:aws:iam::*:user/${aws:username}"]
    },
    {
      "Sid": "NavigateInConsole",
      "Effect": "Allow",
      "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",

```

```

        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
    ],
    "Resource": "*"
}
]
}

```

AWS KMS Richtlinie für den Verschlüsselungskontext

Die folgende Richtlinie gewährt der IAM Rolle „ExampleRole“ die Berechtigung, die Operationen AWS KMS Entschlüsseln und Verschlüsseln für diesen speziellen Zweck zu verwenden. KMS key Diese Richtlinie gilt nur für Anfragen mit mindestens einem Verschlüsselungskontextpaar, in diesem Fall „color:indigoBlue“. Weitere Informationen zum AWS KMS Verschlüsselungskontext finden Sie unter. [AWS KMS Verschlüsselungskontext](#)

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": "*"
    }
  ]
}

```

Confused-Deputy-Prävention-Richtlinie

Im Folgenden finden Sie ein Beispiel für eine Richtlinie „Rolle übernehmen“. Es zeigt, wie Sie mithilfe von `aws:SourceArn` und `aws:SourceAccount` mit verhindern können, dass ein Amazon Transcribe unübersichtliches Stellvertreterproblem entsteht. Weitere Informationen zur Confused-Deputy-Prävention finden Sie unter [Cross-service Prävention verwirrter Abgeordneter](#).

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:*"
        }
      }
    }
  ]
}

```

```

    }
  ]
}

```

Anzeigen von Transkriptionsaufträgen basierend auf Tags

Sie können in Ihrer identitätsbasierten Richtlinie Bedingungen für die Steuerung des Zugriffs auf Amazon Transcribe -Ressourcen auf der Basis von Tags verwenden. In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die die Anzeige eines Transkriptionsauftrags ermöglicht. Die Berechtigung wird jedoch nur erteilt, wenn das `Owner`-Transkriptionsauftrags-Tag den Wert des Benutzernamens dieses Benutzers hat. Diese Richtlinie gewährt auch die Berechtigungen, die für die Ausführung dieser Aktion über die AWS-Managementkonsole erforderlich sind.

Sie können diese Richtlinie den IAM Entitäten in Ihrem Konto zuordnen. Wenn eine Rolle namens `test-role` versucht, einen Transkriptionsauftrag anzuzeigen, muss der Transkriptionsauftrag markiert werden als `Owner=test-role` oder `owner=test-role` (Bedingungsschlüsselnamen werden nicht zwischen Groß- und Kleinschreibung unterschieden), andernfalls wird ihnen der Zugriff verweigert. Weitere Informationen finden Sie unter [IAM -JSON-Richtlinienelemente: Bedingung](#) im IAM -Benutzerhandbuch.

Weitere Informationen zum Taggen finden Sie Amazon Transcribe unter [Taggen von Ressourcen](#).

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}"}  
  }  
}  
]  
}
```

Fehlerbehebung Amazon Transcribe Identität und Zugriff

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit Amazon Transcribe und AWS Identity and Access Management (IAM) auftreten können.

Themen

- [Ich bin nicht berechtigt, eine Aktion durchzuführen in Amazon Transcribe](#)
- [Ich bin nicht berechtigt, iam auszuführen: PassRole](#)
- [Ich möchte Personen außerhalb meiner Umgebung zulassen AWS-Konto um auf meine zuzugreifen Amazon Transcribe Ressourcen](#)

Ich bin nicht berechtigt, eine Aktion durchzuführen in Amazon Transcribe

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht zur Durchführung einer Aktion berechtigt sind, müssen Ihre Richtlinien aktualisiert werden, damit Sie die Aktion durchführen können.

Der folgende Beispielfehler tritt auf, wenn der IAM-Benutzer mateojackson versucht, über die Konsole Details zu einer fiktiven *my-example-widget*-Ressource anzuzeigen, jedoch nicht über `transcribe:GetWidget`-Berechtigungen verfügt.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:  
transcribe:GetWidget on resource: my-example-widget
```

In diesem Fall muss die Richtlinie für den Benutzer mateojackson aktualisiert werden, damit er mit der `transcribe:GetWidget`-Aktion auf die *my-example-widget*-Ressource zugreifen kann.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich bin nicht berechtigt, iam auszuführen: PassRole

Wenn Sie die Fehlermeldung erhalten, dass Sie nicht zum Durchführen der `iam:PassRole`-Aktion autorisiert sind, müssen Ihre Richtlinien aktualisiert werden, um eine Rolle an Amazon Transcribe übergeben zu können.

Einige AWS-Services ermöglichen es Ihnen, eine bestehende Rolle an diesen Dienst zu übergeben, anstatt eine neue Servicerolle oder eine dienstverknüpfte Rolle zu erstellen. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Dienst.

Der folgende Beispielfehler tritt auf, wenn ein IAM-Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion in Amazon Transcribe auszuführen. Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Servicerolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Dienst.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

In diesem Fall müssen die Richtlinien von Mary aktualisiert werden, um die Aktion `iam:PassRole` ausführen zu können.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich möchte Personen außerhalb meiner Umgebung zulassen AWS-Konto um auf meine zuzugreifen Amazon Transcribe Ressourcen

Sie können eine Rolle erstellen, mit der Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation auf Ihre Ressourcen zugreifen können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Im Fall von Diensten, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (Access Control Lists, ACLs) verwenden, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen darüber, ob diese Funktionen Amazon Transcribe unterstützt werden, finden Sie unter [Wie Amazon Transcribe funktioniert mit IAM](#).
- Informationen dazu, wie Sie Zugriff auf Ihre Ressourcen gewähren können, AWS-Konten die Ihnen gehören, finden Sie im IAM-Benutzerhandbuch unter [Gewähren des Zugriffs auf einen IAM-Benutzer in einem anderen AWS-Konto, den Sie besitzen](#).

- Informationen dazu, wie Sie Dritten Zugriff auf Ihre Ressourcen gewähren können AWS-Konten, finden Sie [AWS-Konten im IAM-Benutzerhandbuch unter Gewähren des Zugriffs für Dritte](#).
- Informationen dazu, wie Sie über einen Identitätsverbund Zugriff gewähren, finden Sie unter [Gewähren von Zugriff für extern authentifizierte Benutzer \(Identitätsverbund\)](#) im IAM-Benutzerhandbuch.
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter [Kontoübergreifender Ressourcenzugriff in IAM](#) im IAM-Benutzerhandbuch.

Datenschutz in Amazon Transcribe

Das [Modell der AWS gemeinsamen Verantwortung](#) gilt für den Datenschutz in Amazon Transcribe. Wie in diesem Modell beschrieben, AWS ist verantwortlich für den Schutz der globalen Infrastruktur, auf der alle Systeme laufen AWS Cloud. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services verantwortlich. Weitere Informationen zum Datenschutz finden Sie unter [Häufig gestellte Fragen zum Datenschutz](#) . Informationen zum Datenschutz in Europa finden Sie im [General Data Protection Regulation \(GDPR\) Center](#).

Aus Datenschutzgründen empfehlen wir, dass Sie Ihre AWS-Konto Anmeldeinformationen schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einrichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor-Authentifizierung (MFA).
- Wird verwendet SSL/TLS , um mit AWS Ressourcen zu kommunizieren. Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein AWS CloudTrail. Informationen zur Verwendung von CloudTrail Pfaden zur Erfassung von AWS Aktivitäten finden Sie unter [Arbeiten mit CloudTrail Pfaden](#) im AWS CloudTrail Benutzerhandbuch.
- Verwenden Sie AWS Verschlüsselungslösungen zusammen mit allen darin enthaltenen Standardsicherheitskontrollen AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.

- Wenn Sie für den Zugriff AWS über eine Befehlszeilenschnittstelle oder eine API FIPS 140-3-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter [Federal Information Processing Standard \(FIPS\) 140-3](#).

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie mit der Konsole, der API Amazon Transcribe oder den SDKs arbeiten oder diese anderweitig AWS-Services verwenden. AWS CLI AWS Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn Sie eine URL für einen externen Server bereitstellen, empfehlen wir dringend, keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

Inter-network Datenschutz im Verkehr

Ein Amazon Virtual Private Cloud (Amazon VPC) -Endpunkt für Amazon Transcribe ist eine logische Entität innerhalb einer VPC, die Konnektivität nur für ermöglicht. Amazon Transcribe Amazon VPC leitet Anfragen an die VPC weiter Amazon Transcribe und leitet Antworten zurück an diese weiter. Weitere Informationen finden Sie unter [AWS PrivateLink -Konzepte](#). Informationen zur Verwendung von Amazon VPC Endpunkten finden Sie unter Amazon Transcribe . [Amazon Transcribe und Schnittstellen-VPC-Endpunkte \(AWS PrivateLink\)](#)

Datenverschlüsselung

Die Datenverschlüsselung bezieht sich auf den Schutz von Daten während der Übertragung und im Ruhezustand. Sie können Ihre Daten schützen, indem Sie Amazon S3 verwaltete Schlüssel oder Schlüssel KMS keys im Ruhezustand zusammen mit standardmäßigem Transport Layer Security (TLS) während der Übertragung verwenden.

Verschlüsselung im Ruhezustand

Amazon Transcribe verwendet den Amazon S3 Standardschlüssel (SSE-S3) für die serverseitige Verschlüsselung der in Ihrem Bucket gespeicherten Transkripte. Amazon S3

Wenn Sie den [StartTranscriptionJob](#)Vorgang verwenden, können Sie einen eigenen angeben, KMS key um die Ausgabe eines Transkriptionsauftrags zu verschlüsseln.

Amazon Transcribe verwendet ein Amazon EBS Volume, das mit dem Standardschlüssel verschlüsselt ist.

Verschlüsselung während der Übertragung

Amazon Transcribe verwendet TLS 1.2 mit AWS Zertifikaten, um Daten bei der Übertragung zu verschlüsseln. Dazu gehören auch Streaming-Transkriptionen.

Schlüsselverwaltung

Amazon Transcribe arbeitet mit KMS keys , um eine verbesserte Verschlüsselung für Ihre Daten bereitzustellen. Mit können Sie Ihre Eingabemedien verschlüsseln Amazon S3, wenn Sie einen Transkriptionsauftrag erstellen. Die Integration mit AWS KMS ermöglicht die Verschlüsselung der Ausgabe einer [StartTranscriptionJob](#)Anfrage.

Wenn Sie keine angeben KMS key, wird die Ausgabe des Transkriptionsauftrags mit dem Amazon S3 Standardschlüssel (SSE-S3) verschlüsselt.

Weitere Informationen dazu finden Sie im AWS KMS [AWS Key Management Service Entwicklerhandbuch](#).

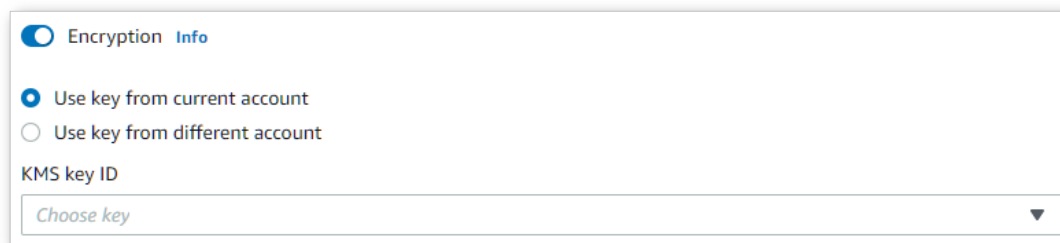
Schlüsselverwaltung mit dem AWS-Managementkonsole

Um die Ausgabe Ihres Transkriptionsauftrags zu verschlüsseln, können Sie wählen, ob Sie ein KMS key für den AWS-Konto , der die Anfrage stellt, oder ein KMS key von einem anderen verwenden möchten. AWS-Konto

Wenn Sie keine angeben KMS key, wird die Ausgabe des Transkriptionsauftrags mit dem Amazon S3 Standardschlüssel () verschlüsselt. SSE-S3

So aktivieren Sie die Verschlüsselung der Ausgabe:

1. Wählen Sie unter Output data (Ausgabedaten) die Option Encryption (Verschlüsselung).



Encryption [Info](#)

☒ Use key from current account

☐ Use key from different account

KMS key ID

Choose key ▼

2. Wählen Sie aus, ob das von dem KMS key stammt, das AWS-Konto Sie gerade verwenden, oder von einem anderen. AWS-Konto Wenn Sie einen Schlüssel aus der aktuellen Version verwenden

möchten AWS-Konto, wählen Sie den Schlüssel aus der KMS key ID aus. Wenn Sie einen Schlüssel von einem anderen verwenden AWS-Konto, müssen Sie den ARN des Schlüssels eingeben. Um einen Schlüssel von einem anderen zu verwenden AWS-Konto, muss der Anrufer über `kms:Encrypt` Berechtigungen für den KMS key verfügen. Weitere Informationen finden Sie unter [Erstellen einer Schlüsselrichtlinie](#).

Schlüsselverwaltung über die API

Um die Ausgabeverschlüsselung mit der API zu verwenden, müssen Sie angeben, dass Sie den `OutputEncryptionKMSKeyId` Parameter der [StartTranscriptionJob](#) Operation [StartCallAnalyticsJob](#) [StartMedicalTranscriptionJob](#), oder KMS key verwenden.

Wenn Sie einen Schlüssel verwenden, der sich in der aktuellen Version befindet AWS-Konto, können Sie Ihren KMS key auf eine von vier Arten angeben:

1. Verwenden Sie die KMS key ID selbst. Beispiel, `1234abcd-12ab-34cd-56ef-1234567890ab`.
2. Verwenden Sie einen Alias für die KMS key ID. Beispiel, `alias/ExampleAlias`.
3. Verwenden Sie den Amazon-Ressourcennamen (ARN) für die KMS key ID. Beispiel, `arn:aws:kms:region:account-ID:key/1234abcd-12ab-34cd-56ef-1234567890ab`.
4. Verwenden Sie den ARN für den KMS key Alias. Beispiel, `arn:aws:kms:region:account-ID:alias/ExampleAlias`.

Wenn Sie einen Schlüssel verwenden, der sich in einem anderen AWS-Konto als dem aktuellen befindet AWS-Konto, können Sie Ihren KMS key Schlüssel auf zwei Arten angeben:

1. Verwenden Sie den ARN für die KMS key ID. Beispiel, `arn:aws:kms:region:account-ID:key/1234abcd-12ab-34cd-56ef-1234567890ab`.
2. Verwenden Sie den ARN für den KMS key Alias. Beispiel, `arn:aws:kms:region:account-ID:alias/ExampleAlias`.

Beachten Sie, dass die anfragende Entität die Erlaubnis haben muss, die angegebene KMS key zu verwenden.

AWS KMS Verschlüsselungskontext

AWS KMS Der Verschlüsselungskontext ist eine Zuordnung von nicht geheimen Schlüssel/Wert-Paaren im Klartext. Diese Zuordnung stellt zusätzliche authentifizierte Daten dar, die als

Verschlüsselungskontextpaare bezeichnet werden und eine zusätzliche Sicherheitsebene für Ihre Daten bieten. Amazon Transcribe erfordert einen symmetrischen Verschlüsselungsschlüssel, um die Transkriptionsausgabe in einen vom Kunden angegebenen Bucket zu verschlüsseln. Amazon S3 Weitere Informationen finden Sie unter [Asymmetrische Schlüssel in AWS KMS](#).

Wenn Sie Ihre Verschlüsselungskontextpaare erstellen, sollten Sie keine sensiblen Informationen einschließen. Der Verschlüsselungskontext ist nicht geheim — er ist im Klartext in Ihren CloudTrail Protokollen sichtbar (sodass Sie ihn verwenden können, um Ihre kryptografischen Operationen zu identifizieren und zu kategorisieren).

Ihr Verschlüsselungskontextpaar kann Sonderzeichen enthalten, z. B. Unterstriche (_), Bindestriche (-), Schrägstriche (/ , \) und Doppelpunkte (:).

Tip

Es kann nützlich sein, die Werte in Ihrem Verschlüsselungskontextpaar mit den zu verschlüsselnden Daten in Beziehung zu setzen. Obwohl dies nicht erforderlich ist, empfehlen wir Ihnen, nicht sensible Metadaten zu verwenden, die sich auf Ihre verschlüsselten Inhalte beziehen, wie z. B. Dateinamen, Header-Werte oder unverschlüsselte Datenbankfelder.

Um die Ausgabeverschlüsselung mit der API zu verwenden, setzen Sie den `KMSEncryptionContext`-Parameter im Vorgang [StartTranscriptionJob](#). Um einen Verschlüsselungskontext für die Ausgabeverschlüsselung bereitzustellen, muss der Parameter `OutputEncryptionKMSKeyId` auf eine symmetrische KMS key -ID verweisen.

[Sie können AWS KMS Bedingungsschlüssel mit IAM Richtlinien verwenden, um den Zugriff auf eine symmetrische Verschlüsselung auf der KMS key Grundlage des Verschlüsselungskontextes zu steuern, der in der Anforderung für einen kryptografischen Vorgang verwendet wurde.](#) Ein Beispiel für eine Verschlüsselungskontextrichtlinie finden Sie unter [AWS KMS Richtlinie für den Verschlüsselungskontext](#).

Die Verwendung des Verschlüsselungskontexts ist optional, wird aber empfohlen. Weitere Informationen finden Sie unter [Verschlüsselungskontext](#).

Abmeldung von der Verwendung Ihrer Daten zur Serviceverbesserung

Amazon Transcribe speichert und verwendet standardmäßig verarbeitete Spracheingaben, um den Service weiterzuentwickeln und Ihr Nutzererlebnis kontinuierlich zu verbessern. Sie können

die Verwendung Ihrer Inhalte zur Entwicklung und Verbesserung deaktivieren, Amazon Transcribe indem Sie eine AWS Organizations Opt-Out-Richtlinie verwenden. Informationen darüber, wie Sie sich abmelden können, finden Sie in den [Richtlinien für das Abmelden von KI-Services](#).

Überwachen Amazon Transcribe

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit und Leistung Ihrer Amazon Transcribe anderen AWS Lösungen. AWS bietet die folgenden Überwachungstools, mit denen Sie beobachten Amazon Transcribe, melden können, wenn etwas nicht stimmt, und gegebenenfalls automatische Maßnahmen ergreifen können:

- Amazon CloudWatch überwacht Ihre AWS Ressourcen und die Anwendungen, auf denen Sie laufen, AWS in Echtzeit. Sie können Kennzahlen erfassen und verfolgen, benutzerdefinierte Dashboards erstellen und Alarme festlegen, die Sie benachrichtigen oder Maßnahmen ergreifen, wenn eine bestimmte Metrik einen von Ihnen festgelegten Schwellenwert erreicht. Sie können beispielsweise die CPU-Auslastung oder andere Messwerte für Ihre Amazon EC2 Instances CloudWatch verfolgen und bei Bedarf automatisch neue Instances starten.
- Amazon CloudWatch Logs kann Ihre Protokolldateien aus Amazon EC2 Instances und anderen Quellen überwachen CloudTrail, speichern und darauf zugreifen. CloudWatch Logs kann Informationen in den Protokolldateien überwachen und Sie benachrichtigen, wenn bestimmte Schwellenwerte erreicht werden. Sie können Ihre Protokolldaten auch in einem sehr robusten Speicher archivieren.
- AWS CloudTrail erfasst API-Aufrufe und zugehörige Ereignisse, die von Ihnen oder in Ihrem Namen getätigt wurden, AWS-Konto und übermittelt die Protokolldateien an einen von Ihnen angegebenen Amazon S3 Bucket. Sie können feststellen, welche Benutzer und Konten angerufen wurden AWS, von welcher Quell-IP-Adresse aus die Anrufe getätigt wurden und wann die Aufrufe erfolgten.

Weitere Informationen finden Sie im [Amazon CloudWatch -Benutzerhandbuch](#).

Amazon EventBridge ist ein serverloser Dienst, der Ereignisse verwendet, um Anwendungskomponenten miteinander zu verbinden, sodass Sie leichter skalierbare, ereignisgesteuerte Anwendungen erstellen können. EventBridge liefert einen Stream von Echtzeitdaten aus Ihren eigenen Anwendungen, SaaS-Anwendungen (Software as a Service) und AWS Diensten und leitet diese Daten an Ziele weiter, wie Lambda z. Sie können Ereignisse überwachen, die in Services auftreten, und ereignisgesteuerte Architekturen aufbauen. Weitere Informationen finden Sie im [Amazon EventBridge -Benutzerhandbuch](#).

Themen

- [Überwachen Amazon Transcribe mit Amazon CloudWatch](#)
- [Überwachen Amazon Transcribe mit AWS CloudTrail](#)
- [Verwenden Amazon EventBridge mit Amazon Transcribe](#)

Überwachen Amazon Transcribe mit Amazon CloudWatch

Sie können die Amazon Transcribe Nutzung überwachen CloudWatch, wobei Rohdaten gesammelt und zu lesbaren Kennzahlen verarbeitet werden, die nahezu in Echtzeit verfügbar sind. Diese Statistiken werden 15 Monate gespeichert, damit Sie auf Verlaufsinformationen zugreifen können und einen besseren Überblick darüber erhalten, wie Ihre Webanwendung oder der Service ausgeführt werden. Sie können auch Alarme einrichten, die auf bestimmte Grenzwerte achten und Benachrichtigungen senden oder Aktivitäten auslösen, wenn diese Grenzwerte erreicht werden. Weitere Informationen finden Sie im [CloudWatch -Benutzerhandbuch](#).

Verwenden Amazon CloudWatch Metriken und Dimensionen mit Amazon Transcribe

Amazon Transcribe unterstützt CloudWatch Metriken und Dimensionen, d. h. Daten, mit denen Sie die Leistung überwachen können. Zu den unterstützten Metrik-Kategorien gehören Datenverkehr, Fehler, Datentransfer und Latenzzeiten im Zusammenhang mit Ihren Transkriptionsaufträgen. Unterstützte Metriken befinden sich CloudWatch im AWS/Transcribe-Namespace.

Note

CloudWatch Monitoring-Metriken sind kostenlos und werden nicht auf die Servicekontingente angerechnet. CloudWatch

Weitere Informationen zu CloudWatch Metriken finden Sie unter [Amazon CloudWatch Metriken verwenden](#).

Überwachen Amazon Transcribe mit AWS CloudTrail

Amazon Transcribe ist in einen Dienst integriert AWS CloudTrail, der eine Aufzeichnung der Aktionen bereitstellt, die Amazon Transcribe von einem AWS Identity and Access Management (IAM) - Benutzer oder einer Rolle oder von einem AWS Dienst ausgeführt wurden. CloudTrail erfasst alle API-Aufrufe für Amazon Transcribe. Dazu gehören Aufrufe von den APIs AWS-Managementkonsole

und Codeaufrufen an die Amazon Transcribe APIs als Ereignisse. Durch die Erstellung eines Trails können Sie die kontinuierliche Bereitstellung von CloudTrail Ereignissen, einschließlich Ereignissen für Amazon Transcribe, an einen Amazon S3 Bucket ermöglichen. Wenn Sie keinen Pfad erstellen, können Sie dennoch die letzten Ereignisse auf CloudTrail AWS-Managementkonsole im Ereignisverlaufeinsehen. Anhand der von gesammelten Informationen können Sie jede Anfrage sehen CloudTrail, an die die Anfrage gestellt wurde Amazon Transcribe, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Details.

Weitere Informationen CloudTrail dazu finden Sie im [AWS CloudTrail Benutzerhandbuch](#).

Amazon Transcribe und CloudTrail

CloudTrail ist auf Ihrem aktiviert AWS-Konto, wenn Sie das Konto erstellen. Wenn eine Aktivität in stattfindet Amazon Transcribe, wird diese Aktivität zusammen mit anderen AWS-Service Ereignissen in der CloudTrail Ereignishistorie in einem CloudTrail Ereignis aufgezeichnet. Sie können aktuelle Ereignisse in Ihrem anzeigen, suchen und herunterladen AWS-Konto. Weitere Informationen finden Sie unter [Anzeigen von Ereignissen mit dem CloudTrail -API-Ereignisverlauf](#).

Um eine fortlaufende Aufzeichnung der Ereignisse in Ihrem Konto zu erhalten AWS-Konto, einschließlich der Ereignisse für Amazon Transcribe, erstellen Sie einen Trail. Ein Trail ist eine Konfiguration, die es CloudTrail ermöglicht, Ereignisse als Protokolldateien an einen bestimmten Amazon S3 Bucket zu übertragen. CloudTrail Protokolldateien enthalten einen oder mehrere Protokolleinträge. Ein Ereignis stellt eine einzelne Anforderung aus einer beliebigen Quelle dar. Sie enthalten Informationen über die angeforderte Aktion, Datum und Uhrzeit der Aktion, Anforderungsparameter usw. CloudTrail Protokolldateien sind kein geordneter Stack-Trace der öffentlichen API-Aufrufe, sodass sie nicht in einer bestimmten Reihenfolge angezeigt werden.

Wenn Sie in der eine Spur erstellen AWS-Managementkonsole, gilt die Spur standardmäßig für alle AWS-Regionen. Der Trail protokolliert alle Ereignisse AWS-Regionen in der AWS Partition und übermittelt die Protokolldateien an den von Ihnen angegebenen Amazon S3 Bucket. Darüber hinaus können Sie andere konfigurieren, AWS-Services um die in den CloudTrail Protokollen gesammelten Ereignisdaten weiter zu analysieren und darauf zu reagieren. Weitere Informationen finden Sie unter:

- [Übersicht zum Erstellen eines Trails](#)
- [CloudTrail Unterstützte Dienste und Integrationen](#)
- [Amazon SNS Benachrichtigungen konfigurieren für CloudTrail](#)
- [Empfangen von CloudTrail -Protokolldateien aus mehreren Regionen](#) und [Empfangen von CloudTrail -Protokolldateien aus mehreren Konten](#)

CloudTrail protokolliert alle Amazon Transcribe Aktionen, die in der [API-Referenz](#) dokumentiert sind. Beispielsweise generieren die [StartTranscriptionJob](#) Operationen [CreateVocabularyGetTranscriptionJob](#), und Einträge in den CloudTrail Protokolldateien. Wenn Amazon Transcribe-API-Operationen CloudTrail protokolliert werden, verwendet der CloudTrail Protokolleintrag leere Zeichenketten für vertrauliche Informationen in den Anfrage- und Antwortparametern, wie z. B. Amazon S3 S3-URI-Werte.

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Mit diesen Informationen können Sie Folgendes bestimmen:

- Ob die Anfrage mit Root- oder IAM Benutzeranmeldedaten gestellt wird
- Ob die Anfrage mit temporären Sicherheitsanmeldeinformationen für eine IAM -Rolle oder einen verbundenen Benutzer ausgeführt wird.
- Ob die Anfrage von einem anderen gestellt wurde AWS-Service

Weitere Informationen finden Sie unter [CloudTrail userIdentity-Element](#).

Sie können auch Amazon Transcribe Protokolldateien aus mehreren AWS-Regionen und mehreren AWS-Konten in einem einzigen Amazon S3 Bucket zusammenfassen. Weitere Informationen finden Sie unter [Empfangen von CloudTrail Protokolldateien aus mehreren Regionen](#) und [Empfangen von CloudTrail Protokolldateien von mehreren Konten](#).

Beispiel: Amazon Transcribe Protokolldateieinträge

Ein Trail ist eine Konfiguration, die die Übertragung von Ereignissen als Protokolldateien an einen bestimmten Amazon S3 Bucket ermöglicht. CloudTrail Protokolldateien enthalten einen oder mehrere Protokolleinträge. Ein Ereignis stellt eine einzelne Anforderung aus einer beliebigen Quelle dar. Es enthält Informationen über die angeforderte Aktion wie Datum und Uhrzeit der Aktion sowie Anforderungsparameter. CloudTrail Protokolldateien sind kein geordneter Stack-Trace der öffentlichen API-Aufrufe, sodass sie nicht in einer bestimmten Reihenfolge angezeigt werden.

Aufrufe an die API-Vorgänge [StartTranscriptionJob](#) und [GetTranscriptionJob](#) erzeugen den folgenden Eintrag.

Note

Wenn Amazon Transcribe-API-Operationen CloudTrail protokolliert werden, verwendet der CloudTrail Protokolleintrag leere Zeichenketten für vertrauliche Informationen in den Anfrage- und Antwortparametern, wie z. B. Amazon S3 S3-URI-Werte.

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

Verwenden Amazon EventBridge mit Amazon Transcribe

Mit Amazon EventBridge können Sie auf Statusänderungen in Ihren Amazon Transcribe Jobs reagieren, indem Sie Ereignisse in anderen Bereichen einleiten. AWS-Services Wenn sich der Status eines Transkriptionsauftrags ändert, EventBridge wird automatisch ein Ereignis an einen Event-Stream gesendet. Sie erstellen Regeln, die die Ereignisse, die Sie im Ereignis-Stream überwachen möchten, sowie die Aktion definieren, die EventBridge im Falle solcher Ereignisse ausführen soll. Beispiel: Das Ereignis an einen anderen Service (oder ein anderes Ziel) weiterleiten, der/das dann eine Aktion ausführen kann. Sie könnten beispielsweise eine Regel so konfigurieren, dass ein Ereignis an eine AWS Lambda Funktion weitergeleitet wird, wenn ein Transkriptionsauftrag erfolgreich abgeschlossen wurde. Informationen zum Definieren von [EventBridge -Regeln](#) finden Sie in den folgenden Abschnitten.

Sie können Benachrichtigungen für Ereignisse über mehrere Kanäle erhalten, einschließlich E-Mail, Chat-Benachrichtigungen von [Amazon Q Developer in Chat-Anwendungen](#) oder [AWS Console Mobile Application](#)-Push-Benachrichtigungen. Benachrichtigungen werden auch im [Console Notifications Center](#) angezeigt. Wenn Sie Benachrichtigungen einrichten möchten, können Sie verwenden [AWS-Benutzerbenachrichtigungen](#). AWS-Benutzerbenachrichtigungen unterstützt die Aggregation, wodurch die Anzahl der Benachrichtigungen, die Sie bei bestimmten Ereignissen erhalten, reduziert werden kann.

Definierend EventBridge Regeln

Um EventBridge Regeln zu definieren, verwenden Sie den [AWS-Managementkonsole](#). Wenn Sie eine Regel definieren, verwenden Sie Amazon Transcribe als Servicenamen. Ein Beispiel für die Erstellung einer EventBridge Regel finden Sie unter [Amazon EventBridge Regeln](#).

Beachten Sie EventBridge vor der Verwendung die folgenden Definitionen:

- Ereignis – Ein Ereignis zeigt an, dass sich der Zustand eines Ihrer Transkriptionsaufträge geändert hat. Zum Beispiel wenn sich der `TranscriptionJobStatus` eines Auftrags von `IN_PROGRESS` in `COMPLETED` ändert.
- Ziel – Ein Ziel ist ein anderes AWS-Service, das ein Ereignis verarbeitet. Zum Beispiel AWS Lambda oder Amazon Simple Notification Service (Amazon SNS). Ein Ziel empfängt Ereignisse im JSON-Format.
- Regel — Eine Regel sucht nach eingehenden Ereignissen, EventBridge nach denen Sie Ausschau halten möchten, und leitet sie zur Verarbeitung an ein oder mehrere Ziele weiter. Wenn eine Regel ein Ereignis an mehrere Ziele weiterleitet, verarbeiten alle Ziele das Ereignis parallel. Eine Regel kann die an das Ziel gesendete JSON-Datei anpassen.

Amazon EventBridge Ereignisse werden nach bestem Wissen und Gewissen ausgegeben. Weitere Informationen zum Erstellen und Verwalten von Ereignissen in EventBridge finden Sie unter [Amazon EventBridge Ereignisse](#) im Amazon EventBridge Benutzerhandbuch.

Im Folgenden finden Sie ein Beispiel für eine EventBridge Regel Amazon Transcribe, die ausgelöst wird, wenn sich der Status eines Transkriptionsauftrags auf `COMPLETED` oder `FAILED` ändert.

```
{
  "source": [
    "aws.transcribe"
  ],
  "detail-type": [
    "Transcribe Job State Change"
  ],
  "detail": {
    "TranscriptionJobStatus": [
      "COMPLETED",
      "FAILED"
    ]
  }
}
```

Die Regel enthält die folgenden Felder:

- `source` – Die Quelle des Ereignisses. Denn Amazon Transcribe das ist immer `aws.transcribe` so.
- `detail-type` – Ein Bezeichner für die Details des Ereignisses. Für Amazon Transcribe ist dies immer `Transcribe Job State Change`.

- `detail` – Der neue Auftragsstatus des Transkriptionsauftrags. In diesem Beispiel löst die Regel ein Ereignis aus, wenn sich der Auftragsstatus zu `COMPLETED` oder `FAILED` ändert.

Amazon Transcribe Veranstaltungen

Amazon EventBridge protokolliert mehrere Amazon Transcribe Ereignisse:

- [Ereignisse in Transkriptionsaufträgen](#)
- [Sprachenidentifikationsereignisse](#)
- [Call-Analytics-Ereignisse aufrufen](#)
- [Call-Analytics-nach-Anruf-Ereignissen aufrufen](#)
- [Ereignisse im Wortschatz](#)

Diese Ereignisse enthalten alle die folgenden gemeinsamen Felder:

- `version`: Die Version der Ereignisdaten. Dieser Wert ist immer 0.
- `id`: Eine eindeutige Kennung, die von EventBridge für das Ereignis generiert wurde.
- `detail-type`: Ein Bezeichner für die Details des Ereignisses. Beispiel, `Transcribe Job State Change`.
- `source`: Die Quelle des Ereignisses. Denn Amazon Transcribe das ist immer `soaws.transcribe`.
- `account`: Die AWS-Konto ID des Kontos, das den API-Aufruf generiert hat.
- `time`: Das Datum und die Uhrzeit, zu der das Ereignis übertragen wird.
- `region`: Die, AWS-Region in der die Anfrage gestellt wird.
- `resources`: Die von dem API-Aufruf verwendeten Ressourcen. Für Amazon Transcribe, dieses Feld ist immer leer.
- `detail`: Zusätzliche Details zum Ereignis.
 - `FailureReason`: Dieses Feld ist vorhanden, wenn der Zustand oder der Status geändert wird zu `FAILED`, und beschreibt den Grund für den `FAILED`-Zustand oder -Status.
 - Jeder Ereignistyp hat zusätzliche eindeutige Felder, die unter `detail` angezeigt werden. Diese eindeutigen Felder werden in den folgenden Abschnitten nach jedem Ereignisbeispiel definiert.

Ereignisse in Transkriptionsaufträgen

Wenn sich der Status eines Jobs von IN_PROGRESS zu COMPLETED oder ändert FAILED, Amazon Transcribe wird ein Ereignis generiert. Um den Job zu identifizieren, der den Zustand geändert und das Ereignis in Ihrem Ziel ausgelöst hat, verwenden Sie das Feld `TranscriptionJobName` des Ereignisses. Ein Amazon Transcribe Ereignis enthält die folgenden Informationen. Ein `FailureReason`-Feld wird unter `detail` hinzugefügt, wenn Ihr Transkriptionsjobstatus FAILED lautet.

Beachten Sie, dass dieses Ereignis nur für den [StartTranscriptionJob](#)-API-Vorgang gilt.

```
{
  "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`: Der eindeutige Name, den Sie für Ihren Transkriptionsauftrag gewählt haben.
- `TranscriptionJobStatus` : Der Status des Transkriptionsauftrags. Mögliche Werte sind COMPLETED oder FAILED.

Sprachenidentifikationsereignisse

Wenn Sie die [automatische Sprachenidentifikation](#) aktivieren, generiert Amazon Transcribe ein Ereignis, wenn der Zustand der Sprachidentifikation COMPLETED oder lautet FAILED. Um den Job zu identifizieren, der den Zustand geändert und das Ereignis in Ihrem Ziel ausgelöst hat, verwenden Sie das Feld `JobName` des Ereignisses. Ein Amazon Transcribe -Ereignis enthält die folgenden Informationen: Ein `FailureReason`-Feld wird unter `detail` hinzugefügt, falls Ihre Sprachidentifikation den Status FAILED hat.

Beachten Sie, dass dieses Ereignis nur für den [StartTranscriptionJob](#)-API-Vorgang gilt, wenn der [LanguageIdSettings](#)-Parameter enthalten ist.

```
{
  "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: Für Transkriptionsaufträge muss dieser Wert TranscriptionJob sein.
- JobName: Der eindeutige Name, den Sie für Ihren Transkriptionsauftrag gewählt haben.
- LanguageIdentificationStatus: Der Status der Sprachidentifikation in einem Transkriptionsauftrag. Mögliche Werte sind COMPLETED oder FAILED.

Call-Analytics-Ereignisse aufrufen

Wenn der Zustand eines [Call-Analytics-Auftrags](#) von IN_PROGRESS zu COMPLETED oder FAILED wechselt, erzeugt Amazon Transcribe ein Ereignis. Um den Call-Analytics-Auftrag zu identifizieren, der den Zustand geändert hat und das Ereignis in Ihrem Ziel auslöst, verwenden Sie das Feld JobName des Ereignisses. Ein Amazon Transcribe -Ereignis enthält die folgenden Informationen: Ein FailureReason-Feld wird unter detail hinzugefügt, wenn Ihr Call-Analytics-Auftragsstatus FAILED lautet.

Beachten Sie, dass dieses Ereignis nur für den [StartCallAnalyticsJob](#)-API-Vorgang gilt.

```
{
  "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:** Der eindeutige Name Ihres Call-Analytics-Transkriptionsauftrags.
- **JobStatus:** Der eindeutige Status Ihres Call-Analytics-Transkriptionsauftrags. Dies kann entweder COMPLETED oder FAILED sein.
- **FailureReason:** Dieses Feld ist nur vorhanden, wenn dies der JobStatus Fall ist FAILED, und beschreibt den Grund für den Fehler.
- **AnalyticsJobDetails:** Die Details Ihres Call Analytics-Transkriptionsauftrags, einschließlich Informationen zu übersprungenen Analysefunktionen.

Call-Analytics-nach-Anruf-Ereignissen aufrufen

Wenn eine [nach-Anruf-Analytics](#)-Transkription den Zustand von IN_PROGRESS zu COMPLETED oder FAILED, Amazon Transcribe ändert, wird ein Ereignis erzeugt. Um den Call-Analytics-Auftrag nach dem Gespräch zu identifizieren, der den Zustand geändert und das Ereignis in Ihrem Ziel ausgelöst hat, verwenden Sie das Feld `StreamingSessionId` des Ereignisses.

Beachten Sie, dass dieses Ereignis nur für den [StartCallAnalyticsStreamTranscription](#)-API-Vorgang gilt, wenn der [PostCallAnalyticsSettings](#)-Parameter enthalten ist.

Ein COMPLETED-Ereignis enthält die folgenden Informationen:

```

{
  "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"
  }
}
}

```

Ein FAILED-Ereignis enthält die folgenden Informationen:

```

{
  "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:** Die Identifikationsnummer, die Ihrer Call-Analytics-Transkriptionsanfrage in Echtzeit zugewiesen wurde.

- **PostCallStatus:** Der Status Ihrer Call-Analytics-Transkription nach Anrufen. Dies kann entweder COMPLETED oder FAILED sein.
- **Transcript:** Die URI Ihrer redigierten und unredigierten Transkripte.
- **Media:** Die URI Ihrer redigierten und unredigierten Audiodateien.

AWS HealthScribe Ereignisse nach der Stream-Analyse

Wenn sich bei einem Analysevorgang AWS HealthScribe nach dem Stream ein Status ändert, z. B. wenn von IN_PROGRESS zu [ClinicalNoteGenerationResult](#) gewechselt wird, wird ein Ereignis mit den folgenden Informationen generiert:

```
{
  "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:** Der Typ des Post-Stream-Analysevorgangs, der das Ereignis generiert hat. Der Inhalt des Ergebnisobjekts variiert je nach UpdateType
- **SessionId:** Die Identifikationsnummer für Ihren AWS HealthScribe Stream. Verwenden Sie diese ID, um die ursprüngliche Streaming-Sitzung zu identifizieren und dann die Post-Stream-Analyse zu finden, die das Ereignis generiert hat.

- **Status:** Der Status des Analysevorgangs nach dem Stream. Mögliche Werte sind `IN_PROGRESS`, `COMPLETED` oder `FAILED`.
- **ClinicalNoteOutputLocation:** Die URI des Amazon S3 S3-Ausgabe-Buckets für `ClinicalNoteGenerationResult`.
- **TranscriptOutputLocation:** Die URI Ihres Transkripts.

Ereignisse im Wortschatz

Wenn der Status eines [benutzerdefinierten Vokabulars](#) von `PENDING` zu `READY` oder `FAILED` wechselt, Amazon Transcribe wird ein Ereignis generiert. Um das benutzerdefinierte Vokabular zu identifizieren, das den Zustand geändert und das Ereignis in Ihrem Ziel ausgelöst hat, verwenden Sie das Feld `VocabularyName` des Ereignisses. Ein Amazon Transcribe Ereignis enthält die folgenden Informationen. Ein `FailureReason`-Feld wird unter `detail` hinzugefügt, wenn Ihr benutzerdefinierter Wortschatzzustand `FAILED` lautet.

Note

Dieses Ereignis gilt nur für den [CreateVocabulary](#) API-Vorgang.

```
{
  "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:** Der eindeutige Name Ihres benutzerdefinierten Vokabulars.
- **VocabularyState:** Der Verarbeitungszustand Ihres benutzerdefinierten Vokabulars. Mögliche Werte sind `READY` oder `FAILED`.

Konformitätsvalidierung für Amazon Transcribe

Informationen darüber, ob AWS-Service ein [AWS-Services in den Geltungsbereich bestimmter Compliance-Programme fällt](#), finden Sie unter [Umfang nach Compliance-Programm AWS-Services unter](#) . Wählen Sie dort das Compliance-Programm aus, an dem Sie interessiert sind. Allgemeine Informationen finden Sie unter [AWS Compliance-Programme AWS](#) .

Sie können Prüfberichte von Drittanbietern unter [herunterladen AWS Artifact](#). Weitere Informationen finden Sie unter [Berichte herunterladen unter](#) .

Ihre Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS-Services hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften ab. Weitere Informationen zu Ihrer Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS-Services finden Sie in der [AWS Sicherheitsdokumentation](#).

Resilienz in Amazon Transcribe

Die AWS globale Infrastruktur basiert auf AWS-Regionen Availability Zones. AWS-Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, die über Netzwerke mit niedriger Latenz, hohem Durchsatz und hoher Redundanz miteinander verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Zonen ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen zu Availability Zones AWS-Regionen und Availability Zones finden Sie unter [AWS Globale](#) Infrastruktur.

Infrastruktursicherheit in Amazon Transcribe

Als verwalteter Dienst Amazon Transcribe ist er durch AWS globale Netzwerksicherheit geschützt. Informationen zu AWS Sicherheitsdiensten und zum AWS Schutz der Infrastruktur finden Sie unter [AWS Cloud-Sicherheit](#). Informationen zum Entwerfen Ihrer AWS Umgebung unter Verwendung der bewährten Methoden für die Infrastruktursicherheit finden Sie unter [Infrastructure Protection](#) in Security Pillar AWS Well-Architected Framework.

Sie verwenden AWS veröffentlichte API-Aufrufe für den Zugriff Amazon Transcribe über das Netzwerk. Kunden müssen Folgendes unterstützen:

- Transport Layer Security (TLS). Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Cipher-Suites mit Perfect Forward Secrecy (PFS) wie DHE (Ephemeral) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Diffie-Hellman Die meisten modernen Systeme wie Java 7 und höher unterstützen diese Modi.

Schwachstellenanalyse und -management in Amazon Transcribe

Für Konfiguration und IT-Steuerung sind Sie, unser Kunde, gemeinsam verantwortlich. AWS Weitere Informationen finden Sie im [Modell der AWS gemeinsamen Verantwortung](#).

Amazon Transcribe und Schnittstellen-VPC-Endpunkte (AWS PrivateLink)

Sie können eine private Verbindung zwischen Ihrer VPC herstellen und Amazon Transcribe einen VPC-Schnittstellen-Endpunkt erstellen. Schnittstellenendpunkte basieren auf einer Technologie [AWS PrivateLink](#), mit der Sie privat auf Amazon Transcribe APIs zugreifen können, ohne dass ein Internet-Gateway, ein NAT-Gerät, eine VPN-Verbindung oder eine Verbindung erforderlich ist. Direct Connect Instances in Ihrer VPC benötigen keine öffentlichen IP-Adressen, um mit Amazon Transcribe APIs zu kommunizieren. Datenverkehr zwischen Ihrer VPC und Amazon Transcribe verlässt das Amazon-Netzwerk nicht.

Jeder Schnittstellenendpunkt wird durch eine oder mehrere [Elastic-Netzwerk-Schnittstellen](#) in Ihren Subnetzen dargestellt.

Weitere Informationen finden Sie unter [Schnittstellen-VPC-Endpunkte \(AWS PrivateLink\)](#) im Amazon-VPC-Benutzerhandbuch.

Überlegungen zu Amazon Transcribe VPC-Endpunkte

Bevor Sie einen Schnittstellen-VPC-Endpunkt für einrichten Amazon Transcribe, stellen Sie sicher, dass Sie die [Eigenschaften und Einschränkungen der Schnittstellenendpunkte](#) im Amazon VPC Benutzerhandbuch lesen.

Amazon Transcribe unterstützt Aufrufe aller API-Aktionen von Ihrer VPC aus.

Erstellen eines Schnittstellen-VPC-Endpunkts für Amazon Transcribe

Sie können mit dem Amazon VPC AWS-Managementkonsole oder AWS CLI einen VPC-Endpunkt für den Amazon Transcribe Service erstellen. Weitere Informationen finden Sie unter [Erstellen eines Schnittstellenendpunkts](#) im Amazon VPC -Benutzerhandbuch.

Erstellen Sie für Batch-Transkriptionen in Amazon Transcribe einen VPC-Endpunkt mit dem folgenden Dienstnamen:

- `com.amazonaws.us-west-2.transkribieren`

Erstellen Sie für Streaming-Transkriptionen Amazon Transcribe einen VPC-Endpunkt mit dem folgenden Dienstnamen:

- `com.amazonaws.us-west-2.transkribiert Streaming`

Wenn Sie privates DNS für den Endpunkt aktivieren, können Sie API-Anfragen an die Amazon Transcribe Verwendung seines Standard-DNS-Namens für stellen AWS-Region, zum Beispiel. `transcribestreaming.us-east-2.amazonaws.com`

Weitere Informationen finden Sie unter [Zugriff auf einen Service über einen Schnittstellenendpunkt](#) im Amazon VPC -Benutzerhandbuch.

Erstellen einer VPC-Endpunktrichtlinie für Amazon Transcribe

Sie können Ihrem VPC-Endpunkt eine Endpunktrichtlinie hinzufügen, die den Zugriff auf den Streaming-Dienst oder den Batch-Transkriptionsdienst von steuert. Amazon Transcribe Die Richtlinie gibt die folgenden Informationen an:

- Prinzipal, der die Aktionen ausführen kann.
- Aktionen, die ausgeführt werden können
- Die Ressourcen, für die Aktionen ausgeführt werden können.

Weitere Informationen finden Sie unter [Steuerung des Zugriffs auf Services mit VPC-Endpunkten](#) im Amazon VPC -Benutzerhandbuch.

Beispiel: VPC-Endpunktrichtlinie für Amazon Transcribe Batch-Transkriptionsaktionen

Es folgt ein Beispiel für eine Endpunktrichtlinie für eine Batch-Transkription in Amazon Transcribe. Wenn diese Richtlinie an einen Endpunkt angefügt wird, gewährt sie Zugriff auf die aufgelisteten Amazon Transcribe -Aktionen für alle Prinzipale auf allen Ressourcen.

```
{  
  "Statement": [  

```

```
{
  "Principal": "*",
  "Effect": "Allow",
  "Action": [
    "transcribe:StartTranscriptionJob",
    "transcribe:ListTranscriptionJobs"
  ],
  "Resource": "*"
}
```

Beispiel: VPC-Endpunktrichtlinie für Amazon Transcribe Streaming-Transkriptionsaktionen

Im Folgenden finden Sie ein Beispiel für eine Endpunktrichtlinie für eine Streaming-Transkription in Amazon Transcribe. Wenn diese Richtlinie an einen Endpunkt angefügt wird, gewährt sie Zugriff auf die aufgelisteten Amazon Transcribe -Aktionen für alle Prinzipale auf allen Ressourcen.

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Allow",
      "Action": [
        "transcribe:StartStreamTranscription",
        "transcribe:StartStreamTranscriptionWebsocket"
      ],
      "Resource": "*"
    }
  ]
}
```

Gemeinsam genutzte Subnetze

Sie können VPC-Endpunkte in Subnetzen, die mit Ihnen gemeinsam genutzt werden, nicht erstellen, beschreiben, ändern oder löschen. Sie können die VPC-Endpunkte jedoch in Subnetzen verwenden, die mit Ihnen geteilt werden. Informationen zur VPC-Sharing finden Sie im [Leitfaden unter Teilen Ihrer VPC mit anderen Konten](#). Amazon Virtual Private Cloud

Bewährte Sicherheitsmethoden für Amazon Transcribe

Die folgenden bewährten Methoden sind allgemeine Richtlinien und keine vollständige Sicherheitslösung. Da diese bewährten Verfahren für Ihre Umgebung möglicherweise nicht geeignet oder ausreichend sind, sollten Sie sie eher als hilfreiche Überlegungen denn als Vorschriften betrachten.

- Verwenden Sie Datenverschlüsselung, z. B. den AWS KMS Verschlüsselungskontext

AWS KMS Der Verschlüsselungskontext ist eine Zuordnung von Schlüssel/Wert-Paaren im Klartext-Format, die nicht geheim sind. Diese Karte stellt zusätzliche authentifizierte Daten dar, die als Verschlüsselungskontextpaare bekannt sind und eine zusätzliche Sicherheitsebene für Ihre Daten bieten.

Weitere Informationen finden Sie unter [AWS KMS Verschlüsselungskontext](#).

- Verwenden Sie, wann immer möglich, temporäre Anmeldeinformationen

Verwenden Sie nach Möglichkeit temporäre Anmeldeinformationen anstelle von langfristigen Anmeldeinformationen, wie z. B. Zugangsschlüsseln. Für Szenarien, in denen Sie IAM Benutzer mit programmatischem Zugriff und langfristigen Anmeldeinformationen benötigen, empfehlen wir, die Zugriffsschlüssel abwechselnd zu verwenden. Regelmäßig rotierende langfristige Anmeldeinformationen helfen Ihnen dabei, sich mit dem Prozess vertraut zu machen. Dies ist nützlich, wenn Sie sich jemals in einer Situation befinden, in der Sie Anmeldeinformationen rotieren müssen, z. B. wenn ein Mitarbeiter Ihr Unternehmen verlässt. Wir empfehlen Ihnen, IAM -Zugriff auf zuletzt verwendete Informationen zu verwenden, um Zugriffsschlüssel sicher zu rotieren und zu entfernen.

Weitere Informationen finden Sie unter [Rotierende Zugriffsschlüssel](#) und [Bewährte Sicherheitspraktiken in IAM](#).

- Verwenden Sie IAM Rollen für Anwendungen und AWS Dienste, die Zugriff benötigen Amazon Transcribe

Verwenden Sie eine IAM Rolle, um temporäre Anmeldeinformationen für Anwendungen oder Dienste zu verwalten, auf die zugegriffen Amazon Transcribe werden muss. Wenn Sie eine Rolle verwenden, müssen Sie keine langfristigen Anmeldeinformationen wie Passwörter oder Zugriffsschlüssel an eine Amazon EC2 Instanz oder einen AWS Dienst verteilen. IAM Rollen können temporäre Berechtigungen bereitstellen, die Anwendungen verwenden können, wenn sie Anfragen an AWS Ressourcen stellen.

Weitere Informationen finden Sie unter [IAM -Rollen](#) und [Gemeinsame Szenarien für Rollen: Benutzer, Anwendungen und Services](#).

- Tag-basierte Zugriffssteuerung verwenden

Sie können Tags verwenden, um den Zugriff auf Ihre Daten zu kontrollieren AWS-Konten. In Amazon Transcribe, können Tags hinzugefügt werden zu: Transkriptionsaufträgen, benutzerdefinierten Vokabularen, benutzerdefinierten Wortschatzfiltern und benutzerdefinierten Sprachmodellen.

Weitere Informationen finden Sie unter [Tag-based Zugriffskontrolle](#).

- Verwenden Sie AWS Überwachungstools

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Sicherheit, Verfügbarkeit Amazon Transcribe und Leistung Ihrer AWS Lösungen. Sie können überwachen Amazon Transcribe mit CloudTrail.

Weitere Informationen finden Sie unter [Überwachen Amazon Transcribe mit AWS CloudTrail](#).

- Aktivieren AWS Config

AWS Config kann die Konfigurationen Ihrer AWS Ressourcen beurteilen, prüfen und auswerten. Mithilfe AWS Config können Sie Änderungen an Konfigurationen und Beziehungen zwischen AWS Ressourcen überprüfen. Sie können auch die detaillierte Historie der Ressourcenkonfiguration untersuchen und feststellen, ob Sie die in Ihren internen Richtlinien festgelegten Konfigurationen insgesamt einhalten. Dadurch können Sie die Compliance-Prüfung, die Sicherheitsanalyse, das Änderungsmanagement und die Fehlerbehebung bei Betriebsabläufen vereinfachen.

Weitere Informationen finden Sie unter [Was ist AWS Config?](#)

Amazon Transcribe Medical

Amazon Transcribe Medical ist ein automatischer Spracherkennungsdienst (ASR), der für medizinisches Fachpersonal konzipiert wurde, das medizinische Sprache transkribieren möchte, z. B. vom Arzt diktierte Notizen, die Überwachung der Arzneimittelsicherheit, telemedizinische Termine oder Gespräche zwischen Arzt und Patient. Amazon Transcribe Medizinische Daten sind entweder durch Echtzeit-Streaming (über Mikrofon) oder durch Transkription einer hochgeladenen Datei (Batch) verfügbar.

Important

Amazon Transcribe Medizin ist kein Ersatz für professionelle medizinische Beratung, Diagnose oder Behandlung. Identifizieren Sie den richtigen Konfidenzschwellenwert für Ihren Anwendungsfall, und verwenden Sie hohe Konfidenzschwellenwerte in Situationen, die eine hohe Genauigkeit erfordern. In bestimmten Anwendungsfällen sollten die Ergebnisse von entsprechend geschulten menschlichen Gutachtern überprüft und verifiziert werden. Amazon Transcribe Medizinische Transkriptionen sollten für die Patientenversorgung nur verwendet werden, nachdem sie von geschultem medizinischem Fachpersonal auf Richtigkeit und fundiertes medizinisches Urteilsvermögen überprüft wurden.

Amazon Transcribe Medical arbeitet nach einem Modell der geteilten Verantwortung, wobei AWS es für den Schutz der Infrastruktur verantwortlich ist, auf der Amazon Transcribe Medical betrieben wird, und Sie für die Verwaltung Ihrer Daten verantwortlich sind. Weitere Informationen finden Sie unter [Modell der geteilten Verantwortung](#).

Amazon Transcribe Medical ist in US-Englisch (en-US) erhältlich.

Die besten Ergebnisse erzielen Sie, wenn Sie ein verlustfreies Audioformat wie FLAC oder WAV mit PCM 16-Bit-Kodierung verwenden. Amazon Transcribe Medical unterstützt Sampleraten von 16.000 Hz oder höher.

Für die Analyse Ihrer Transkripte können Sie andere verwenden AWS-Services, z. [Amazon Comprehend Medical](#)

Unterstützte Fachgebiete

Fachgebiet	Sub-specialty	Audioeingang
Kardiologie	Keine	nur Streaming
Neurologie	Keine	nur Streaming
Onkologie	Keine	nur Streaming
Grundversorgung	Allgemeinmedizin	Batch, Streaming
Grundversorgung	Innere Medizin	Batch, Streaming
Grundversorgung	Geburtshilfe und Gynäkologie () OB-GYN	Batch, Streaming
Grundversorgung	Pädiatrie	Batch, Streaming
Radiologie	Keine	nur Streaming
Urologie	Keine	nur Streaming

Regionale Verfügbarkeit und Quotas

Call Analytics wird in den folgenden Bereichen unterstützt AWS-Regionen:

Region	Transkriptionstyp
af-south-1 (Kapstadt)	Batch
ap-east-1 (Hongkong)	Batch
ap-northeast-1 (Tokyo)	Batch, Streaming
ap-northeast-2 (Seoul)	Batch, Streaming
ap-south-1 (Mumbai)	Batch
ap-southeast-1 (Singapur)	Batch

Region	Transkriptionstyp
ap-southeast-2 (Sydney)	Batch, Streaming
ca-central-1 (Kanada, Zentral)	Batch, Streaming
eu-central-1 (Frankfurt)	Batch, Streaming
eu-nord-1 (Stockholm)	Batch
eu-west-1 (Irland)	Batch, Streaming
eu-west-2 (London)	Batch, Streaming
eu-west-3 (Paris)	Batch
me-south-1 (Bahrain)	Batch
sa-east-1 (São Paulo)	Batch, Streaming
us-east-1 (N. Virginia)	Batch, Streaming
us-east-2 (Ohio)	Batch, Streaming
us-gov-east-1 (,) GovCloud US-East	Batch, Streaming
us-gov-west-1 (,) GovCloud US-West	Batch, Streaming
us-west-1 (San Francisco)	Batch
us-west-2 (Oregon)	Batch, Streaming

Beachten Sie, dass die Unterstützung der Regionen unterschiedlich ist für [Amazon Transcribe](#), Amazon Transcribe Medical, und [Call Analytics](#).

Um die Endpunkte für jede unterstützte Region zu erhalten, siehe [Dienstendpunkte](#) in der Allgemeinen AWS -Referenz.

Eine Liste der Quotas, die sich auf Ihre Transkriptionen beziehen, finden Sie in der Allgemeinen AWS -Referenz unter [Service Quotas](#). Einige Quotas können auf Anfrage geändert werden. Steht in der

Spalte Anpassbar Ja, können Sie eine Erhöhung beantragen. Wählen Sie dazu den angegebenen Link.

Medizinische Fachgebiete und Begriffe

Beim Erstellen eines medizinischen Transkriptionsauftrags geben Sie die Sprache, das medizinische Fachgebiet und den Audiotyp der Quelldatei an. Sie geben US English (en-US) als Sprache und PRIMARYCARE als medizinisches Fachgebiet ein. Die Eingabe der Primärversorgung als Wert ermöglicht es Ihnen, Transkriptionen aus Quellaudio in den folgenden medizinischen Fachrichtungen zu generieren:

- Allgemeinmedizin
- Innere Medizin
- Geburtshilfe und Gynäkologie () OB-GYN
- Pädiatrie

Sie haben als Audiotyp die Wahl zwischen Diktat und Gespräch. Wählen Sie Diktat für Audiodateien, bei denen der Arzt über eine Patientenvsiste oder eine Prozedur berichtet. Wählen Sie für Audiodateien, die ein Gespräch zwischen einem Arzt und einem Patienten oder ein Gespräch zwischen Ärzten beinhalten, Konversation aus.

Um die Ausgabe Ihres Transkriptionsauftrags zu speichern, wählen Sie einen Amazon S3 Bucket aus, den Sie bereits erstellt haben. Weitere Informationen zu Amazon S3 Buckets finden Sie unter [Erste Schritte mit](#) Amazon Simple Storage Service

Im Folgenden finden Sie die Mindestanzahl der Anfrageparameter, die in das JSON-Beispiel eingegeben werden müssen.

```
{
  "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 Mit Medical können Sie alternative Transkriptionen erstellen. Weitere Informationen finden Sie unter [Erstellen alternativer Transkriptionen](#).

Sie können auch die Partitionierung von Sprechern aktivieren oder Kanäle in Ihrem Audio identifizieren. Weitere Informationen erhalten Sie unter [Aktivieren der Sprecherpartitionierung](#) und [Transkribieren von Mehrkanal-Audio](#).

Transkribieren von medizinischen Begriffen und Messungen

Amazon Transcribe Medical kann medizinische Begriffe und Messungen transkribieren. Amazon Transcribe Medical gibt Abkürzungen für gesprochene Begriffe aus. Zum Beispiel wird „Blutdruck“ als „BP“ transkribiert. In der Tabelle auf dieser Seite finden Sie eine Liste der Konventionen, die Amazon Transcribe in der Medizin für medizinische Begriffe und Maße verwendet werden. Die Spalte Gesprochener Begriff bezieht sich auf den im Quellaudio gesprochenen Begriff. Die Spalte Output (Ausgabe) bezieht sich auf die Abkürzung, die Sie in den Transkriptionsergebnissen sehen.

Hier können Sie sehen, wie die Begriffe, die im Quellaudio gesprochen werden, mit der Transkriptionsausgabe übereinstimmen.

In der Audioquelle gesprochener Begriff	In der Ausgabe verwendete Abkürzung	Beispielausgabe
Grad Celsius	C	Die Temperatur des Patienten beträgt 37,4 C.
Celsius	C	Die Temperatur des Patienten beträgt 37,4 C.
Fahrenheit	F	Die Temperatur des Patienten beträgt 101 F.
Gramm	g	Eine Masse von 100 g wurde aus dem Patienten extrahiert.
Meter	m	Der Patient ist 1,8 m groß.
Fuß	ft	Der Patient ist 6 Fuß groß.
Kilo	kg	Der Patient wiegt 80 kg.

In der Audioquelle gesprochener Begriff	In der Ausgabe verwendete Abkürzung	Beispielausgabe
Kilogramm	kg	Der Patient wiegt 80 kg.
c c	cc	Patient erhielt 100 cc Kochsalzlösung.
Kubikzentimeter	cc	Patient erhielt 100 cc Kochsalzlösung.
Milliliter	mL	Patient schied 100 ml Urin aus.
Blutdruck	BP	Patienten-Blutdruck wurde erhöht.
b p	BP	Patienten-Blutdruck wurde erhöht.
X über Y	X/Y	Der Blutdruck des Patienten war 120/80.
Schläge pro Minute	BPM	Patient hatte Vorhofflimmern mit einer Herzfrequenz von 160 BPM.
Schläge pro Minute	BPM	Patient hatte Vorhofflimmern mit einer Herzfrequenz von 160 BPM.
O 2	O2	O2-Sättigung des Patienten betrug 98 %.
CO2	CO2	Patient benötigte Atemunterstützung bei erhöhtem CO2-Wert.

In der Audioquelle gesprochener Begriff	In der Ausgabe verwendete Abkürzung	Beispielausgabe
postoperativ	POSTOP	Patient kam zur POSTOP-Auswertung.
postoperativ	POSTOP	Patient kam zur POSTOP-Auswertung.
Computertomografie	CT-Scan	Patientenindikation einer Hirnblutung erforderte ein CT-Scan.
Puls 80	P 80	Die Vitalwerte der Patienten waren P 80, R 17,...
Respiration 17	R 17	Die Vitalwerte der Patienten waren P 80, R 17,...
Ein- und Ausgabe	I/O	Der Patient hatte einen I/O Sinusrhythmus
L fünf	L5	Lumbalpunktion wurde zwischen L4 und L5 durchgeführt

Transkription von Zahlen

Amazon Transcribe Die Medizin transkribiert Zahlen als Zahlen statt als Wörter. Beispiel: Die gesprochene Zahl „Eintausendzweihundertzweiundvierzig“ wird als 1242 transkribiert.

Die Transkription der Zahlen erfolgt nach den folgenden Regeln.

Regel	Description
Konvertieren von Kardinalzahlen größer als 10 zu Ziffern.	„Fünfundfünfzig“ > 55 „Einhundert“ > 100 „Eintausendundeinunddreißig“ > 1031

Regel	Description
	„Einhundertdreißig Millionen vierhundertsechsfünftausend siebenhundertneun“ > 123.456.789
Konvertieren von Kardinalzahlen gefolgt von „Million“ oder „Milliarde“ in Zahlen oder gefolgt von einem Wort, wenn auf „Million“ oder „Milliarde“ keine Zahl folgt.	„hundert Millionen“ > 100 Millionen „eine Milliarde“ > 1 Milliarde „zwei Punkt drei Millionen“ > 2,3 Millionen
Konvertieren von Ordinalzahlen größer als 10 zu Ziffern.	„Der Dreißigste“ > 43. „Sechszwanzigste Straße“ > 26. Straße
Konvertieren von Bruchzahlen in ihr numerisches Format.	„ein Viertel“ > 1/4 „drei Sechzehntel“ > 3/16 „eine halbe“ > 1/2 „ein Hundertstel“ > 1/100
Konvertieren von Zahlen unter 10 zu Ziffern, wenn mehr als drei hintereinander folgen.	„drei vier fünf“ > 345 „Meine Telefonnummer ist vier zwei fünf fünf fünf fünf eins zwei eins zwei“ > 4255551212
Dezimalstellen werden durch „Komma“ angegeben	„Dreihundertunddrei Komma Fünf“ > 303,5 „Drei Komma Zwei Drei“ > 3,23 „Null Komma Vier“ > 0,4 „Komma Drei“ > 0,3
Konvertieren des Wortes „Prozent“ nach einer Zahl in das Prozentzeichen (%).	„Dreißig Prozent“ > 23 % „Dreißig Komma Vier Fünf Prozent“ > 23,45 %
Konvertieren der Wörter „Dollar“, „US-Dollar“, „Australischer Dollar“, „AUD“ oder „USD“ nach einer Zahl in ein Dollarzeichen vor der Zahl.	„Ein Dollar und fünfzehn Cent“ > \$ 1,15 „dreißig USD“ > 23 USD „dreißig australische Dollar“ > \$23

Regel	Description
Konvertieren der Wörter „Pfund“ oder „Milligramm“ in „lbs“ oder „mg“.	• „dreiundzwanzig Pfund“ > 23 £ • „fünfundvierzig Milligramm“ > 45 mg
Konvertieren der Wörter „Rupien“, „Indische Rupien“ oder „INR“ nach einer Zahl in das Rupien-Zeichen (#) vor der Zahl.	• „dreiundzwanzig Rupien“ > #23 • „fünfzig Rupien dreißig Paise“ > #50,30
Konvertieren von Zeiten in Zahlen.	• „Sieben Uhr morgens Eastern Standard Time“ > 07:00 Eastern Standard Time • „12:30“ > 12:30 Uhr
Kombinieren mit zwei Ziffern ausgedrückten Jahreszahlen zu vier Ziffern. Nur gültig für das 20., 21. und 22. Jahrhundert.	• „Neunzehnhundertundzweiundsechzig“ > 1962 • „das Jahr ist Zwanzig Zwölf“ > das Jahr ist 2012 • „Zwanzig Neunzehn“ > 2019 • „Einundzwanzig Dreißig“ > 2130
Umwandeln von Datumsangaben in Zahlen.	• „Fünfter Mai Zweitausendundzwölf“ > 5. Mai 2012 • „Fünfter Mai 2012“ > 5/5/2012 • „Fünfter Mai 2012“ > 5. Mai 2012
Trennen von Zahlenbereichen durch „bis“	• „Dreiundzwanzig bis Siebenundzwanzig“ > 23 bis 27

Transkribieren eines medizinischen Gesprächs

Sie können Amazon Transcribe Medical verwenden, um ein medizinisches Gespräch zwischen einem Arzt und einem Patienten zu transkribieren, indem Sie entweder einen Batch-Transkriptionsjob oder einen Echtzeit-Stream verwenden. Mit Batch-Transkriptionsaufträgen können Sie Audiodateien transkribieren. Um sicherzustellen, dass Amazon Transcribe Medical die Transkriptionsergebnisse mit der höchstmöglichen Genauigkeit liefert, müssen Sie in Ihrem Transkriptionsjob oder Stream das medizinische Fachgebiet des Kliniklers angeben.

Sie können einen Arzt-Patientenbesuch in den folgenden medizinischen Fachgebieten transkribieren:

- Kardiologie – nur in Streaming-Transkription verfügbar
- Neurologie – nur in Streaming-Transkription verfügbar
- Onkologie – nur in Streaming-Transkription verfügbar
- Grundversorgung (Allgemeinmedizin/Hausarzt) – umfasst die folgenden Arten von Arztpraxen:
 - Allgemeinmedizin
 - Innere Medizin
 - Geburtshilfe und Gynäkologie () OB-GYN
 - Pädiatrie
- Urologie – nur in Streaming-Transkription verfügbar

Sie können die Transkriptionsgenauigkeit verbessern, indem Sie benutzerdefinierte medizinische Vokabulare verwenden. Informationen über die Funktionsweise von medizinischen benutzerdefinierten Vokabularen finden Sie unter [Verbesserung der Transkriptionsgenauigkeit mit benutzerdefinierten medizinischen Vokabularen](#).

Standardmäßig gibt Amazon Transcribe Medical die Transkription mit der höchsten Konfidenzstufe zurück. Wenn Sie es so konfigurieren möchten, dass alternative Transkriptionen zurückgegeben werden, siehe [Erstellen alternativer Transkriptionen](#).

Informationen darüber, wie Zahlen und medizinische Maße in der Transkriptionsausgabe erscheinen, finden Sie unter [Transkription von Zahlen](#) und [Transkribieren von medizinischen Begriffen und Messungen](#).

Themen

- [Transkribieren einer Audiodatei eines medizinischen Gesprächs](#)
- [Transkribieren eines medizinischen Gesprächs in einem Echtzeit-Stream](#)
- [Aktivieren der Sprecherpartitionierung](#)
- [Transkribieren von Mehrkanal-Audio](#)

Transkribieren einer Audiodatei eines medizinischen Gesprächs

Verwenden Sie einen Batch-Transkriptionsauftrag, um Audiodateien von medizinischen Gesprächen zu transkribieren. Damit können Sie einen Arzt-Patienten-Dialog transkribieren. Sie können einen

Batch-Transkriptionsauftrag entweder über die [StartMedicalTranscriptionJob](#)-API oder AWS-Managementkonsole starten.

Wenn Sie einen medizinischen Transkriptionsauftrag mit der [StartMedicalTranscriptionJob](#)-API starten, geben Sie PRIMARYCARE als Wert für den Parameter Specialty an.

AWS-Managementkonsole

Um einen Dialog zwischen Arzt und Patient zu transkribieren (AWS-Managementkonsole)

Um den zur Transkription eines Gesprächs zwischen Arzt und Patient AWS-Managementkonsole zu verwenden, erstellen Sie einen Transkriptionsauftrag und wählen Sie als Audioeingabetyp „Konversation“ aus.

1. Melden Sie sich an der [AWS-Managementkonsole](#) an.
2. Wählen Sie im Navigationsbereich unter Amazon Transcribe Medizinisch die Option Transkriptionsaufträge aus.
3. Wählen Sie Job erstellen aus.
4. Geben Sie auf der Seite Auftragsdetails an unter Auftragseinstellungen Folgendes an.
 - a. Name – der Name des Transkriptionsauftrags.
 - b. Art des Audioeingangs - Konversation
5. Geben Sie für die übrigen Felder den Amazon S3 Speicherort Ihrer Audiodatei an und geben Sie an, wo Sie die Ausgabe Ihres Transkriptionsauftrags speichern möchten.
6. Wählen Sie Next.
7. Wählen Sie Erstellen aus.

API

So transkribieren Sie ein medizinisches Gespräch mit einem Batch-Transkriptionsauftrag (API)

- Für die [StartMedicalTranscriptionJob](#)-API geben Sie Folgendes an.
 - a. Geben Sie unter MedicalTranscriptionJobName einen Namen an, der in AWS-Konto eindeutig ist.
 - b. Geben Sie unter LanguageCode den Sprachcode an, der in Ihrer Audiodatei gesprochenen Sprache entspricht, sowie die Sprache Ihres Wortschatzfilters.

- c. Geben Sie unter `MediaFileUri`-Parameter des `Media`-Objekts den Namen der Audiodatei an, die Sie transkribieren möchten.
- d. Geben Sie unter `Specialty` das medizinische Fachgebiet des Arztes, der in der Audiodatei spricht, als `PRIMARYCARE` an.
- e. Legen Sie für `Type` die Option `CONVERSATION` fest.
- f. Unter `OutputBucketName` geben Sie den Amazon S3 -Bucket an, in dem die Transkriptionsergebnisse gespeichert werden.

Im Folgenden finden Sie eine Beispielanforderung, mit der AWS SDK für Python (Boto3) eine medizinische Konversation eines `PRIMARYCARE` Facharztes und eines Patienten transkribiert wird.

```
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)
```

Der folgende Beispielcode zeigt die Transkriptionsergebnisse eines Arzt-Patienten-Gesprächs.

```
{
  "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

Um ein medizinisches Gespräch mithilfe eines Batch-Transkriptionsauftrags zu transkribieren (AWS CLI)

- Führen Sie folgenden Code aus.

```
aws transcribe start-medical-transcription-job \  
--region us-west-2 \  
--cli-input-json file://example-start-command.json
```

Der folgende Code zeigt den Inhalt von `example-start-command.json`.

```
{  
  "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"  
}
```

Transkribieren eines medizinischen Gesprächs in einem Echtzeit-Stream

Sie können einen Audiostream einer medizinischen Konversation mit den Protokollen oder transkribieren. HTTP/2 [WebSocket](#) Hinweise zum Starten eines Streams mithilfe des WebSocket Protokolls finden Sie unter. [Einen WebSocket Stream einrichten](#) Verwenden Sie die [StartMedicalStreamTranscription](#)API, um einen HTTP/2 Stream zu starten.

Sie können Streaming-Audio in den folgenden medizinischen Fachgebieten transkribieren:

- Kardiologie

- Neurologie
- Onkologie
- Grundversorgung
- Urologie

Jedes medizinische Fachgebiet umfasst viele Arten von Verfahren und Terminen. Ärzte diktieren daher viele verschiedene Arten von Notizen. Verwenden Sie die folgenden Beispiele als Leitfaden, um den Wert des `specialty` URI-Parameters der WebSocket Anfrage oder des `Specialty` Parameters der [StartMedicalStreamTranscription](#) API anzugeben:

- Für elektrophysiologische oder echokardiografische Untersuchungen wählen Sie `CARDIOLOGY`.
- Für eine Untersuchung im Bereich medizinische Onkologie, chirurgische Onkologie oder Radio-Onkologie, wählen Sie `ONCOLOGY`.
- Für einen Arzt, der einen Patienten behandelt, der einen Schlaganfall hatte, entweder eine transitorische ischämische Attacke oder eine zerebrovaskuläre Attacke, wählen Sie `NEUROLOGY`.
- Für eine Beratung zum Thema Harninkontinenz wählen Sie `UROLOGY`.
- Für jährliche Kontrolluntersuchungen oder dringende Behandlungen wählen Sie `PRIMARYCARE`.
- Für stationäre Visiten von Krankenhausärzten wählen Sie `PRIMARYCARE`.
- Für Untersuchungen zu den Themen Fruchtbarkeit, Eileiterunterbindung, Einsetzen der Spirale oder Schwangerschaftsabbruch wählen Sie `PRIMARYCARE`.

AWS-Managementkonsole

Um ein medizinisches Streaming-Gespräch zu transkribieren (AWS-Managementkonsole)

Um ein Gespräch zwischen Arzt und Patient in Echtzeit AWS-Managementkonsole zu transkribieren, wählen Sie die Option, ein medizinisches Gespräch zu transkribieren, starten Sie den Stream und beginnen Sie, ins Mikrofon zu sprechen.

1. Melden Sie sich an der [AWS-Managementkonsole](#) an.
2. Wählen Sie im Navigationsbereich unter Medizinisch die Option Transkription aus. Amazon Transcribe Real-time
3. Wählen Sie Gespräch.
4. Wählen Sie unter Medizinisches Fachgebiet das Fachgebiet des Arztes.

5. Wählen Sie dann Start Streaming (Streamen starten).
6. Sprechen Sie in das Mikrofon.

Transkribieren einer medizinischen Konversation in einen Stream HTTP/2

Im Folgenden finden Sie die Syntax für die Parameter einer HTTP/2 Anfrage.

Um einen HTTP/2 Stream einer medizinischen Konversation zu transkribieren, verwenden Sie die [StartMedicalStreamTranscription](#)API und geben Sie Folgendes an:

- `LanguageCode` – Der Sprachcode. Der gültige Wert ist `en-US`
- `MediaEncoding` – Die für das Eingangsaudio verwendete Kodierung. Gültige Werte sind `pcm`, `ogg-opus` und `flac`.
- `Specialty` – Das Fachgebiet der medizinischen Fachkraft.
- `Type` – `CONVERSATION`

Um die Transkriptionsgenauigkeit bestimmter Begriffe in einem Echtzeit-Stream zu verbessern, verwenden Sie ein benutzerdefiniertes Vokabular. Um ein benutzerdefiniertes Vokabular zu aktivieren, setzen Sie den Wert des Parameters `VocabularyName` auf den Namen des benutzerdefinierten Vokabulars, das Sie verwenden möchten. Weitere Informationen finden Sie unter [Verbesserung der Transkriptionsgenauigkeit mit benutzerdefinierten medizinischen Vokabularen](#).

Um die Sprache von verschiedenen Sprechern zu beschriften, setzen Sie den Parameter `ShowSpeakerLabel` auf `true`. Weitere Informationen finden Sie unter [Aktivieren der Sprecherpartitionierung](#).

Weitere Informationen zum Einrichten eines HTTP/2 Streams zum Transkribieren einer medizinischen Konversation finden Sie unter [Einen HTTP/2 Stream einrichten](#)

Eine medizinische Konversation in einen Stream transkribieren WebSocket

Sie können eine WebSocket Anfrage verwenden, um ein medizinisches Gespräch zu transkribieren. Wenn Sie eine WebSocket Anfrage stellen, erstellen Sie eine vorsignierte URI. Dieser URI enthält die Informationen, die zum Einrichten des Audiostroms zwischen Ihrer Anwendung und Amazon Transcribe -Medical benötigt werden. Weitere Informationen zum Erstellen von WebSocket Anfragen finden Sie unter [Einen WebSocket Stream einrichten](#).

Verwenden Sie die folgende Vorlage, um Ihren vordefinierten URI zu erstellen.

```
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
```

Um die Transkriptionsgenauigkeit bestimmter Begriffe in einem Echtzeit-Stream zu verbessern, verwenden Sie ein benutzerdefiniertes Vokabular. Um ein benutzerdefiniertes Vokabular zu aktivieren, setzen Sie den Wert von `vocabulary-name` auf den Namen des benutzerdefinierten Vokabulars, das Sie verwenden möchten. Weitere Informationen finden Sie unter [Verbesserung der Transkriptionsgenauigkeit mit benutzerdefinierten medizinischen Vokabularen](#).

Um die Sprache von verschiedenen Sprechern zu beschriften, stellen Sie den Parameter `show-speaker-label` auf `true` ein. Weitere Informationen finden Sie unter [Aktivieren der Sprecherpartitionierung](#).

Weitere Informationen zur Erstellung von vorsignierten URIs finden Sie unter [Einen WebSocket Stream einrichten](#).

Aktivieren der Sprecherpartitionierung

Um die Lautsprecherpartitionierung in Amazon Transcribe Medical zu aktivieren, verwenden Sie die Lautsprecher-Diarisierung. So können Sie in der Transkriptionsausgabe sehen, was der Patient und was der Arzt gesagt hat.

Wenn Sie die Lautsprecher-Diarisierung aktivieren, Amazon Transcribe kennzeichnet Medical jede Sprecheräußerung mit einer eindeutigen Kennung für jeden Sprecher. Eine Äußerung ist eine

Sprecheinheit, die normalerweise durch Stille von anderen Äußerungen getrennt ist. Bei der Batch-Transkription könnte eine Äußerung des Arztes das Label `spk_0` und eine Äußerung des Patienten das Label `spk_1` erhalten.

Wenn sich eine Äußerung eines Sprechers mit einer Äußerung eines anderen Sprechers überschneidet, ordnet Amazon Transcribe -Medical sie in der Transkription nach ihren Anfangszeiten. Äußerungen, die in der eingegebenen Audiodatei überlappen, überlappen nicht in der ausgegebenen Transkription.

Sie können die Sprecherdiarisierung aktivieren, wenn Sie eine Audiodatei mit einem Batch-Transkriptionsauftrag oder in einem Echtzeit-Stream transkribieren.

Themen

- [Aktivieren der Sprecherpartitionierung bei Batch-Transkriptionen](#)
- [Ermöglichung der Sprecherpartitionierung in Echtzeit-Streams](#)

Aktivieren der Sprecherpartitionierung bei Batch-Transkriptionen

Sie können die Sprecherpartitionierung in einem Batch-Transkriptionsauftrag entweder mit der [StartMedicalTranscriptionJob](#)-API oder mit AWS-Managementkonsole aktivieren. So können Sie den Text in einem Arzt-Patienten-Gespräch nach Sprechern aufteilen und in der Transkriptionsausgabe feststellen, wer was gesagt hat.

AWS-Managementkonsole

AWS-Managementkonsole Um die Diarisierung von Sprechern in Ihrem Transkriptionsjob zu aktivieren, aktivieren Sie zunächst die Audioidentifikation und anschließend die Sprecherpartitionierung.

1. Melden Sie sich an der [AWS-Managementkonsole](#) an.
2. Wählen Sie im Navigationsbereich unter Amazon Transcribe Medizinisch die Option Transkriptionsaufträge aus.
3. Wählen Sie Job erstellen aus.
4. Geben Sie auf der Seite Auftragsdetails angeben Informationen zu Ihrem Transkriptionsauftrag an.
5. Wählen Sie Weiter aus.
6. Aktivieren Sie die Audio-Identifikation.

7. Wählen Sie als Audio-Identifikationstyp die Option Sprecherpartitionierung.
8. Geben Sie bei Maximale Anzahl der Sprecher die maximale Anzahl der Sprecher ein, die Ihrer Meinung nach in Ihrer Audiodatei sprechen.
9. Wählen Sie Erstellen aus.

API

So aktivieren Sie die Sprecherpartitionierung in einem Batch-Transkriptionsauftrag (API)

- Für die [StartMedicalTranscriptionJob](#)-API geben Sie Folgendes an.
 - a. Geben Sie unter `MedicalTranscriptionJobName` einen Namen an, der in Ihrem AWS-Konto eindeutig ist.
 - b. Geben Sie unter `LanguageCode` den Sprachcode an, der der Sprache entspricht, die in der Audiodatei gesprochen wird.
 - c. Geben Sie unter `MediaFileUri`-Parameter des Media-Objekts den Namen der Audiodatei an, die Sie transkribieren möchten.
 - d. Unter `Specialty` können Sie das medizinische Fachgebiet des Arztes angeben, der in der Audiodatei spricht.
 - e. Legen Sie für `Type` die Option `CONVERSATION` fest.
 - f. Geben Sie für den Amazon S3 Bucket `anOutputBucketName`, in dem die Transkriptionsergebnisse gespeichert werden sollen.
 - g. Für das `Settings`-Objekt geben Sie Folgendes an.
 - i. `ShowSpeakerLabels – true`.
 - ii. `MaxSpeakerLabels` – Eine ganze Zahl zwischen 2 und 10, um die Anzahl der Sprecher anzugeben, von denen Sie glauben, dass sie in Ihrem Audio sprechen.

Die folgende Anfrage verwendet die AWS SDK für Python (Boto3) , um einen Batch-Transkriptionsjob für einen Patientendialog mit einem Arzt in der Grundversorgung zu starten, wobei die Lautsprecherpartitionierung aktiviert ist.

```
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)

```

Der folgende Beispielcode zeigt die Transkriptionsergebnisse eines Transkriptionsauftrags mit aktivierter Sprecherpartitionierung.

```

{
  "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

Um eine Audiodatei eines Gesprächs zwischen einem Arzt, der in der Grundversorgung tätig ist, und einem Patienten zu transkribieren (AWS CLI)

- Führen Sie folgenden Code aus.

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

Der folgende Code zeigt den Inhalt von `example-start-command.json`.

```
{
  "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
}
}
```

Ermöglichung der Sprecherpartitionierung in Echtzeit-Streams

Verwenden Sie die AWS-Managementkonsole oder eine Streaming-Anfrage, um Sprecher zu partitionieren und ihre Sprache in einem Echtzeit-Stream zu kennzeichnen. Die Sprecherpartitionierung funktioniert am besten bei zwei bis fünf Sprechern in einem Stream. Amazon Transcribe Medical kann zwar mehr als fünf Sprecher in einem Stream partitionieren, aber die Genauigkeit der Partitionen nimmt ab, wenn Sie diese Anzahl überschreiten.

Verwenden Sie die [StartMedicalStreamTranscription](#)API, um eine HTTP/2 Anfrage zu starten. Verwenden Sie eine vorsignierte URI, um eine WebSocket Anfrage zu starten. Der URI enthält die Informationen, die für die Einrichtung der bidirektionalen Kommunikation zwischen Ihrer Anwendung und Amazon Transcribe -Medical erforderlich sind.

Aktivieren der Lautsprecherpartitionierung bei Audio, das in Ihr Mikrofon eingespielt wird (AWS-Managementkonsole)

Sie können den verwenden AWS-Managementkonsole , um einen Echtzeit-Stream eines Gesprächs zwischen Arzt und Patient oder ein Diktat zu starten, das in Echtzeit in Ihr Mikrofon gesprochen wird.

1. Melden Sie sich an der [AWS-Managementkonsole](#) an.
2. Wählen Sie im Navigationsbereich für Amazon Transcribe Medizin die Option Transkription aus. Real-time
3. Wählen Sie unter Typ der Audioeingabe die Art der medizinischen Sprache, die Sie transkribieren möchten.
4. Wählen Sie unter Zusätzliche Einstellungen die Option Sprecherpartitionierung.
5. Wählen Sie Streaming starten , um mit der Transkription Ihres Echtzeit-Audios zu beginnen.
6. Sprechen Sie in das Mikrofon.

Lautsprecherpartitionierung in einem Stream aktivieren HTTP/2

Um die Sprecherpartitionierung in einem HTTP/2 Stream einer medizinischen Konversation zu aktivieren, verwenden Sie die [StartMedicalStreamTranscription](#) API und geben Sie Folgendes an:

- Geben Sie unter `LanguageCode` den Sprachencode an, der der Sprache im Stream entspricht. Der gültige Wert lautet `en-US`.
- Unter `MediaSampleHertz` geben Sie die Samplerate des Tons an.
- Unter `Specialty` das medizinische Fachgebiet des Arztes anzugeben.
- `ShowSpeakerLabel` – `true`

Weitere Informationen zum Einrichten eines HTTP/2 Streams zur Transkription einer medizinischen Konversation finden Sie unter. [Einen HTTP/2 Stream einrichten](#)

Aktivieren der Lautsprecherpartitionierung in einer Anfrage WebSocket

Um Lautsprecher mit der API in WebSocket Streams zu partitionieren, verwenden Sie das folgende Format, um eine vorsignierte URI zu erstellen, um eine WebSocket Anfrage zu starten, und setzen Sie `show-speaker-label` auf `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
```

Der folgende Code zeigt die verkürzte Beispiellantwort auf eine Streaming-Anfrage.

```
{
  "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": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
    "StartTime": 18.74
}

```

```
]
}
}
```

Amazon Transcribe Medical unterbricht Ihren eingehenden Audiostream auf der Grundlage natürlicher Sprachsegmente, z. B. eines Sprecherwechsels oder einer Audiopause. Die Transkription wird progressiv an Ihre Anwendung zurückgegeben, wobei jede Antwort mehr transkribierte Sprachausgabe enthält, bis das gesamte Segment transkribiert ist. Der vorstehende Code ist ein verkürztes Beispiel für ein vollständig transkribiertes Sprachsegment. Sprecherbeschriftungen erscheinen nur für vollständig transkribierte Segmente.

Die folgende Liste zeigt die Organisation der Objekte und Parameter in einer Streaming-Transkriptionsausgabe.

Transcript

Jedes Sprachsegment hat sein eigenes Transcript-Objekt.

Results

Jedes Transcript-Objekt hat sein eigenes Results-Objekt. Dieses Objekt enthält das Feld `isPartial`. Wenn der Wert `false` ist, werden die Ergebnisse für ein ganzes Sprachsegment zurückgegeben.

Alternatives

Zu jedem Results-Objekt gehört ein Alternatives-Objekt.

Items

Jedes Alternatives-Objekt hat sein eigenes Items-Objekt, das Informationen über jedes Wort und Satzzeichen in der Transkriptionsausgabe enthält. Wenn Sie die Sprecherpartitionierung aktivieren, hat jedes Wort eine `Speaker` Bezeichnung für vollständig transkribierte Sprachsegmente. Amazon Transcribe Medical verwendet diese Bezeichnung, um jedem Sprecher im Stream eine eindeutige Ganzzahl zuzuweisen. Der Parameter `Type` mit dem Wert `speaker-change` zeigt an, dass eine Person aufgehört hat zu sprechen und dass eine andere Person beginnen wird.

Transcript

Jedes Item-Objekt enthält ein transkribiertes Sprachsegment als Wert des Feldes `Transcript`.

Weitere Informationen zu WebSocket Anfragen finden Sie unter [Einen WebSocket Stream einrichten](#).

Transkribieren von Mehrkanal-Audio

Wenn Sie eine Audiodatei oder einen Stream mit mehreren Kanälen haben, können Sie die Kanalidentifikation verwenden, um die Sprache von jedem dieser Kanäle zu transkribieren. Amazon Transcribe Medical transkribiert die Sprache von jedem Kanal separat. Es kombiniert die separaten Transkriptionen der einzelnen Kanäle zu einer einzigen Transkriptionsausgabe.

Verwenden Sie die Kanalidentifikation, um die einzelnen Kanäle in Ihrem Audiomaterial zu identifizieren und die Sprache von jedem dieser Kanäle zu transkribieren. Aktivieren Sie dies in Situationen wie einem Anrufer- und Kundendienstmitarbeiterszenario. Verwenden Sie dies, um einen Anrufer von einem Kundendienstmitarbeiter in Aufzeichnungen oder Streams von Contact Centern zu unterscheiden, die eine Überwachung der Arzneimittelsicherheit durchführen.

Sie können die Kanalidentifizierung sowohl für die Stapelverarbeitung als auch für das Echtzeit-Streaming aktivieren. In der folgenden Liste wird beschrieben, wie man sie für jede Methode aktiviert.

- Batch-Transkription — AWS-Managementkonsole und API [StartMedicalTranscriptionJob](#)
- Streaming-Transkription — WebSocket Streaming und API [StartMedicalStreamTranscription](#)

Transkribieren von Mehrkanal-Audiodateien

Wenn Sie eine Audiodatei transkribieren, gibt Amazon Transcribe Medical eine Liste mit Elementen für jeden Kanal zurück. Ein Element ist ein transkribiertes Wort oder Interpunktionszeichen. Jedes Wort hat eine Startzeit und eine Endzeit. Wenn eine Person auf einem Kanal mit einer Person auf einem anderen Kanal spricht, überschneiden sich die Start- und Endzeiten der Elemente für jeden Kanal, während die Personen übereinander sprechen.

Standardmäßig können Sie Audiodateien mit zwei Kanälen transkribieren. Sie können eine Erhöhung des Quotas beantragen, wenn Sie Dateien mit mehr als zwei Kanälen transkribieren müssen. Informationen zur Beantragung einer Quotaerhöhung finden Sie unter [AWS-Service -Quotas](#).

Um Mehrkanal-Audio in einem Batch-Transkriptionsauftrag zu transkribieren, verwenden Sie die oder die API. AWS-Managementkonsole [StartMedicalTranscriptionJob](#)

AWS-Managementkonsole

AWS-Managementkonsole Um die Kanalidentifikation in Ihrem Batch-Transkriptionsauftrag zu aktivieren, aktivieren Sie die Audioidentifikation und anschließend die Kanalidentifikation. Die Kanalidentifikation ist eine Untergruppe der Audioidentifikation in der AWS-Managementkonsole

1. Melden Sie sich an der [AWS-Managementkonsole](#) an.
2. Wählen Sie im Navigationsbereich unter Amazon Transcribe Medizinisch die Option Transkriptionsjobs aus.
3. Wählen Sie Job erstellen aus.
4. Geben Sie auf der Seite Auftragsdetails angeben Informationen zu Ihrem Transkriptionsauftrag an.
5. Wählen Sie Weiter aus.
6. Aktivieren Sie die Audio-Identifikation.
7. Wählen Sie als Art der Audioidentifikation Kanalidentifikation.
8. Wählen Sie Erstellen aus.

API

So transkribieren Sie eine Mehrkanal-Audiodatei (API)

- Für die [StartMedicalTranscriptionJob](#)-API geben Sie Folgendes an.
 - a. Geben Sie für `TranscriptionJobName` einen eindeutigen Namen für AWS-Konto an.
 - b. Geben Sie unter `LanguageCode` den Sprachcode an, der der Sprache entspricht, die in der Audiodatei gesprochen wird. Der gültige Wert lautet `en-US`.
 - c. Geben Sie unter `MediaFileUri`-Parameter des Media-Objekts den Namen der Mediendatei an, die Sie transkribieren möchten.
 - d. Für das `Settings`-Objekt setzen Sie `ChannelIdentification` auf `true`.

Nachfolgend ein Beispiel für eine Anfrage unter Verwendung von AWS SDK für Python (Boto3).

```
from __future__ import print_function
import time
import boto3
```

```
transcribe = boto3.client('transcribe', 'us-west-2')
job_name = "my-first-transcription-job"
job_name = "my-first-med-transcription-job"
job_uri = "s3://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

Um eine Mehrkanal-Audiodatei mithilfe eines Batch-Transkriptionsauftrags zu transkribieren (AWS CLI)

- Führen Sie folgenden Code aus.

```
aws transcribe start-medical-transcription-job \
--region us-west-2 \
--cli-input-json file://example-start-command.json
```

Im Folgenden finden Sie den Code von `example-start-command.json`.

```
{
  "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
  }
}
```

Der folgende Code zeigt die Transkriptionsausgabe für eine Audiodatei, die ein Gespräch auf zwei Kanälen enthält.

```
{
  "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"
    },
    ...
  ]
}

```

Transkribieren von Mehrkanal-Audiostreams

Mithilfe der API können Sie Audio von separaten Kanälen in Streams oder HTTP/2 Streams transkribieren. WebSocket [StartMedicalStreamTranscription](#)

Standardmäßig können Sie Streams mit zwei Kanälen transkribieren. Sie können eine Erhöhung des Quotas beantragen, wenn Sie Streams mit mehr als zwei Kanälen transkribieren müssen. Informationen zur Beantragung einer Quotaerhöhung finden Sie unter [AWS -Service Quotas](#).

Transkribieren von Mehrkanal-Audio in einem Stream HTTP/2

Um Mehrkanal-Audio in einem HTTP/2 Stream zu transkribieren, verwenden Sie die [StartMedicalStreamTranscription](#) API und geben Sie Folgendes an:

- **LanguageCode** – Der Sprachcode des Audios. Der gültige Wert lautet en-US.

- `MediaEncoding` – Die Kodierung des Audios. Gültige Werte sind `ogg-opus`, `flac` und `pcm`.
- `EnableChannelIdentification` – `true`
- `NumberOfChannels` – Die Anzahl der Kanäle in Ihrem Audiostreaming.

Weitere Informationen zum Einrichten eines HTTP/2 Streams zur Transkription einer medizinischen Konversation finden Sie unter [Einen HTTP/2 Stream einrichten](#)

Transkribieren von Mehrkanal-Audio in einem Stream WebSocket

Um Lautsprecher in WebSocket Streams zu partitionieren, verwenden Sie das folgende Format, um eine vorsignierte URI zu erstellen und eine Anfrage zu starten. WebSocket Geben Sie `enable-channel-identification` als `true` und die Anzahl der Kanäle in Ihrem Stream in `number-of-channels` an. Eine vorsignierte URI enthält die Informationen, die für die Einrichtung einer bidirektionalen Kommunikation zwischen Ihrer Anwendung und Medical erforderlich sind. Amazon Transcribe

```
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
```

Parameterdefinitionen finden Sie in der [API-Referenz](#). Parameter, die allen AWS API-Operationen gemeinsam sind, sind im Abschnitt [Allgemeine Parameter](#) aufgeführt.

Weitere Informationen zu WebSocket Anfragen finden Sie unter [Einen WebSocket Stream einrichten](#).

Multi-channel Streaming-Ausgabe

Die Ausgabe einer Streaming-Transkription ist für HTTP/2 und für WebSocket Anfragen identisch. Im Folgenden finden Sie eine Beispielausgabe.

```
{
  "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"
}
```

Für jedes Sprachsegment gibt es ein `channelId`-Kennzeichen, das anzeigt, zu welchem Kanal die Sprache gehört.

Transkribieren eines medizinischen Diktats

Sie können Amazon Transcribe Medical verwenden, um vom Arzt diktierte medizinische Notizen zu transkribieren, indem Sie entweder einen Batch-Transkriptionsjob oder einen Echtzeit-Stream

verwenden. Mit Batch-Transkriptionsaufträgen können Sie Audiodateien transkribieren. Sie geben in Ihrem Transkriptionsjob oder Stream das medizinische Fachgebiet des Arztes an, um sicherzustellen, dass Amazon Transcribe Medical die Transkriptionsergebnisse mit der höchstmöglichen Genauigkeit liefert.

Sie können ein medizinisches Diktat in den folgenden Fachgebieten transkribieren:

- Kardiologie – nur in Streaming-Transkription verfügbar
- Neurologie – nur in Streaming-Transkription verfügbar
- Onkologie – nur in Streaming-Transkription verfügbar
- Grundversorgung (Allgemeinmedizin/Hausarzt) – umfasst die folgenden Arten von Arztpraxen:
 - Allgemeinmedizin
 - Innere Medizin
 - Geburtshilfe und Gynäkologie () OB-GYN
 - Pädiatrie
- Radiologie – nur in Streaming-Transkription verfügbar
- Urologie – nur in Streaming-Transkription verfügbar

Sie können die Transkriptionsgenauigkeit durch die Verwendung von benutzerdefinierten Vokabularen verbessern. Informationen über die Funktionsweise von medizinischen benutzerdefinierten Vokabularen finden Sie unter [Verbesserung der Transkriptionsgenauigkeit mit benutzerdefinierten medizinischen Vokabularen](#).

Standardmäßig gibt Amazon Transcribe Medical die Transkription mit der höchsten Konfidenzstufe zurück. Wenn Sie es so konfigurieren möchten, dass alternative Transkriptionen zurückgegeben werden, siehe [Erstellen alternativer Transkriptionen](#).

Informationen darüber, wie Zahlen und medizinische Maße in der Transkriptionsausgabe erscheinen, finden Sie unter [Transkription von Zahlen](#) und [Transkribieren von medizinischen Begriffen und Messungen](#).

Themen

- [Transkribieren einer Audiodatei eines medizinischen Diktats](#)
- [Transkribieren eines medizinischen Diktats in einem Echtzeit-Stream](#)

Transkribieren einer Audiodatei eines medizinischen Diktats

Verwenden Sie einen Batch-Transkriptionsauftrag, um Audiodateien von medizinischen Gesprächen zu transkribieren. Damit können Sie einen Arzt-Patienten-Dialog transkribieren. Sie können einen Batch-Transkriptionsauftrag entweder über die [StartMedicalTranscriptionJob](#)-API oder AWS-Managementkonsole starten.

Wenn Sie einen medizinischen Transkriptionsauftrag mit der [StartMedicalTranscriptionJob](#)-API starten, geben Sie PRIMARYCARE als Wert für den Parameter Specialty an.

AWS-Managementkonsole

Um einen Dialog zwischen Arzt und Patient zu transkribieren (AWS-Managementkonsole)

Um den zur Transkription eines Gesprächs zwischen Arzt und Patient AWS-Managementkonsole zu verwenden, erstellen Sie einen Transkriptionsauftrag und wählen Sie als Audioeingabetyp „Konversation“ aus.

1. Melden Sie sich an der [AWS-Managementkonsole](#) an.
2. Wählen Sie im Navigationsbereich unter Amazon Transcribe Medizinisch die Option Transkriptionsaufträge aus.
3. Wählen Sie Job erstellen aus.
4. Geben Sie auf der Seite Auftragsdetails an unter Auftragseinstellungen Folgendes an.
 - a. Name – der Name des Transkriptionsauftrags.
 - b. Audio-Eingangstyp – Diktat
5. Geben Sie für die übrigen Felder den Amazon S3 Speicherort Ihrer Audiodatei an und geben Sie an, wo Sie die Ausgabe Ihres Transkriptionsauftrags speichern möchten.
6. Wählen Sie Next.
7. Wählen Sie Erstellen aus.

API

So transkribieren Sie ein medizinisches Gespräch mit einem Batch-Transkriptionsauftrag (API)

- Für die [StartMedicalTranscriptionJob](#)-API geben Sie Folgendes an.

- a. Geben Sie unter `MedicalTranscriptionJobName` einen Namen an, der in AWS-Konto eindeutig ist.
- b. Geben Sie unter `LanguageCode` den Sprachcode an, der der in Ihrer Audiodatei gesprochenen Sprache entspricht, sowie die Sprache Ihres Wortschatzfilters.
- c. Geben Sie unter `MediaFileUri`-Parameter des `Media`-Objekts den Namen der Audiodatei an, die Sie transkribieren möchten.
- d. Unter `Specialty` können Sie das medizinische Fachgebiet des Arztes angeben, der in der Audiodatei spricht.
- e. Legen Sie für `Type` die Option `DICTATION` fest.
- f. Unter `OutputBucketName` geben Sie den Amazon S3 -Bucket an, in dem die Transkriptionsergebnisse gespeichert werden.

Im Folgenden finden Sie eine Beispielanforderung, mit der AWS SDK für Python (Boto3) ein ärztliches Diktat eines Facharztes transkribiert wird. `PRIMARYCARE`

```
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)
```

Der folgende Beispielcode zeigt die Transkriptionsergebnisse eines medizinischen Diktats.

```
{
  "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

Um die Lautsprecherpartitionierung in einem Batch-Transkriptionsauftrag zu aktivieren (AWS CLI)

- Führen Sie folgenden Code aus.

```
aws transcribe start-medical-transcription-job \  
--region us-west-2 \  
--cli-input-json file://example-start-command.json
```

Der folgende Code zeigt den Inhalt von `example-start-command.json`.

```
{  
  "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"  
}
```

Transkribieren eines medizinischen Diktats in einem Echtzeit-Stream

Verwenden Sie einen WebSocket Stream, um ein medizinisches Diktat als Audiostream zu transkribieren. Sie können den auch verwenden AWS-Managementkonsole , um Sprache, die Sie oder andere sprechen, direkt in ein Mikrofon zu transkribieren.

Für einen Stream HTTP/2 oder einen WebSocket Stream können Sie Audio in den folgenden medizinischen Fachgebieten transkribieren:

- Kardiologie
- Onkologie
- Neurologie
- Grundversorgung
- Radiologie
- Urologie

Jedes medizinische Fachgebiet umfasst viele Arten von Verfahren und Terminen. Ärzte diktieren daher viele verschiedene Arten von Notizen. Verwenden Sie die folgenden Beispiele als Leitfaden, um den Wert des `specialty` URI-Parameters der WebSocket Anfrage oder des `Specialty` API-Parameters anzugeben: [StartMedicalStreamTranscription](#)

- Für ein Diktat nach einem elektrophysiologischen oder echokardiographischen Eingriff wählen Sie `CARDIOLOGY`.
- Für ein Diktat nach einem chirurgisch-onkologischen oder radio-onkologischen Eingriff wählen Sie `ONCOLOGY`.
- Für einen Arzt, der Notizen diktiert, die eine Diagnose von Enzephalitis angeben, wählen Sie `NEUROLOGY`.
- Für ein Diktat von Behandlungsnotizen zur Zertrümmerung eines Blasensteins, wählen Sie `UROLOGY`.
- Für ein Diktat von Arztnotizen nach einer internistischen Untersuchung wählen Sie `PRIMARYCARE`.
- Für ein Diktat eines Arztes, der die Ergebnisse eines CT-Scans, PET-Scans, MRTs oder einer Röntgenaufnahme mitteilt, wählen Sie `RADIOLOGY`.
- Für ein Diktat von Arztnotizen nach einer gynäkologischen Untersuchung wählen Sie `PRIMARYCARE`.

Um die Transkriptionsgenauigkeit bestimmter Begriffe in einem Echtzeit-Stream zu verbessern, verwenden Sie ein benutzerdefiniertes Vokabular. Um ein benutzerdefiniertes Vokabular zu aktivieren, setzen Sie den Wert von `vocabulary-name` auf den Namen des benutzerdefinierten Vokabulars, das Sie verwenden möchten.

Transkribieren Sie ein in Ihr Mikrofon gesprochenes Diktat mit der AWS-Managementkonsole

Um das Streaming-Audio eines medizinischen Diktats mit der AWS-Managementkonsole zu transkribieren, wählen Sie die Option zum Transkribieren eines medizinischen Diktats, starten Sie den Stream und beginnen Sie, in das Mikrofon zu sprechen.

Um Streaming-Audio eines medizinischen Diktats zu transkribieren (AWS-Managementkonsole)

1. Melden Sie sich an der [AWS-Managementkonsole](#) an.
2. Wählen Sie im Navigationsbereich unter Amazon Transcribe Medizinisch die Option Transkription aus. Real-time
3. Wählen Sie Diktat.
4. Wählen Sie unter Medizinisches Fachgebiet das medizinische Fachgebiet des Arztes, der im Stream spricht.
5. Wählen Sie dann Start Streaming (Streamen starten).
6. Sprechen Sie in das Mikrofon.

Ein Diktat in einem Stream transkribieren HTTP/2

Um einen HTTP/2 Stream eines medizinischen Diktats zu transkribieren, verwenden Sie die [StartMedicalStreamTranscription](#) API und geben Sie Folgendes an:

- `LanguageCode` – Der Sprachcode. Der gültige Wert ist `en-US`
- `MediaEncoding` – Die für das Eingangsaudio verwendete Kodierung. Gültige Werte sind `pcm`, `ogg-opus` und `flac`.
- `Specialty` – Das Fachgebiet der medizinischen Fachkraft.
- `Type` – `DICTATION`

Weitere Informationen zum Einrichten eines HTTP/2 Streams zum Transkribieren eines medizinischen Diktats finden Sie unter [Einen HTTP/2 Stream einrichten](#)

Verwenden einer WebSocket Streaming-Anfrage zum Transkribieren eines medizinischen Diktats

Um ein medizinisches Diktat mithilfe einer WebSocket Anfrage in einem Echtzeit-Stream zu transkribieren, erstellen Sie eine vorsignierte URI. Diese URI enthält die Informationen, die für die Einrichtung des Audiostreams zwischen Ihrer Anwendung und Medical erforderlich sind. Amazon

Transcribe Weitere Informationen zum Erstellen von WebSocket Anfragen finden Sie unter [Einen WebSocket Stream einrichten](#).

Verwenden Sie die folgende Vorlage, um Ihren vordefinierten URI zu erstellen.

```
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
```

Weitere Informationen zur Erstellung von vorsignierten URIs finden Sie unter [Einen WebSocket Stream einrichten](#).

Verbesserung der Transkriptionsgenauigkeit mit benutzerdefinierten medizinischen Vokabularen

Um die Genauigkeit der Transkription in Amazon Transcribe Medical zu verbessern, erstellen und verwenden Sie ein oder mehrere benutzerdefinierte medizinische Vokabeln. Ein benutzerdefiniertes Vokabular ist eine Sammlung von Wörtern oder Ausdrücken, die domänenspezifisch sind. Diese Sammlung trägt dazu bei, die Leistung von Amazon Transcribe Medical bei der Transkription dieser Wörter oder Ausdrücke zu verbessern.

Sie sind für die Integrität Ihrer eigenen Daten verantwortlich, wenn Sie Medical verwenden Amazon Transcribe . Geben Sie keine vertraulichen Daten, personenbezogene Daten (PII) oder geschützte Gesundheitsinformationen (PHI) in ein benutzerdefiniertes Vokabular ein.

Die besten Ergebnisse erzielen Sie, wenn Sie separate kleine benutzerdefinierte Vokabulare erstellen, die jeweils zur Transkription einer bestimmten Audioaufnahme beitragen. Sie erhalten eine höhere Transkriptionsgenauigkeit, als wenn Sie ein großes benutzerdefiniertes Vokabular für alle Ihre Aufnahmen erstellen würden.

Standardmäßig können Sie bis zu 100 benutzerdefinierte Vokabulare in Ihrem AWS-Konto haben. Ein benutzerdefiniertes Vokabular darf nicht mehr als 50 KB groß sein. Informationen dazu, wie Sie eine Erhöhung der Anzahl der benutzerdefinierten Vokabulare beantragen können, finden Sie unter [AWS-Konto AWS Servicekontingente](#).

Benutzerdefinierte Vokabulare sind in US-Englisch (en-US) verfügbar.

Themen

- [Erstellen einer Textdatei für Ihr benutzerdefiniertes medizinisches Vokabular](#)
- [Verwendung einer Textdatei zur Erstellung eines benutzerdefinierten medizinischen Vokabulars](#)
- [Transkribieren einer Audiodatei unter Verwenden eines benutzerdefinierten medizinischen Vokabulars](#)
- [Transkription eines Echtzeit-Streams unter Verwenden eines benutzerdefinierten medizinischen Vokabulars](#)
- [Zeichensatz für Amazon Transcribe Medical](#)

Erstellen einer Textdatei für Ihr benutzerdefiniertes medizinisches Vokabular

Um ein benutzerdefiniertes Vokabular zu erstellen, erstellen Sie eine Textdatei im UTF-8 Format. In dieser Datei erstellen Sie eine vierspaltige Tabelle, wobei jede Spalte ein Feld angibt. Jedes Feld teilt Amazon Transcribe Medical mit, wie die domänenspezifischen Begriffe ausgesprochen werden oder wie diese Begriffe in Ihren Transkriptionen angezeigt werden sollen. Sie speichern die Textdatei, die diese Felder enthält, in einem Amazon S3 -Bucket.

Verstehen, wie man eine Textdatei formatiert

Um ein medizinisches benutzerdefiniertes Vokabular zu erstellen, geben Sie die Spaltennamen in einem Header ein. Sie geben die Werte für jede Spalte unterhalb des Headers ein.

Im Folgenden sind die Namen der vier Spalten der Tabelle aufgeführt:

- `Phrase` – Spalte erforderlich, Werte erforderlich

- IPA – Spalte erforderlich, Werte können optional sein
- SoundsLike – Spalte erforderlich, Werte können optional sein
- DisplayAs – Spalte erforderlich, Werte können optional sein

Wenn Sie ein benutzerdefiniertes Vokabular erstellen, stellen Sie sicher, dass Sie:

- Trennen Sie jede Spalte durch ein einzelnes Tabulatorzeichen. Amazon Transcribe gibt eine Fehlermeldung aus, wenn Sie versuchen, die Spalten durch Leerzeichen oder mehrere Tabulatorzeichen zu trennen.
- Vergewissern Sie sich, dass nach jedem Wert in einer Spalte keine Leerzeichen oder Leerstellen folgen.

Achten Sie darauf, dass die Werte, die Sie für jede Spalte eingeben:

- Weniger als 256 Zeichen haben, einschließlich Bindestriche
- Verwenden Sie nur Zeichen aus dem zulässigen Zeichensatz, siehe [Zeichensatz für Amazon Transcribe Medical](#).

Eingabe von Werten für die Spalten der Tabelle

Die folgenden Informationen zeigen Ihnen, wie Sie Werte für die vier Spalten der Tabelle angeben können:

- **Phrase** – Das Wort oder die Formulierung, die erkannt werden soll. Sie müssen in dieser Spalte Werte eingeben.

Trennen Sie die Wörter durch einen Bindestrich (-) voneinander, wenn es sich bei dem Eintrag um einen Ausdruck handelt. Geben Sie beispielsweise **cerebral autosomal dominant arteriopathy with subcortical infarcts and leukoencephalopathy** als **cerebral-autosomal-dominant-arteriopathy-with-subcortical-infarcts-and-leukoencephalopathy** ein.

Geben Sie Akronyme oder andere Wörter, deren Buchstaben einzeln ausgesprochen werden sollen, als einzelne Buchstaben gefolgt von Punkten ein, z. B. **D.N.A.** oder **S.T.E.M.I.**. Um die Pluralform eines Akronyms, z. B. „STEMIs“, einzugeben, trennen Sie das „s“ vom Akronym mit einem Bindestrich: **„S.T.E.M.I-s“** Sie können für Akronyme entweder Groß- oder Kleinbuchstaben verwenden.

Die Spalte `Phrase` ist obligatorisch. Sie können jedes zulässige Zeichen für die Eingabesprache verwenden. Zulässige Zeichen finden Sie unter [Zeichensatz für Amazon Transcribe Medical](#). Wenn Sie die `DisplayAs` Spalte nicht angeben, verwendet Amazon Transcribe Medical den Inhalt der `Phrase` Spalte in der Ausgabedatei.

- **IPA** (Spalte erforderlich, Werte können optional sein) – Um die Aussprache eines Wortes oder Satzes anzugeben, können Sie in dieser Spalte Zeichen des [Internationalen Phonetischen Alphabets \(IPA\)](#) angeben. Die Spalte IPA darf keine Leerzeichen zu Beginn oder am Ende enthalten und Sie müssen ein einzelnes Leerzeichen verwenden, um jedes Phonem in der Eingabe voneinander zu trennen. Auf Englisch geben Sie beispielsweise den Ausdruck **acute-respiratory-distress-syndrome** folgendermaßen ein: **ə k j u t # # s p # # ə t # # i d # s t # # s s # n d # o # m**. Geben Sie beispielsweise den Ausdruck **A.L.L.** folgendermaßen ein: **e # # l # l**.

Auch wenn Sie den Inhalt der IPA-Spalte nicht angeben, müssen Sie eine leere IPA-Spalte einfügen. Wenn Sie Werte in die Spalte IPA einfügen, können Sie keine Werte für die Spalte `SoundsLike` angeben.

Eine Liste der zulässigen IPA-Zeichen für eine bestimmte Sprache finden Sie unter [Zeichensatz für Amazon Transcribe Medical](#). US-Englisch ist die einzige Sprache, die in Amazon Transcribe Medical verfügbar ist.

- **SoundsLike** (Spalte erforderlich, Werte können optional sein) – Sie können ein Wort oder einen Satz in kleinere Segmente unterteilen und für jedes Segment eine Aussprache in der Standardorthografie der Sprache angeben, um den Klang des Wortes nachzuahmen. Geben Sie beispielsweise Aussprachehinweise für den Ausdruck **cerebral-autosomal-dominant-arteriopathy-with-subcortical-infarcts-and-leukoencephalopathy** folgendermaßen ein: **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**. Der Hinweis für den Ausdruck **atrioventricular-nodal-reentrant-tachycardia** würde wie folgt aussehen: **ay-tree-o-ven-trick-u-lar-node-al-re-entr-ant-tack-ih-card-ia**. Alle Teile des Hinweises werden durch einen Bindestrich (-) voneinander getrennt.

Selbst wenn Sie keine Werte für die Spalte `SoundsLike` angeben, müssen Sie eine leere Spalte `SoundsLike` einfügen. Wenn Sie Werte in die Spalte `SoundsLike` einfügen, können Sie keine Werte für die Spalte IPA angeben.

Sie können jedes zulässige Zeichen für die Eingabesprache verwenden. Die Liste der zulässigen Zeichen finden Sie unter [Zeichensatz für Amazon Transcribe Medical](#).

- **DisplayAs** (Spalte erforderlich, Werte können optional sein) – Legt fest, wie das Wort oder die Formulierung bei der Ausgabe aussieht. Beispiel: Wenn das Wort oder der Ausdruck **cerebral-autosomal-dominant-arteriopathy-with-subcortical-infarcts-and-leukoencephalopathy** lautet, können Sie die Darstellungsform auf `cerebral autosomal dominant arteriopathy with subcortical infarcts and leukoencephalopathy` festlegen, damit der Bindestrich nicht angezeigt wird. Sie können **DisplayAs** auch als **CADASIL** angeben, wenn Sie das Akronym anstelle des vollständigen Begriffs in der Ausgabe anzeigen möchten.

Wenn Sie die **DisplayAs** Spalte nicht angeben, verwendet Amazon Transcribe Medical die **Phrase** Spalte aus der Eingabedatei in der Ausgabe.

Sie können jedes UTF-8 Zeichen in der **DisplayAs** Spalte verwenden.

Sie können Leerzeichen nur für die Werte in den Spalten **IPA** und **DisplayAs** eingeben.

Um die Textdatei Ihres benutzerdefinierten Vokabulars zu erstellen, setzen Sie jedes Wort oder jeden Satz in Ihrer Textdatei in eine eigene Zeile. Trennen Sie die Spalten durch Tabulatorzeichen. Fügen Sie Leerzeichen nur für Werte in den Spalten **IPA** und **DisplayAs** ein. Speichern Sie die Datei mit der Erweiterung `.txt` in einem Amazon S3 Bucket in derselben AWS-Region Datei, in der Sie Amazon Transcribe Medical verwenden, um Ihr benutzerdefiniertes Vokabular zu erstellen.

Ihre benutzerdefinierte Vokabeldatei unterstützt LF sowohl CRLF Zeilenenden als auch Zeilenenden.

Die folgenden Beispiele zeigen Text, mit dem Sie benutzerdefinierte Vokabulare erstellen können. Um ein benutzerdefiniertes Vokabular aus diesen Beispielen zu erstellen, kopieren Sie ein Beispiel in einen Texteditor, ersetzen Sie `[TAB]` durch ein Tabulatorzeichen und laden Sie die gespeicherte Textdatei auf Amazon S3 hoch.

```
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]
```

Sie können Spalten in beliebiger Reihenfolge eingeben. Die folgenden Beispiele zeigen andere gültige Strukturen für die Eingabedatei des benutzerdefinierten Vokabulars.

```
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
```

Zur besseren Lesbarkeit sind in den folgenden Tabellen die vorhergehenden Beispiele in HTML-Format deutlicher dargestellt. Sie dienen nur zur Veranschaulichung der Beispiele.

Phrase	IPA	SoundsLike	DisplayAs
acute-respiratory-distress-syndrome			acute respiratory distress syndrome
A.L.L.	eɪ ɛ ɛ		ALL
atrioventricular-nodal-reentrant-tachycardia		ay-tree-o-ven-trick-u-lar-node-al-re-entr-ant-tack-ih-card-ia	

Phrase	SoundsLike	IPA	DisplayAs
acute-respiratory-distress-syndrome			acute respiratory distress syndrome
atrioventricular-nodal-reentrant-tachycardia	ay-tree-o-ven-trick-u-lar-node-al-re-entr-ant-tack-ih-card-ia		
A.L.L.		eɪ ɛ ɛ	ALL

DisplayAs	SoundsLike	IPA	Phrase
acute respiratory distress syndrome			acute-respiratory-distress-syndrome
ALL		eɪ ɛ l ɛ l	A.L.L.
	ay-tree-o-ven-trick-ular-node-al-re-entr-ant-tack-ih-card-ia		atrioventricular-nodal-reentrant-tachycardia

Verwendung einer Textdatei zur Erstellung eines benutzerdefinierten medizinischen Vokabulars

Um ein benutzerdefiniertes Vokabular zu erstellen, müssen Sie eine Textdatei vorbereitet haben, die eine Sammlung von Wörtern oder Ausdrücken enthält. Amazon Transcribe Medical verwendet diese Textdatei, um ein benutzerdefiniertes Vokabular zu erstellen, mit dem Sie die Transkriptionsgenauigkeit dieser Wörter oder Ausdrücke verbessern können. Sie können mithilfe der [CreateMedicalVocabulary](#) API oder der Amazon Transcribe Medical Console ein benutzerdefiniertes Vokabular erstellen.

AWS-Managementkonsole

Um das zu verwenden, AWS-Managementkonsole um ein benutzerdefiniertes Vokabular zu erstellen, geben Sie den Amazon S3 URI der Textdatei an, die Ihre Wörter oder Ausdrücke enthält.

1. Melden Sie sich an der [AWS-Managementkonsole](#) an.
2. Wählen Sie im Navigationsbereich unter Amazon Transcribe Medizinisch die Option Benutzerdefiniertes Vokabular aus.
3. Wählen Sie unter Name in den Vokabular-Einstellungen einen Namen für Ihr benutzerdefiniertes Vokabular.
4. Geben Sie den Speicherort Ihrer Audio- oder Videodatei in Amazon S3 an:
 - Geben Sie für den Speicherort der Vokabular-Eingabedatei auf S3 unter Vokabular-Einstellungen den Amazon S3 -URI an, der die Textdatei identifiziert, die Sie zur Erstellung Ihres benutzerdefinierten Vokabulars verwenden werden.

- Wählen Sie für den Speicherort der Vokabulareingabedatei in S3 die Option S3 durchsuchen, um nach der Textdatei zu suchen und sie auszuwählen.

5. Wählen Sie Vokabular erstellen.

Sie können den Bearbeitungsstatus Ihres benutzerdefinierten Vokabulars unter AWS-Managementkonsole einsehen.

API

So erstellen Sie ein benutzerdefiniertes medizinisches Vokabular (API)

- Für die [StartTranscriptionJob](#)-API geben Sie Folgendes an.
 - a. Legen Sie für LanguageCode die Option en-US fest.
 - b. Geben Sie für VocabularyFileUri den Amazon S3 Speicherort der Textdatei an, mit der Sie Ihr benutzerdefiniertes Vokabular definieren.
 - c. Geben Sie unter VocabularyName einen Namen für Ihr benutzerdefiniertes Vokabular an. Der Name, den Sie angeben, muss innerhalb Ihres eindeutig sein AWS-Konto.

Um den Verarbeitungsstatus Ihres benutzerdefinierten Vokabulars zu sehen, verwenden Sie die [GetMedicalVocabulary](#)-API.

Im Folgenden finden Sie eine Beispielanforderung, mit der AWS SDK für Python (Boto3) Sie ein benutzerdefiniertes Vokabular erstellen können.

```
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

Um die Sprecherpartitionierung in einem Batch-Transkriptionsauftrag zu aktivieren (AWS CLI)

- Führen Sie folgenden Code aus.

```
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
```

Transkribieren einer Audiodatei unter Verwenden eines benutzerdefinierten medizinischen Vokabulars

Verwenden Sie das [StartMedicalTranscriptionJob](#) oder das AWS-Managementkonsole, um einen Transkriptionsauftrag zu starten, der ein benutzerdefiniertes Vokabular verwendet, um die Transkriptionsgenauigkeit zu verbessern.

AWS-Managementkonsole

1. Melden Sie sich an der [AWS-Managementkonsole](#) an.
2. Wählen Sie im Navigationsbereich unter Amazon Transcribe Medizinisch die Option Transkriptionsaufträge aus.
3. Wählen Sie Job erstellen aus.
4. Geben Sie auf der Seite Auftragsdetails angeben Informationen zu Ihrem Transkriptionsauftrag an.
5. Wählen Sie Weiter aus.
6. Aktivieren Sie unter Anpassung die Option Benutzerdefiniertes Vokabular.
7. Wählen Sie unter Vokabularauswahl ein benutzerdefiniertes Vokabular aus.

8. Wählen Sie Erstellen aus.

API

So aktivieren Sie die Sprecherpartitionierung in einer Audiodatei mithilfe eines Batch-Transkriptionsauftrags (API)

- Für die [StartMedicalTranscriptionJob](#)-API geben Sie Folgendes an.
 - a. Geben Sie unter `MedicalTranscriptionJobName` einen Namen an, der in Ihrem AWS-Konto eindeutig ist.
 - b. Geben Sie unter `LanguageCode` den Sprachcode an, der der in Ihrer Audiodatei gesprochenen Sprache entspricht, sowie die Sprache Ihres Wortschatzfilters.
 - c. Geben Sie unter `MediaFileUri`-Parameter des Media-Objekts den Namen der Audiodatei an, die Sie transkribieren möchten.
 - d. Unter `Specialty` können Sie das medizinische Fachgebiet des Arztes angeben, der in der Audiodatei spricht.
 - e. Geben Sie unter `Type` an, ob es sich bei der Audiodatei um ein Gespräch oder ein Diktat handelt.
 - f. Unter `OutputBucketName` geben Sie den Amazon S3 -Bucket an, in dem die Transkriptionsergebnisse gespeichert werden.
 - g. Für das `Settings`-Objekt geben Sie Folgendes an.
 - `VocabularyName` – der Name Ihres benutzerdefinierten Vokabulars.

Die folgende Anfrage verwendet den AWS SDK für Python (Boto3) , um einen Batch-Transkriptionsauftrag mit einem benutzerdefinierten Wortschatz zu starten.

```
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)

```

Transkription eines Echtzeit-Streams unter Verwenden eines benutzerdefinierten medizinischen Vokabulars

Um die Transkriptionsgenauigkeit in einem Echtzeit-Stream zu verbessern, können Sie ein benutzerdefiniertes Vokabular verwenden, indem Sie entweder Streams HTTP/2 oder WebSocket Streams verwenden. Verwenden Sie die [StartMedicalStreamTranscriptionAPI](#), um eine HTTP/2 Anfrage zu starten. Sie können ein benutzerdefiniertes Vokabular in Echtzeit verwenden AWS-Managementkonsole, indem Sie entweder die [StartMedicalStreamTranscriptionAPI](#) oder das WebSocket Protokoll verwenden.

Transkribieren eines Diktats, das in Ihr Mikrofon gesprochen wird (AWS-Managementkonsole)

Um das Streaming-Audio eines medizinischen Diktats AWS-Managementkonsole zu transkribieren, wählen Sie die Option zum Transkribieren eines medizinischen Diktats, starten Sie den Stream und beginnen Sie, in das Mikrofon zu sprechen.

Um Streaming-Audio eines medizinischen Diktats zu transkribieren (AWS-Managementkonsole)

1. Melden Sie sich an der [AWS-Managementkonsole](#) an.
2. Wählen Sie im Navigationsbereich unter Amazon Transcribe Medizinisch die Option Transkription aus. Real-time
3. Wählen Sie unter Medizinisches Fachgebiet das medizinische Fachgebiet des Arztes, der im Stream spricht.
4. Wählen Sie als Audioeingangstyp entweder Gespräch oder Diktat.
5. Wählen Sie für Zusätzliche Einstellungen die Option Benutzerdefiniertes Vokabular.
 - Wählen Sie für die Vokabularauswahl das benutzerdefinierte Vokabular.
6. Wählen Sie dann Start Streaming (Streamen starten).
7. Sprechen Sie in das Mikrofon.

Lautsprecherpartitionierung in einem Stream aktivieren HTTP/2

Im Folgenden finden Sie die Syntax für die Parameter einer HTTP/2 Anfrage.

```
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
```

Beschreibung der Parameter:

- **host:** Aktualisiere das AWS-Region ('us-west-2' im vorherigen Beispiel) mit dem, das AWS-Region du anrufst. [Eine Liste der gültigen Endpoints finden Sie AWS-Regionen unter Endpoints AWS-Regionen](#).
- **Berechtigung:** Dies ist ein generiertes Feld. Weitere Informationen zum Erstellen einer Signatur finden Sie unter [Signieren von AWS Anfragen mit Signature Version 4](#).
- **x-amz-ziel:** Ändern Sie dieses Feld nicht; verwenden Sie den Inhalt aus dem vorangegangenen Beispiel.
- **x-amz-content-sha256:** Dies ist ein generiertes Feld. Weitere Informationen zur Berechnung einer Signatur finden Sie unter [Signieren von AWS Anfragen mit Signature Version 4](#).
- **x-amz-date:** Das Datum und die Uhrzeit der Erstellung der Signatur. Das Format ist JJJJMMTTTHHMMSSZ, wobei JJJJ=Jahr, MM=Monat, TT=Tag, HH=Stunde, MM=Minute, SS=Sekunden und „T“ und „Z“ feste Zeichen sind. Weitere Informationen finden Sie unter [Handhabung von Daten in Signature Version 4](#).
- **x-amzn-transcribe-session-id:** Der Name für Ihre Streaming-Sitzung.
- **x-amzn-transcribe-language-code:** Die Kodierung, die für Ihre Audioeingabe verwendet wird. Eine Liste gültiger Werte finden Sie unter [StartMedicalStreamTranscription](#) oder [Unterstützte Sprachen und sprachspezifische Funktionen](#).
- **x-amzn-transcribe-media-encoding:** Die Kodierung, die für Ihre Audioeingabe verwendet wird. Gültige Werte sind pcm, ogg-opus und flac.
- **x-amzn-transcribe-sample-rate:** Die Samplerate des Eingangs-Audios (in Hertz). Amazon Transcribe unterstützt einen Bereich von 8.000 Hz bis 48.000 Hz. Low-quality Audio, z. B. Telefonaudio, hat normalerweise eine Frequenz von etwa 8.000 Hz. High-quality Audio reicht typischerweise von 16.000 Hz bis 48.000 Hz. Beachten Sie, dass die von Ihnen angegebene Samplerate mit der Ihres Audios übereinstimmen muss.
- **x-amzn-transcribe-vocabulary-name:** Der Name des Vokabulars, das Sie für Ihre Transkription verwenden möchten.
- **x-amzn-transcribe-specialty:** Das zu transkribierende medizinische Fachgebiet.
- **x-amzn-transcribe-type:** Wählen Sie, ob es sich um ein Diktat oder ein Gespräch handelt.
- **x-amzn-transcribe-show-speaker-label:** Um die Diarisierung zu aktivieren, muss dieser Wert true sein.
- **Inhaltstyp:** Ändern Sie dieses Feld nicht; verwenden Sie den Inhalt aus dem vorangegangenen Beispiel.

Aktivieren der Lautsprecherpartitionierung in einer Anfrage WebSocket

Um Lautsprecher mit der API in WebSocket Streams zu partitionieren, verwenden Sie das folgende Format, um einen vorsignierten URI zu erstellen, um eine WebSocket Anfrage zu starten, und setzen Sie ihn `vocabulary-name` auf den Namen des benutzerdefinierten Vokabulars.

```
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
```

Zeichensatz für Amazon Transcribe Medical

Verwenden Sie den folgenden Zeichensatz, um benutzerdefinierte Vokabulare in Amazon Transcribe Medical zu verwenden.

Englischer Zeichensatz

Für ein englisches benutzerdefiniertes Vokabular sind folgende Zeichen in den Spalten `Phrase` und `SoundsLike` zulässig:

- a-z
- A-Z
- ' (Apostroph)
- - (Bindestrich)
- . (Punkt)

Sie können die folgenden Zeichen des internationalen phonetischen Alphabets (IPA) in der Spalte IPA der Vokabulareingabedatei verwenden.

Zeichen	Code	Zeichen	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
ɫ	006C 0329	ɪ	026A
m	006D	ɹ	0279
n	006E	ʃ	0283
ŋ	006E 0329	ʊ	028A
ou	006F 028A	ʌ	028C
p	0070	ʌ	028D
s	0073	ʒ	0292

Zeichen	Code	Zeichen	Code
t	0074	dʒ	02A4
u	0075	tʃ	02A7
v	0076	θ	03B8

Identifizierung persönlicher Gesundheitsinformationen (Personal Health Information, PHI) in einer Transkription

Verwenden Sie die Identifizierung persönlicher Gesundheitsinformationen, um persönliche Gesundheitsinformationen (PHI) in Ihren Transkriptionsergebnissen zu kennzeichnen. Durch die Überprüfung der Kennzeichnungen können Sie PHI finden, die zur Identifizierung eines Patienten verwendet werden könnten.

Sie können PHI entweder mit einem Echtzeit-Stream oder einem Batch-Transkriptionsauftrag identifizieren.

Sie können Ihre eigene Nachbearbeitung verwenden, um die in der Transkriptionsausgabe identifizierten PHI zu schwärzen.

Verwenden Sie die Identifizierung persönlicher Gesundheitsinformationen, um die folgenden Arten von PHI zu identifizieren:

- Persönliche PHI:
 - Namen – vollständiger Name oder Nachname und Initialen
 - Gender
 - Age
 - Telefonnummern
 - Daten (ohne das Jahr), die sich direkt auf den Patienten beziehen
 - E-Mail-Adressen
- Geografische PHI:
 - Physische Adresse
 - Postleitzahl
 - Name des medizinischen Zentrums oder der Praxis

- Konto-PHI:
 - Fax-Nummern
 - Sozialversicherungsnummern (Social Security Numbers, SSNs)
 - Nummern der Leistungsempfänger in der Krankenversicherung
 - Kontonummern
 - Zertifikats- oder Lizenznummern
- Fahrzeug-PHI:
 - Fahrgestellnummern (VIN)
 - Kfz-Kennzeichen
- Sonstige PHI:
 - Web Uniform Resource Location (URL)
 - Internet-Protokoll (IP)-Adressnummern

Amazon Transcribe Medical ist eine nach dem Health Insurance Portability and Accountability Act von 1996 (HIPAA) zugelassene Dienstleistung. Weitere Informationen finden Sie unter [Amazon Transcribe Medical](#). Informationen zur Identifizierung von PHI in einer Audiodatei finden Sie unter [Identifizierung von PHI in einer Audiodatei](#). Informationen zur Identifizierung von PHI in einem Stream finden Sie unter [Identifizierung von PHI in einem Echtzeit-Stream](#).

Themen

- [Identifizierung von PHI in einer Audiodatei](#)
- [Identifizierung von PHI in einem Echtzeit-Stream](#)

Identifizierung von PHI in einer Audiodatei

Verwenden Sie einen Batch-Transkriptionsauftrag, um Audiodateien zu transkribieren und die darin enthaltenen persönlichen Gesundheitsinformationen (PHI) zu identifizieren. Wenn Sie die Identifizierung persönlicher Gesundheitsinformationen (PHI) aktivieren, kennzeichnet Amazon Transcribe Medical das PHI, das es in den Transkriptionsergebnissen identifiziert hat. Informationen zu den PHI, die Amazon Transcribe Medical identifizieren kann, finden Sie unter [Identifizierung persönlicher Gesundheitsinformationen \(Personal Health Information, PHI\) in einer Transkription](#).

Sie können einen Batch-Transkriptionsauftrag entweder über die [StartMedicalTranscriptionJob](#)-API oder AWS-Managementkonsole starten.

AWS-Managementkonsole

AWS-Managementkonsole Um den zum Transkribieren eines Dialogs zwischen Arzt und Patient zu verwenden, erstellen Sie einen Transkriptionsauftrag und wählen Sie als Audioeingabetyp „Konversation“.

Um eine Audiodatei zu transkribieren und ihre PHI zu identifizieren (AWS-Managementkonsole)

1. Melden Sie sich an der [AWS-Managementkonsole](#) an.
2. Wählen Sie im Navigationsbereich unter Amazon Transcribe Medizinisch die Option Transkriptionsjobs aus.
3. Wählen Sie Job erstellen aus.
4. Geben Sie auf der Seite Auftragsdetails an, was unter Auftragseinstellungen Folgendes an.
 - a. Name — Der Name des Transkriptionsauftrags, der für Sie einzigartig ist. AWS-Konto
 - b. Audioeingangstyp – Gespräch oder Diktat.
5. Geben Sie für die übrigen Felder den Amazon S3 Speicherort Ihrer Audiodatei an und geben Sie an, wo Sie die Ausgabe Ihres Transkriptionsauftrags speichern möchten.
6. Wählen Sie Weiter aus.
7. Wählen Sie unter Audioeinstellungen die Option PHI-Identifikation.
8. Wählen Sie Erstellen aus.

API

Transkribieren einer Audiodatei und Identifizierung des PHI mit einem Batch-Transkriptionsauftrag (API)

- Für die [StartMedicalTranscriptionJob](#)-API geben Sie Folgendes an.
 - a. Geben Sie für `MedicalTranscriptionJobName` einen Namen an, der für AWS-Konto eindeutig ist.
 - b. Geben Sie unter `LanguageCode` den Sprachcode an, der der Sprache entspricht, die in Ihrer Audiodatei gesprochen wird.
 - c. Geben Sie unter `MediaFileUri`-Parameter des Media-Objekts den Namen der Audiodatei an, die Sie transkribieren möchten.

- d. Geben Sie unter `Specialty` das medizinische Fachgebiet des Arztes, der in der Audiodatei spricht, als `PRIMARYCARE` an.
- e. Geben Sie für `Type` entweder `CONVERSATION` oder `DICTATION` an.
- f. Unter `OutputBucketName` geben Sie den Bereich Amazon S3 an, in dem Sie die Transkriptionsergebnisse speichern möchten.

Im Folgenden finden Sie eine Beispielanforderung, mit der AWS SDK für Python (Boto3) eine Audiodatei transkribiert und die PHI eines Patienten identifiziert wird.

```
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)
```

Der folgende Beispielcode zeigt die Transkriptionsergebnisse mit identifizierten PHI des Patienten.

```
{
  "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

Um eine Audiodatei zu transkribieren und PHI mithilfe eines Batch-Transkriptionsauftrags zu identifizieren (AWS CLI)

- Führen Sie folgenden Code aus.

```
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
```

Identifizierung von PHI in einem Echtzeit-Stream

Sie können persönliche Gesundheitsinformationen (PHI) entweder in WebSocket Streams HTTP/2 oder in Streams identifizieren. Wenn Sie die PHI-Identifizierung aktivieren, kennzeichnet Amazon Transcribe Medical das PHI, das es in den Transkriptionsergebnissen identifiziert hat. Informationen zu den PHI, die Amazon Transcribe Medical identifizieren kann, finden Sie unter [Identifizierung persönlicher Gesundheitsinformationen \(Personal Health Information, PHI\) in einer Transkription](#).

Identifizieren von PHI in einem Diktat, das in Ihr Mikrofon gesprochen wird

AWS-Managementkonsole Um die von Ihrem Mikrofon aufgenommene Sprache zu transkribieren und etwaige PHI zu identifizieren, wählen Sie Diktat als Audioeingabetyp, starten Sie den Stream und beginnen Sie, in das Mikrofon Ihres Computers zu sprechen.

Um PHI in einem Diktat zu identifizieren, verwenden Sie AWS-Managementkonsole

- Melden Sie sich an der [AWS-Managementkonsole](#) an.
- Wählen Sie im Navigationsbereich Real-time Transkription aus.

3. Wählen Sie als Audioeingangstyp Diktat.
4. Wählen Sie unter Zusätzliche Einstellungendie Option PHI-Identifizieren.
5. Wählen Sie Streaming starten und sprechen Sie in das Mikrofon.
6. Wählen Sie Streaming beenden, um das Diktat zu beenden.

Identifizieren von PHI in einem Stream HTTP/2

Um einen HTTP/2 Stream mit aktivierter PHI-Identifikation zu starten, verwenden Sie die [StartMedicalStreamTranscription](#)API und geben Sie Folgendes an:

- Geben Sie unter LanguageCodeden Sprachencode für die im Stream gesprochene Sprache an. Für US-Englisch geben Sie en-US an.
- Unter MediaSampleHertz geben Sie die Samplerate des Tons an.
- Legen Sie für content-identification-type die Option PHI fest.

Identifizieren von PHI in einem WebSocket Stream

Um einen WebSocket Stream mit aktivierter PHI-Identifikation zu starten, verwenden Sie das folgende Format, um eine vorsignierte URL zu erstellen.

```
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
```

Parameterdefinitionen finden Sie in der [API-Referenz](#). Parameter, die allen AWS API-Operationen gemeinsam sind, sind im Abschnitt [Allgemeine Parameter](#) aufgeführt.

Erstellen alternativer Transkriptionen

Wenn Sie Amazon Transcribe Medical verwenden, erhalten Sie die Transkription mit dem höchsten Konfidenzniveau. Sie können Amazon Transcribe Medical jedoch so konfigurieren, dass zusätzliche Transkriptionen mit niedrigerem Konfidenzniveau zurückgegeben werden.

Verwenden Sie alternative Transkriptionen, um verschiedene Interpretationen des transkribierten Audios anzuzeigen. In einer Anwendung, die es einer Person ermöglicht, die Transkription zu überprüfen, können Sie beispielsweise der Person die alternativen Transkriptionen zur Auswahl präsentieren.

Sie können alternative Transkriptionen mit der oder der AWS-Managementkonsole API generieren. [StartMedicalTranscriptionJob](#)

AWS-Managementkonsole

Um alternative Transkriptionen AWS-Managementkonsole zu generieren, aktivieren Sie alternative Ergebnisse, wenn Sie Ihren Job konfigurieren.

1. Melden Sie sich an der [AWS-Managementkonsole](#) an.
2. Wählen Sie im Navigationsbereich unter Amazon Transcribe Medizinisch die Option Transkriptionsjobs aus.
3. Wählen Sie Job erstellen aus.
4. Geben Sie auf der Seite Auftragsdetails angeben Informationen zu Ihrem Transkriptionsauftrag an.
5. Wählen Sie Weiter aus.
6. Aktivieren Sie Alternative Ergebnisse.
7. Geben Sie unter Maximale Alternativen einen ganzzahligen Wert zwischen 2 und 10 ein, der die maximale Anzahl der alternativen Transkriptionen angibt, die Sie in der Ausgabe haben möchten.
8. Wählen Sie Erstellen aus.

API

So trennen Sie Text nach Sprechern in einer Audiodatei mit einem Batch-Transkriptionsauftrag (API)

- Für die [StartMedicalTranscriptionJob](#)-API geben Sie Folgendes an.

- a. Geben Sie unter `MedicalTranscriptionJobName` einen Namen an, der in Ihrem AWS-Konto eindeutig ist.
- b. Geben Sie unter `LanguageCode` den Sprachcode an, der der in Ihrer Audiodatei gesprochenen Sprache entspricht, sowie die Sprache Ihres Wortschatzfilters.
- c. Geben Sie im Parameter `MediaFileUri` des Objekts `Media` den Speicherort der Audiodatei an, die Sie transkribieren möchten.
- d. Unter `Specialty` können Sie das medizinische Fachgebiet des Arztes angeben, der in der Audiodatei spricht.
- e. Unter `Type` können Sie angeben, ob Sie ein medizinisches Gespräch oder ein Diktat transkribieren.
- f. Unter `OutputBucketName` geben Sie den Amazon S3 -Bucket an, in dem die Transkriptionsergebnisse gespeichert werden.
- g. Für das `Settings`-Objekt geben Sie Folgendes an.
 - i. `ShowAlternatives` – `true`.
 - ii. `MaxAlternatives` – Eine ganze Zahl zwischen 2 und 10, um die Anzahl der alternativen Transkriptionen anzugeben, die Sie in der Transkriptionsausgabe haben möchten.

Die folgende Anfrage verwendet den AWS SDK für Python (Boto3) , um einen Transkriptionsauftrag zu starten, der bis zu zwei alternative Transkriptionen generiert.

```
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

Um eine Audiodatei eines Gesprächs zwischen einem Hausarzt und einem Patienten in einer Audiodatei zu transkribieren (AWS CLI)

- Führen Sie folgenden Code aus.

```

aws transcribe start-transcription-job \
--cli-input-json file://filepath/example-start-command.json

```

Der folgende Code zeigt den Inhalt von `example-start-command.json`.

```

{
    "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 VPC-Endpunkte für medizinische Geräte und Schnittstellen (AWS PrivateLink)

Sie können eine private Verbindung zwischen Ihrer VPC und Amazon Transcribe Medical herstellen, indem Sie einen VPC-Schnittstellen-Endpunkt erstellen. Schnittstellen-Endpunkte basieren auf einer Technologie [AWS PrivateLink](#), die es Ihnen ermöglicht, privat auf Amazon Transcribe medizinische APIs zuzugreifen, ohne ein Internet-Gateway, ein NAT-Gerät, eine VPN-Verbindung oder eine Verbindung zu benötigen. Direct Connect Instances in Ihrer VPC benötigen keine öffentlichen IP-Adressen, um mit Amazon Transcribe medizinischen APIs zu kommunizieren. Der Datenverkehr zwischen Ihrer VPC und Amazon Transcribe Medical verlässt das Amazon-Netzwerk nicht.

Jeder Schnittstellenendpunkt wird durch eine oder mehrere [Elastic-Netzwerk-Schnittstellen](#) in Ihren Subnetzen dargestellt.

Weitere Informationen finden Sie unter [Schnittstellen-VPC-Endpunkte \(AWS PrivateLink\)](#) im Amazon VPC -Benutzerhandbuch.

Überlegungen zu Amazon Transcribe Medizinische VPC-Endpunkte

Bevor Sie einen Schnittstellen-VPC-Endpunkt für Amazon Transcribe Medical einrichten, sollten Sie die [Eigenschaften und Einschränkungen der Schnittstellenendpunkte](#) im Amazon VPC Benutzerhandbuch lesen.

Amazon Transcribe Medical unterstützt Aufrufe aller API-Aktionen von Ihrer VPC aus.

Erstellen eines Schnittstellen-VPC-Endpunkts für Amazon Transcribe Medical

Sie können einen VPC-Endpunkt für den Amazon Transcribe medizinischen Dienst erstellen, indem Sie entweder den AWS-Managementkonsole oder den AWS CLI verwenden. Weitere Informationen finden Sie unter [Erstellen eines Schnittstellenendpunkts](#) im Amazon VPC -Benutzerhandbuch.

Erstellen Sie für die Batch-Transkription in Amazon Transcribe Medical einen VPC-Endpunkt mit dem folgenden Dienstnamen:

- `com.amazonaws.us-west-2.transkribieren`

Erstellen Sie für die Streaming-Transkription in Amazon Transcribe Medical einen VPC-Endpunkt mit dem folgenden Dienstnamen:

- `com.amazonaws.us-west-2.transkribiert Streaming`

Wenn Sie privates DNS für den Endpunkt aktivieren, können Sie API-Anfragen an Amazon Transcribe Medical stellen, indem Sie dessen Standard-DNS-Namen für verwenden AWS-Region, zum Beispiel `transcribestreaming.us-east-2.amazonaws.com`

Weitere Informationen finden Sie unter [Zugriff auf einen Service über einen Schnittstellenendpunkt](#) im Amazon VPC -Benutzerhandbuch.

Erstellen einer VPC-Endpunkttrichtlinie für Amazon Transcribe Medizinisches Streaming

Sie können Ihrem VPC-Endpunkt eine Endpunkttrichtlinie hinzufügen, die den Zugriff auf Amazon Transcribe Medical steuert. Die Richtlinie gibt die folgenden Informationen an:

- Prinzipal, der die Aktionen ausführen kann.
- Aktionen, die ausgeführt werden können
- Die Ressourcen, für die Aktionen ausgeführt werden können.

Weitere Informationen finden Sie unter [Steuerung des Zugriffs auf Services mit VPC-Endpunkten](#) im Amazon-VPC-Benutzerhandbuch.

Beispiel: VPC-Endpunktrichtlinie für Amazon Transcribe Streaming-Transkriptionsaktionen für medizinische Zwecke

Im Folgenden finden Sie ein Beispiel für eine Endpunktrichtlinie für Streaming-Transkription in Amazon Transcribe Medical. Wenn diese Richtlinie an einen Endpunkt angehängt ist, gewährt sie allen Schulleitern auf allen Ressourcen Zugriff auf die aufgelisteten Amazon Transcribe medizinischen Aktionen.

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Allow",
      "Action": [
        "transcribe:StartMedicalStreamTranscription",
      ],
      "Resource": "*"
    }
  ]
}
```

Beispiel: VPC-Endpunktrichtlinie für Amazon Transcribe Aktionen zur Batch-Transkription für medizinische Zwecke

Im Folgenden finden Sie ein Beispiel für eine Endpunktrichtlinie für die Batch-Transkription in Amazon Transcribe Medical. Wenn diese Richtlinie an einen Endpunkt angehängt ist, gewährt sie allen Schulleitern auf allen Ressourcen Zugriff auf die aufgelisteten Amazon Transcribe medizinischen Aktionen.

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Allow",
      "Action": [
        "transcribe:StartMedicalTranscriptionJob"
      ],
      "Resource": "*"
    }
  ]
}
```

Gemeinsam genutzte Subnetze

Sie können VPC-Endpunkte in Subnetzen, die mit Ihnen gemeinsam genutzt werden, nicht erstellen, beschreiben, ändern oder löschen. Sie können die VPC-Endpunkte jedoch in Subnetzen verwenden, die mit Ihnen geteilt werden. Informationen zur VPC-Sharing finden Sie im [Leitfaden unter Teilen Ihrer VPC mit anderen Konten](#). Amazon Virtual Private Cloud

AWS HealthScribe

Wir stellen vor: Amazon Connect Health Ambient Agent

Sie können jetzt den Ambient Agent verwenden, um Ihren Arbeitsablauf im Gesundheitswesen voranzutreiben. Weitere Informationen finden Sie in der Dokumentation zu [Amazon Connect Health Ambient](#).

AWS HealthScribe ist eine Funktion für HIPAA-eligible maschinelles Lernen (ML), die Spracherkennung und generative KI kombiniert, um Gespräche zwischen Patienten und Ärzten zu transkribieren und leicht zu überprüfende klinische Notizen zu generieren. AWS HealthScribe unterstützt Anbieter von Gesundheitssoftware bei der Entwicklung klinischer Anwendungen, die den Dokumentationsaufwand reduzieren und das Beratungserlebnis verbessern. Der Service stellt automatisch umfangreiche Gesprächsprotokolle bereit, identifiziert Sprecherrollen, klassifiziert Dialoge, extrahiert medizinische Fachbegriffe und generiert vorläufige klinische Notizen. AWS HealthScribekombiniert diese Funktionen, sodass keine separaten KI-Services integriert und optimiert werden müssen, sodass Sie die Implementierung beschleunigen können.

Häufige Anwendungsfälle:

- Verkürzen Sie den Zeitaufwand für die Dokumentation — Ermöglichen Sie es Ärzten, die klinische Dokumentation schnell zu vervollständigen — mit AI-generated klinischen Notizen, die Sie in Ihrer Anwendung einfach überprüfen, anpassen und finalisieren können.
- Steigern Sie die Effizienz der medizinischen Mitarbeiter — Statten Sie medizinische Schreiberlinge mit AI-generated Transkripten und klinischen Notizen sowie dem Audio der Konsultation aus, um die Bearbeitungszeit der Dokumentation zu verkürzen.
- Effiziente Rekapitulation des Patientenbesuchs – Schaffen Sie ein Erlebnis, das es den Nutzern ermöglicht, sich schnell an die wichtigsten Punkte ihres Gesprächs in Ihrer Anwendung zu erinnern.

Important

Die Ergebnisse von AWS HealthScribe sind probabilistisch und aufgrund verschiedener Faktoren wie Audioqualität, Hintergrundgeräusche, Klarheit der Sprecher, Komplexität der medizinischen Terminologie, kontextspezifischer Sprachnuancen und der Art des maschinellen Lernens und der generativen KI möglicherweise nicht immer korrekt. AWS

HealthScribe ist so konzipiert, dass es Klinikern und medizinischem Fachpersonal assistieren kann. AWS HealthScribe Die Ergebnisse sollten nur für Patientenversorgungsszenarien verwendet werden, einschließlich, aber nicht beschränkt auf elektronische Patientenakten, nachdem sie von geschultem medizinischem Fachpersonal auf ihre Richtigkeit überprüft und ein fundiertes medizinisches Urteilsvermögen vorausgesetzt wurde. AWS HealthScribe Die Ergebnisse sind kein Ersatz für professionelle medizinische Beratung, Diagnose oder Behandlung und dienen nicht der Heilung, Behandlung, Linderung, Vorbeugung oder Diagnose von Krankheiten oder Gesundheitszuständen.

Themen

- [Sicherheit](#)
- [Service-Verfügbarkeit](#)
- [Technische Anforderungen](#)
- [Unterstützte medizinische Fachgebiete](#)
- [Arbeitsabläufe](#)
- [AWS HealthScribe Transkriptdatei](#)
- [AWS HealthScribe Datei mit klinischer Dokumentation](#)
- [AWS HealthScribe Transkriptionsjobs](#)
- [AWS HealthScribe streamen](#)
- [Datenverschlüsselung im Ruhezustand für AWS HealthScribe](#)

Sicherheit

AWS HealthScribe arbeitet nach einem Modell der geteilten Verantwortung, wobei AWS Sie für den Schutz der laufenden Infrastruktur verantwortlich sind AWS HealthScribe und Sie für die Verwaltung Ihrer Daten verantwortlich sind. Weitere Informationen finden Sie unter [-Modell der geteilten Verantwortung](#).

AWS HealthScribe Bietet standardmäßig Verschlüsselung im Ruhezustand, um sensible Kundendaten mithilfe von Amazon S3-verwalteten Schlüsseln zu schützen. Wenn Sie einen AWS HealthScribe Transkriptionsauftrag erstellen oder einen Stream starten, können Sie einen vom Kunden verwalteten Schlüssel angeben. Dadurch wird eine zweite Verschlüsselungsebene hinzugefügt. Weitere Informationen finden Sie unter [Datenverschlüsselung im Ruhezustand für AWS HealthScribe](#).

Service-Verfügbarkeit

AWS HealthScribe ist in der Region USA Ost (Nord-Virginia) verfügbar.

Technische Anforderungen

- Unterstützte Sprache: US-Englisch (en-US)
- Empfohlenes Audioformat: Verlustfreies Audio (wie FLAC oder WAV)
- Kodierung: PCM 16-Bit
- Abtastrate: 16.000 Hz oder höher

Unterstützte medizinische Fachgebiete

AWS HealthScribe unterstützt derzeit die folgenden Fachgebiete:

- Allergie-Immunologie
- Kardiologie
- Dermatologie
- Endokrinologie
- Gastroenterologie
- Hematology/Oncology
- Infektiöse Krankheit
- Nephrologie
- Neurologie
- FRAUENHEILKUNDE
- Onkologie
- Augenheilkunde
- Orthopädie
- Hals-Nasen-Ohrenheilkunde
- Schmerzmedizin
- Pädiatrie
- Grundversorgung

- Psychiatrie
- Pulmonologie
- Rheumatologie
- Chirurgie
- Urologie

Arbeitsabläufe

AWS HealthScribe Zu den Workflows gehören Transkriptionsaufträge und Streaming. Nachdem Sie einen Transkriptionsauftrag ausgeführt oder einen Stream abgeschlossen haben, AWS HealthScribe generiert es eine Transkriptdatei mit Transkriptionsausgabe und Erkenntnissen für jede Konversationsrunde. Außerdem wird eine klinische Dokumentationsdatei mit Zusammenfassungen und Links zu Nachweisen generiert. Weitere Informationen finden Sie unter [AWS HealthScribe Transkriptdatei](#) und [AWS HealthScribe Datei mit klinischer Dokumentation](#).

- Transkriptionsaufträge — AWS HealthScribe Analysiert bei Transkriptionsaufträgen abgeschlossene Mediendateien für medizinische Konsultationen aus einem Amazon S3 S3-Bucket. Im Folgenden sind API-Operationen aufgeführt, die für AWS HealthScribe Transkriptionsaufträge spezifisch sind.
 - [StartMedicalScribeJob](#)
 - [ListMedicalScribeJobs](#)
 - [GetMedicalScribeJob](#)
 - [DeleteMedicalScribeJob](#)

Weitere Informationen, einschließlich Codebeispielen, finden Sie unter [AWS HealthScribe Transkriptionsjobs](#).

- Streaming — AWS HealthScribe Streaming ist ein auf HTTP2 basierender bidirektionaler Echtzeitdienst, der Audiostreams auf einem Kanal akzeptiert und eine Audiotranskription auf dem anderen Kanal verkauft.

Im Folgenden sind API-Operationen aufgeführt, die für Streaming spezifisch sind: AWS HealthScribe

- [StartMedicalScribeStream](#)
- [GetMedicalScribeStream](#)

Weitere Informationen, einschließlich Codebeispielen, finden Sie unter [AWS HealthScribe streamen](#).

AWS HealthScribe Transkriptdatei

In der Transkriptdatei stehen Ihnen zusätzlich zur standardmäßigen Transkriptionsausgabe mit Zeitstempeln auf Wortebene folgende Informationen zur Verfügung: AWS HealthScribe

- Erkennung der Teilnehmerrolle, sodass Sie im Gesprächsprotokoll die Patienten von den Ärzten unterscheiden können.
- Transkriptionsgliederung, die Transkriptdialoge nach ihrer klinischen Relevanz kategorisiert, z. B. Small Talk, subjektiv, objektiv usw. Damit lassen sich bestimmte Teile des Transkripts anzeigen.
- Klinische Entitäten, die strukturierte Informationen wie Medikamente, Erkrankungen und Behandlungen enthalten, die im Gespräch erwähnt wurden.

Darüber hinaus werden für jeden Schritt die folgenden Erkenntnisse geliefert:

- Rolle des Teilnehmers – Jeder Teilnehmer ist entweder als Arzt oder als Patient gekennzeichnet. Wenn ein Gespräch mehr als einen Teilnehmer in jeder Kategorie hat, wird jedem Teilnehmer eine Nummer zugewiesen. Zum Beispiel: CLINICIAN_0, CLINICIAN_1 und PATIENT_0, PATIENT_1.
- Abschnitt – Jeder Dialog wird einem von vier möglichen Abschnitten zugewiesen, basierend auf dem identifizierten Inhalt.
 - Subjektiv – Vom Patienten bereitgestellte Informationen zu seinen gesundheitlichen Problemen.
 - Zielsetzung – Informationen, die der Arzt im Rahmen von körperlichen Untersuchungen, Labor-, Bildgebungs- oder Diagnosetests beobachtet hat.
 - Beurteilung und Plan – Informationen, die sich auf die Beurteilung und den Behandlungsplan des Arztes beziehen.
 - Besuchen Sie Flow Management — Informationen zu Smalltalk oder Übergängen.
- Einblicke — Extrahieren Sie klinisch relevante Entitäten (ClinicalEntity), die in der Konversation anwesend sind. AWS HealthScribe erkennt alle klinischen Einheiten, die von [Amazon Comprehend Medical](#) unterstützt werden.

Ein Beispiel für ein Transkript eines Transkriptionsauftrags finden Sie in der Transkriptausgabe unter [Beispiele für die Ausgabe von Transkriptionsjobs](#). Ein Beispiel für ein Transkript aus dem Streaming finden Sie in der Transkriptausgabe unter [Beispiele für die Ausgabe von Streaming-Transkriptionen](#).

AWS HealthScribe Datei mit klinischer Dokumentation

AWS HealthScribe kann eine der folgenden Vorlagen für die Zusammenfassung der klinischen Notiz verwenden. Der Standardwert ist HISTORY_AND_PHYSICAL.

- **HISTORY_AND_PHYSICAL:** Enthält Zusammenfassungen der wichtigsten Abschnitte der klinischen Dokumentation. Zu den Abschnitten gehören beispielsweise Hauptbeschwerde, Vorgeschichte der aktuellen Krankheit, Überblick über die Systeme, Krankengeschichte in der Vergangenheit, Bewertung und Plan.
- **GIRPP:** Bietet Zusammenfassungen, die auf den Fortschritten der Patienten bei der Erreichung ihrer Ziele basieren. Zu den Abschnitten gehören beispielsweise „Ziel“, „Intervention“, „Reaktion“, „Fortschritt“ und „Plan“.
- **BIRP:** Konzentriert sich auf die Verhaltensmuster und Reaktionen des Patienten. Zu den Abschnitten gehören beispielsweise Verhalten, Intervention, Reaktion und Plan.
- **SIRP:** Betont den situativen Kontext der Therapie. Zu den Abschnitten gehören beispielsweise Situation, Intervention, Reaktion und Plan.
- **DAP:** Bietet ein vereinfachtes Format für die klinische Dokumentation. Zu den Abschnitten gehören beispielsweise Daten, Bewertung und Plan.
- **BH_SOAP:** Dokumentationsformat mit Fokus auf Verhaltensgesundheit. Zu den Abschnitten gehören beispielsweise „Subjektiv“, „Objektiv“, „Bewertung“ und „Plan“.
- **PH_SOAP:** Dokumentationsformat mit Fokus auf körperliche Gesundheit. Zu den Abschnitten gehören beispielsweise „Subjektiv“, „Objektiv“, „Bewertung“ und „Plan“.

Gehen Sie wie folgt vor, um anzugeben, welche Vorlage verwendet werden soll:

- Geben Sie für Transkriptionsaufträge die Vorlage an, die im NoteTemplate Rahmen Ihres [StartMedicalScribeJob](#)API-Vorgangs verwendet werden soll. [ClinicalNoteGenerationSettings](#)Settings
- Für das Streaming geben Sie die Vorlage an, die in der NoteTemplate [ClinicalNoteGenerationSettings](#)PostStreamAnalyticsSettingsvon Ihnen [MedicalScribeConfigurationEvent](#)verwendet werden soll.

Kontext von Medical Scribe

Das `MedicalScribeContext` Objekt enthält Kontextinformationen, mit denen die Erstellung klinischer Notizen mit patientenspezifischen Details individuell angepasst werden kann.

Das `MedicalScribeContext` Objekt umfasst die folgenden Komponenten:

- **PatientContext**- Enthält patientenspezifische Informationen, die zur individuellen Generierung klinischer Notizen verwendet werden. Dies umfasst:
 - **Pronomen** — Die bevorzugten Pronomen des Patienten, die Sie als Kontext für die Erstellung klinischer Notizen angeben möchten. Diese Pronomen werden verwendet, wenn in der generierten klinischen Ausgabe auf den Patienten Bezug genommen wird.

Informationen zur Einrichtung des Medical Scribe-Kontextes finden Sie unter. [MedicalScribeContext](#)

Themen

- [Abschnitte mit den Vorlagen HISTORY_AND_PHYSICAL](#)
- [Abschnitte mit GIRPP-Vorlagen](#)
- [Abschnitte mit BIRP-Vorlagen](#)
- [Abschnitte mit SIRP-Vorlagen](#)
- [Abschnitte mit DAP-Vorlagen](#)
- [Abschnitte der BH_SOAP-Vorlage](#)
- [Abschnitte der PH_SOAP-Vorlage](#)

Abschnitte mit den Vorlagen HISTORY_AND_PHYSICAL

Die Insights-Vorlage HISTORY_AND_PHYSICAL umfasst die folgenden Abschnitte.

Abschnitt	Description
HAUPTBESCHWERDE	Kurze Beschreibung des Grundes, aus dem der Patient den Arzt aufgesucht hat.
VORGESCHICHTE DER AKTUELLEN KRANKHEIT	Notizen, die Informationen über die Krankheit des Patienten enthalten, einschließlich Angaben über Schweregrad, Beginn und

Abschnitt	Description
	Zeitpunkt der Symptome, aktuelle Behandlungen und betroffene Bereiche.
SYSTEMÜBERPRÜFUNG	Patient-reported Bewertung der Symptome in verschiedenen Körpersystemen.
KRANKGESCHICHTE IN DER VERGANGENHEIT	Informationen zu den früheren Erkrankungen, Operationen und Behandlungen eines Patienten.
BEWERTUNG	Notizen, die Aufschluss über die Beurteilung des Gesundheitszustandes des Patienten durch den Arzt geben.
PLAN	Notizen zu medizinischen Behandlungen, Anpassungen des Lebensstils und weiteren Terminen.
KÖRPERLICHE UNTERSUCHUNG	Dokumentation der Ergebnisse der körperlichen Untersuchung des Körpersystems und der Vitalfunktionen des Patienten durch den Arzt.
VERGANGENE FAMILIENGESCHICHTE	Informationen über Gesundheitsprobleme, die in der Familie des Patienten auftreten.
PAST_SOCIAL_HISTORY	Einzelheiten über das Sozialleben, die Gewohnheiten, den Beruf und Umweltfaktoren des Patienten, die sich auf die Gesundheit auswirken.
DIAGNOSTISCHES TESTEN	Ergebnisse und Interpretation von Labortests, bildgebenden Studien und anderen diagnostischen Verfahren.

Jeder Satz in den Summary enthält EvidenceLinks die Angaben zu SegmentId den relevanten Dialogen im Protokoll, die zusammengefasst wurden. Auf diese Weise können Benutzer die Richtigkeit der Zusammenfassung in Ihrer Anwendung überprüfen. Wie die Erklärbarkeit steht

auch die Gewährleistung der Rückverfolgbarkeit und Transparenz von AI-generated Erkenntnissen im Einklang mit den Prinzipien der verantwortungsvollen KI. Die Bereitstellung dieser Verweise zusammen mit den zusammenfassenden Hinweisen für Ärzte oder medizinische Fachangestellte trägt dazu bei, das Vertrauen zu stärken und den sicheren Einsatz von KI im klinischen Umfeld zu fördern.

Ein Beispiel für eine Datei mit klinischer Dokumentation aus einem Transkriptionsauftrag finden Sie im Beispiel für eine Datei mit klinischer Dokumentation unter. [Beispiele für die Ausgabe von Transkriptionsjobs](#) Ein Beispiel für eine Datei mit klinischer Dokumentation aus dem Streaming finden Sie im [Beispiele für die Ausgabe von Streaming-Transkriptionen](#) Beispiel für eine Datei mit klinischer Dokumentation unter.

Abschnitte mit GIRPP-Vorlagen

Die GIRPP-Insights-Vorlage umfasst die folgenden Abschnitte.

Abschnitt	Description
Ziel	Das identifizierte Problem, die Herausforderung oder das Verhalten, das durch eine Behandlung angegangen werden muss.
Intervention	Die spezifische Behandlung, Methode oder Technik, die der Arzt anwendet, um dem Patienten zu helfen, das festgelegte Ziel zu erreichen.
Antwort	Wie der Patient auf die Intervention reagiert hat, einschließlich seiner Beteiligung, seiner Reaktionen und seines Feedbacks.
Progress	Einschätzung der Fortschritte beim Erreichen der Behandlungsziele durch den Arzt, einschließlich Beobachtungen von Verbesserungen oder Hindernissen beim Patienten.
Plan	Die nächsten Behandlungsschritte, einschließlich future Interventionen, Hausaufgaben und Überweisungen.

Abschnitte mit BIRP-Vorlagen

Die BIRP-Insights-Vorlage umfasst die folgenden Abschnitte.

Abschnitt	Description
Behavior	Die Probleme, mit denen der Patient konfrontiert ist, und seine Reaktion auf die Behandlung.
Intervention	Die spezifische Behandlung, Methode oder Technik, die der Arzt anwendet, um dem Patienten zu helfen, das festgelegte Ziel zu erreichen.
Antwort	Wie der Patient auf die Intervention reagiert hat, einschließlich seiner Beteiligung, seiner Reaktionen und seines Feedbacks.
Plan	Die nächsten Behandlungsschritte, einschließlich future Interventionen, Hausaufgaben und Überweisungen.

Abschnitte mit SIRP-Vorlagen

Die SIRP-Insights-Vorlage umfasst die folgenden Abschnitte.

Abschnitt	Description
Situation	Das Problem, mit dem der Patient konfrontiert ist, und sein Ziel für die Suche nach einer Therapie.
Intervention	Die spezifische Behandlung, Methode oder Technik, die der Arzt anwendet, um dem Patienten zu helfen, das festgelegte Ziel zu erreichen.

Abschnitt	Description
Antwort	Wie der Patient auf die Intervention reagiert hat, einschließlich seiner Beteiligung, seiner Reaktionen und seines Feedbacks.
Plan	Die nächsten Behandlungsschritte, einschließlich future Interventionen, Hausaufgaben und Überweisungen.

Abschnitte mit DAP-Vorlagen

Die Vorlage für DAP Insights umfasst die folgenden Abschnitte.

Abschnitt	Description
Daten	Die Gründe des Patienten für die Inanspruchnahme einer Behandlung und Informationen über den Patienten.
Bewertung	Die Diagnose der Situation des Patienten durch den Arzt.
Plan	Die nächsten Behandlungsschritte, einschließlich future Interventionen, Hausaufgaben und Überweisungen.

Abschnitte der BH_SOAP-Vorlage

Die Vorlage BH_SOAP Insights umfasst die folgenden Abschnitte.

Abschnitt	Description
Subjektiv	Die Behandlungsziele des Klienten und seine Erfahrungen sowie bestehende und vergangene Probleme.

Abschnitt	Description
Ziel	Daten und Fakten über den Kunden.
Bewertung	Die Diagnose der Situation des Patienten durch den Arzt.
Plan	Die nächsten Behandlungsschritte, einschließlich future Interventionen, Hausaufgaben und Überweisungen.

Abschnitte der PH_SOAP-Vorlage

Die Vorlage PH_SOAP Insights umfasst die folgenden Abschnitte.

Abschnitt	Description
Subjektiv	Die Behandlungsziele des Klienten und seine Erfahrungen sowie bestehende und vergangene Probleme.
Ziel	Daten und Fakten über den Kunden.
Bewertung	Die Diagnose der Situation des Patienten durch den Arzt.
Plan	Die nächsten Behandlungsschritte, einschließlich future Interventionen, Hausaufgaben und Überweisungen.

AWS HealthScribe Transkriptionsjobs

Ein AWS HealthScribe Transkriptionsjob verarbeitet Mediendateien aus einem Amazon S3 S3-Bucket. [Bei der Verarbeitung einer Mediendatei werden Gespräche zwischen Patient und Arzt transkribiert und medizinische Konsultationen analysiert, um zwei JSON-Ausgabedateien zu erstellen: eine Transkriptdatei und eine klinische Dokumentationsdatei.](#)

Die folgenden API-Operationen sind spezifisch für Transkriptionsaufträge: AWS HealthScribe

- [StartMedicalScribeJob](#)
- [ListMedicalScribeJobs](#)
- [GetMedicalScribeJob](#)
- [DeleteMedicalScribeJob](#)

Starten eines AWS HealthScribe Transkriptionsjob

Sie können einen AWS HealthScribe Job mit der AWS CLI oder den AWS SDKs starten.

AWS CLI

In diesem Beispiel wird der Befehl [start-medical-scribe-job](#) verwendet. Weitere Informationen finden Sie unter [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
```

Hier ein weiteres Beispiel mit dem Befehl [start-medical-scribe-job](#) und einem Anforderungstext mit zusätzlichen Einstellungen.

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

Die Datei `my-first-medical-scribe-job.json` enthält den folgenden Anforderungstext.

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

AWS SDK für Python (Boto3)

Im folgenden Beispiel wird die verwendete, AWS SDK für Python (Boto3) um eine [start_medical_scribe_job-Anfrage](#) zu stellen. Weitere Informationen finden Sie unter [StartMedicalScribeJob](#).

```
from __future__ import print_function
import time
import 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

Die AWS Management Console unterstützt derzeit keine Jobs. AWS HealthScribe

Beispiele für die Ausgabe von Transkriptionsjobs

Zusätzlich zu einem Transkript erzeugen `StartMedicalScribeJob`-Anforderungen eine separate Datei mit klinischer Dokumentation. Beide Dateien haben das JSON-Format und werden an dem Ausgabeort gespeichert, den Sie in Ihrer Anfrage angeben. Hier finden Sie Beispiele für die einzelnen Ausgabearten:

Beispiel einer Transkriptionsausgabe

Eine AWS HealthScribe Transkriptdatei (aus einer `StartMedicalScribeJob` Anfrage) hat das folgende Format:

```

{
  "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"
  }
]
}
}

```

Hier ein weiteres Beispiel mit dem Befehl [start-medical-scribe-job](#) und einem Anforderungstext mit zusätzlichen Einstellungen.

```

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

```

Die Datei `my-first-medical-scribe-job.json` enthält den folgenden Anforderungstext.

```

{
  "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"
  }
]
}
```

Beispiel für die Ausgabe klinischer Dokumentation

Eine Datei mit Dokumentationseinblicken (aus einer `StartMedicalScribeJob`-Anfrage) hat folgendes Format:

```
{
  "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 streamen

Mit AWS HealthScribe Streaming können Sie medizinische Gespräche in Echtzeit transkribieren. AWS HealthScribe Streaming ist ein auf HTTP2 basierender bidirektionaler Echtzeitdienst, der Audiostreams auf einem Kanal akzeptiert und eine Audiotranskription auf dem anderen Kanal verkauft. AWS HealthScribe analysiert nach Abschluss des Streamings den Inhalt des Streams und erstellt eine JSON-Datei mit Transkript und eine JSON-Datei für klinische Notizen.

Verwenden Sie den [StartMedicalScribeStream](#) API-Vorgang, um das Streaming zu starten. Diese API startet einen HTTP2-basierten bidirektionalen Kanal, den Sie zum Streamen von Audioereignissen verwenden.

Wenn Sie einen Stream starten, geben Sie zunächst die Stream-Konfiguration an. `MedicalScribeConfigurationEvent` Dieses Ereignis umfasst Kanaldefinitionen, Verschlüsselungseinstellungen und Einstellungen für die Post-Stream-Analyse, z. B. die Ausgabekonfiguration für aggregierte Transkripte und die Generierung klinischer Notizen.

Nachdem Sie mit dem Audio-Streaming begonnen haben, verwalten Sie den Stream wie folgt:

- Wenn Sie fertig sind und mit der Verarbeitung der Ergebnisse mit der Post-Stream-Analyse beginnen möchten, senden Sie ein Zeichen `MedicalScribeSessionControlEvent` mit dem Zeichen `Type` von `END_OF_SESSION` und AWS HealthScribe starten Sie die Analyse.
- Um das Streaming zu unterbrechen, schließen Sie den Eingabestream ab, ohne das `MedicalScribeSessionControlEvent` zu senden.
- Um einen angehaltenen Stream wieder aufzunehmen, verwenden Sie den `StartMedicalScribeStream` API-Vorgang und geben Sie ihn an. `SessionId` Dies ist der, den `SessionId` Sie verwendet haben, als Sie den Stream ursprünglich gestartet haben.

Themen

- [Richtlinien und Anforderungen](#)

- [ResourceAccessRoleArn Rollen, Berechtigungen](#)
- [Wird gestartet AWS HealthScribe Streaming-Transkription](#)

Richtlinien und Anforderungen

Im Folgenden finden Sie Richtlinien und Anforderungen für das AWS HealthScribe Streaming:

- Bevor Sie Audioereignisse senden, müssen Sie zunächst die Stream-Konfiguration in a [angebenMedicalScribeConfigurationEvent](#).
- Um Post-Stream-Analysen ausführen zu können, `MedicalScribeConfigurationEvent` müssen Sie über die richtigen Berechtigungen verfügen. `ResourceAccessRoleArn` Weitere Informationen finden Sie unter [ResourceAccessRoleArn Rollen, Berechtigungen](#).
- Sie können eine Sitzung innerhalb von 5 Stunden nach der ersten Stream-Erstellung beliebig oft wieder aufnehmen.
- Sie können in einer Sitzung für alle Streaming-Anfragen maximal 2 Stunden Audio streamen.
- AWS HealthScribe Bietet standardmäßig Verschlüsselung im Ruhezustand, um sensible Kundendaten mithilfe von Amazon S3-verwalteten Schlüsseln zu schützen. Wenn Sie einen Stream starten, können Sie einen AWS KMS Schlüssel für eine zweite Verschlüsselungsebene angeben. Sie `ResourceAccessRoleArn` müssen über die Erlaubnis verfügen, Ihren AWS KMS Schlüssel zu verwenden. Weitere Informationen finden Sie unter [Datenverschlüsselung im Ruhezustand für AWS HealthScribe](#).
- Sie können AWS HealthScribe Streaming mit den AWS SDKs verwenden, mit Ausnahme des SDK for Python (Boto3) und des SDK for PHP.
- Wenn nach dem Beenden eines Streams eine `LimitExceededException` Ausnahme auftritt, können Sie die Sitzung neu starten und trotzdem Analysen nach dem Stream generieren. Verwenden Sie die [StartMedicalScribeStream](#) API und verwenden Sie dieselbe `SessionID`, um den Stream neu zu starten. Senden Sie dann eine `MedicalScribeSessionControlEvent` mit einem Type von `END_OF_SESSION` und AWS HealthScribe starten Sie die Analyse.

ResourceAccessRoleArn Rollen, Berechtigungen

Um Post-Stream-Analysen ausführen zu können, `MedicalScribeConfigurationEvent` muss Ihr `ResourceAccessRoleArn` in der Lage sein, auf Ihren Amazon S3 S3-Ausgabe-Bucket und, falls Sie ihn angeben, auf Ihren AWS KMS Schlüssel zuzugreifen. Außerdem muss die

Vertrauensrichtlinie der Rolle dem `transcribe.streaming.amazonaws.com` Service die Erlaubnis erteilen, die Rolle zu übernehmen.

Im Folgenden finden Sie ein Beispiel für eine IAM-Richtlinie, die Amazon S3 S3-Bucket-Berechtigungen und AWS KMS Schlüsselberechtigungen gewährt. Weitere Informationen finden Sie unter [Datenverschlüsselung im Ruhezustand für AWS HealthScribe](#).

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"
    }
  ]
}
```

Im Folgenden finden Sie ein Beispiel für eine Vertrauensrichtlinie.

JSON

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

Wird gestartet AWS HealthScribe Streaming-Transkription

Das folgende Codebeispiel zeigt, wie Sie eine AWS HealthScribe Streaming-Transkription mithilfe der AWS SDKs einrichten.

Themen

- [SDK für Java 2.x](#)
- [Beispiele für die Ausgabe von Streaming-Transkriptionen](#)

SDK für Java 2.x

Das folgende Beispiel verwendet das SDK for Java 2.x, um Streaming einzurichten und eine [StartMedicalScribeStream](#)Anfrage zu stellen.

```
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.DefaultConfiguration;

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);
}
}
}

```

Beispiele für die Ausgabe von Streaming-Transkriptionen

AWS HealthScribe analysiert nach Abschluss des Streamings den Inhalt des Streams und erstellt eine JSON-Datei mit Transkript und eine JSON-Datei für klinische Notizen. Hier finden Sie Beispiele für die einzelnen Ausgabearten:

Beispiel einer Transkriptionsausgabe

Im Folgenden finden Sie ein Beispiel für eine AWS HealthScribe Transkriptdatei aus einer Streaming-Sitzung.

```

{
  "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"
    }
}
```

Beispiel für eine Ausgabe klinischer Unterlagen

Im Folgenden finden Sie ein Beispiel für eine Datei mit Erkenntnissen zur AWS HealthScribe klinischen Dokumentation aus einer Streaming-Sitzung.

```
{
  "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"
}

```

Datenverschlüsselung im Ruhezustand für AWS HealthScribe

AWS HealthScribe bietet standardmäßig Verschlüsselung im Ruhezustand, um vertrauliche Kundendaten mithilfe von AWS HealthScribe verwalteten AWS Key Management Service (AWS KMS) -Schlüsseln zu schützen. Die standardmäßige Verschlüsselung von Daten im Ruhezustand trägt dazu bei, den betrieblichen Aufwand und die Komplexität zu reduzieren, die mit dem Schutz vertraulicher Daten verbunden sind. Außerdem ermöglicht es Ihnen, sichere Anwendungen zu entwickeln, die strenge Verschlüsselungsvorschriften und regulatorische Anforderungen erfüllen. Wenn Sie einen AWS HealthScribe Transkriptionsauftrag erstellen oder einen Stream starten, können Sie einen vom Kunden verwalteten Schlüssel angeben. Dadurch wird eine zweite Verschlüsselungsebene hinzugefügt.

- **AWS HealthScribe verwaltete AWS KMS Schlüssel** — AWS HealthScribe verwendet standardmäßig AWS HealthScribe verwaltete AWS Key Management Service (AWS KMS) -Schlüssel, um Zwischendateien automatisch zu verschlüsseln. Sie können diese Verschlüsselungsebene nicht deaktivieren oder einen anderen Verschlüsselungstyp wählen. Sie können die Schlüssel nicht einsehen, verwalten oder verwenden oder ihre Verwendung überprüfen. Sie müssen jedoch keine Maßnahmen ergreifen oder Programme zum Schutz der Schlüssel ändern, die zur Verschlüsselung Ihrer Daten verwendet werden.
- **Vom Kunden verwaltete Schlüssel** — AWS HealthScribe unterstützt die Verwendung eines symmetrischen, vom Kunden verwalteten Schlüssels, den Sie erstellen, besitzen und verwalten, um eine zweite Verschlüsselungsebene über der vorhandenen AWS-eigenen Verschlüsselung hinzuzufügen. Da Sie die volle Kontrolle über diese Verschlüsselungsebene haben, können Sie beispielsweise folgende Aufgaben ausführen:
 - Festlegung und Pflege wichtiger Richtlinien
 - Festlegung und Aufrechterhaltung von IAM Richtlinien und Zuschüssen
 - Aktivieren und Deaktivieren wichtiger Richtlinien
 - Kryptographisches Material mit rotierendem Schlüssel
 - Hinzufügen von -Tags
 - Erstellen von Schlüsselaliasen
 - Schlüssel für das Löschen von Schlüsseln planen

Weitere Informationen finden Sie unter vom [Kunden verwalteter Schlüssel](#) im AWS Key Management Service Entwicklerhandbuch.

 Note

AWS HealthScribe aktiviert automatisch die Verschlüsselung im AWS Ruhezustand mithilfe eigener Schlüssel, um personenbezogene Daten kostenlos zu schützen. Für die Verwendung eines vom Kunden verwalteten Schlüssels fallen jedoch AWS KMS Gebühren an. Weitere Informationen zur Preisgestaltung finden Sie unter [AWS Key Management Service – Preise](#). Weitere Informationen zu AWS KMS finden Sie unter [Was ist AWS Key Management Service](#).

Themen

- [Einen vom Kunden verwalteten Schlüssel erstellen](#)
- [Angabe eines vom Kunden verwalteten Schlüssels für AWS HealthScribe](#)
- [AWS KMS Verschlüsselungskontext](#)
- [Überwachen Sie Ihre Verschlüsselungsschlüssel für AWS HealthScribe](#)

Einen vom Kunden verwalteten Schlüssel erstellen

Sie können einen symmetrischen, vom Kunden verwalteten Schlüssel mithilfe der AWS-Managementkonsole, AWS KMS APIs oder erstellen. Um einen symmetrischen, vom Kunden verwalteten Schlüssel zu erstellen, folgen Sie den Schritten unter [Erstellen eines symmetrischen kundenverwalteten Schlüssels im AWS Key Management Service Entwicklerhandbuch](#).

Schlüsselrichtlinien steuern den Zugriff auf den vom Kunden verwalteten Schlüssel. Jeder vom Kunden verwaltete Schlüssel muss über genau eine Schlüsselrichtlinie verfügen, die aussagt, wer den Schlüssel wie verwenden kann. Wenn Sie Ihren kundenseitig verwalteten Schlüssel erstellen, können Sie eine Schlüsselrichtlinie angeben. Weitere Informationen finden Sie unter [Verwaltung des Zugriffs auf vom Kunden verwaltete Schlüssel](#) im AWS Key Management Service Entwicklerhandbuch.

AWS KMS wichtige Richtlinien für AWS HealthScribe

Wenn Sie einen Schlüssel in demselben Konto verwenden wie die IAM Rolle, die Sie `DataAccessRole` in Ihrer Anfrage [StartMedicalScribeJob](#) oder `ResourceAccessRole` in Ihrer [StartMedicalScribeStream](#) Anfrage angegeben haben, müssen Sie die Schlüsselrichtlinie nicht aktualisieren. Um Ihren vom Kunden verwalteten Schlüssel in einem anderen Konto als Ihrem

Konto `DataAccessRole` (für Transkriptionsaufträge) oder `ResourceAccessRole` (für Streaming) zu verwenden, müssen Sie der jeweiligen Rolle in der Schlüsselrichtlinie für die folgenden Aktionen vertrauen:

- [kms:Encrypt](#) – Ermöglicht die Verwendung des vom Kunden verwalteten Schlüssels zur Verschlüsselung der Daten
- [kms:Decrypt](#) – Ermöglicht die Verwendung des vom Kunden verwalteten Schlüssels zur Entschlüsselung der Daten
- [kms:DescribeKey](#)— Stellt dem Kunden verwaltete Schlüsseldetails zur Verfügung, damit AWS HealthScribe der Schlüssel validiert werden kann

Im Folgenden finden Sie ein Beispiel für eine Schlüsselrichtlinie, mit der Sie Ihren `ResourceAccessRole` kontoübergreifenden Berechtigungen zur Verwendung Ihres vom Kunden verwalteten Schlüssels für das AWS HealthScribe Streaming gewähren können. Um diese Richtlinie für Transkriptionsaufträge zu verwenden, aktualisieren Sie den `Principal`, sodass er den `DataAccessRole` ARN verwendet, und entfernen oder ändern Sie den Verschlüsselungskontext.

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": "*"
    }
]
}

```

IAM-Richtlinienberechtigungen für Zugriffsrollen

Die IAM-Richtlinie, die Ihrem DataAccessRole Konto beigefügt ist, ResourceAccessRole muss Ihnen Berechtigungen zur Durchführung der erforderlichen AWS KMS Aktionen gewähren, unabhängig davon, ob sich der vom Kunden verwaltete Schlüssel und die Rolle in denselben oder unterschiedlichen Konten befinden. Außerdem muss die Vertrauensrichtlinie der Rolle die AWS HealthScribe Erlaubnis zur Übernahme der Rolle gewähren.

Das folgende Beispiel für eine IAM-Richtlinie zeigt, wie ResourceAccessRole AWS HealthScribe Streaming-Berechtigungen erteilt werden. Um diese Richtlinie für Transkriptionsaufträge zu verwenden, `transcribe.streaming.amazonaws.com` ersetzen Sie den Verschlüsselungskontext durch `transcribe.amazonaws.com` und entfernen oder ändern Sie ihn.

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"
        }
      }
    }
  ]
}
```

Im Folgenden finden Sie ein Beispiel für eine Vertrauensrichtlinie für `ResourceAccessRole`. Für `DataAccessRole`, ersetzen Sie `transcribe.streaming.amazonaws.com` durch `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:*"
        }
      }
    }
  ]
}
```

Weitere Informationen zur [Angabe von Berechtigungen in einer Richtlinie](#) oder zur [Problembehandlung beim Schlüsselzugriff](#) finden Sie im AWS Key Management Service Entwicklerhandbuch.

Angabe eines vom Kunden verwalteten Schlüssels für AWS HealthScribe

Sie können einen vom Kunden verwalteten Schlüssel als zweite Verschlüsselungsebene für Transkriptionsaufträge oder Streaming angeben.

- Bei Transkriptionsaufträgen geben Sie Ihren Schlüssel in Ihrem [OutputEncryptionKMSKeyIdStartMedicalScribeJob](#) API-Vorgang an.
- Für Streaming geben Sie den Schlüssel [MedicalScribeEncryptionSettings](#) in Ihrem [MedicalScribeConfigurationEvent](#) an.

AWS KMS Verschlüsselungskontext

AWS KMS Der Verschlüsselungskontext ist eine Zuordnung von nicht geheimen Schlüssel-/Wert-Paaren im Klartext. Diese Zuordnung stellt zusätzliche authentifizierte Daten dar, die als Verschlüsselungskontextpaare bezeichnet werden und eine zusätzliche Sicherheitsebene für Ihre Daten bieten. AWS HealthScribe erfordert einen symmetrischen Verschlüsselungsschlüssel, um die AWS HealthScribe Ausgabe in einen vom Kunden angegebenen Bucket zu verschlüsseln. Amazon S3 Weitere Informationen finden Sie unter [Asymmetrische Schlüssel in AWS KMS](#).

Wenn Sie Ihre Verschlüsselungskontextpaare erstellen, sollten Sie keine sensiblen Informationen einschließen. Der Verschlüsselungskontext ist nicht geheim — er ist im Klartext in Ihren CloudTrail Protokollen sichtbar (sodass Sie ihn verwenden können, um Ihre kryptografischen Operationen zu identifizieren und zu kategorisieren). Ihr Verschlüsselungskontextpaar kann Sonderzeichen enthalten, z. B. Unterstriche (), Bindestriche (-), Schrägstriche (/ , \) und Doppelpunkte (:).

Tip

Es kann nützlich sein, die Werte in Ihrem Verschlüsselungskontextpaar mit den zu verschlüsselnden Daten in Beziehung zu setzen. Obwohl dies nicht erforderlich ist, empfehlen wir Ihnen, nicht sensible Metadaten zu verwenden, die sich auf Ihre verschlüsselten Inhalte beziehen, wie z. B. Dateinamen, Header-Werte oder unverschlüsselte Datenbankfelder. Um die Ausgabeverschlüsselung mit der API zu verwenden, setzen Sie den [KMSEncryptionContext](#)-Parameter im Vorgang [StartMedicalScribeJob](#). Um den Verschlüsselungskontext für den Ausgabeverschlüsselungsvorgang bereitzustellen, muss der [OutputEncryptionKMSKeyId](#)Parameter auf eine symmetrische AWS KMS Schlüssel-ID verweisen.

Für das Streaming geben Sie die Schlüssel-Wert-Paare für die `KmsEncryptionContext` [MedicalScribeEncryptionSettings](#) in Ihrem [MedicalScribeConfigurationEvent](#) an.

Sie können [AWS KMS Bedingungsschlüssel](#) mit IAM Richtlinien verwenden, um den Zugriff auf einen symmetrischen AWS KMS Verschlüsselungsschlüssel auf der Grundlage des Verschlüsselungskontextes zu steuern, der in der Anforderung für einen [kryptografischen](#) Vorgang verwendet wurde. Ein Beispiel für eine Verschlüsselungskontextrichtlinie finden Sie unter [AWS KMS -Verschlüsselungskontextrichtlinie](#).

Die Verwendung des Verschlüsselungskontexts ist optional, wird aber empfohlen. Weitere Informationen finden Sie unter [Verschlüsselungskontext](#).

AWS HealthScribe Verschlüsselungskontext

AWS HealthScribe verwendet bei allen AWS Key Management Service kryptografischen Vorgängen denselben Verschlüsselungskontext. Der Verschlüsselungskontext ist eine Zuordnung von Zeichenfolge zu Zeichenfolge, die beliebig angepasst werden kann.

```
"encryptionContext": {  
  "ECKey": "ECValue"  
  ...  
}
```

Für AWS HealthScribe Streams ist der folgende Standardkontext der vom Dienst generierte Verschlüsselungskontext. Dieser Kontext wird zusätzlich zu jedem von Ihnen bereitgestellten Verschlüsselungskontext angewendet.

```
"encryptionContext": {  
  "aws:<region>:transcribe:medical-scribe:session-id":  
    "1234abcd-12ab-34cd-56ef-123456SAMPLE"  
}
```

Für AWS HealthScribe Transkriptionsaufträge ist der folgende Standardkontext der vom Dienst generierte Verschlüsselungskontext. Dieser Kontext wird zusätzlich zu jedem von Ihnen angegebenen Verschlüsselungskontext angewendet.

```
"encryptionContext": {  
  "aws:<region>:transcribe:medical-scribe:job-name": "<job-name>",  
  "aws:<region>:transcribe:medical-scribe:start-time-epoch-ms": "<job-start-time>"  
}
```

Wenn Sie keinen Verschlüsselungskontext angeben, wird nur der vom Dienst generierte Verschlüsselungskontext für alle AWS KMS kryptografischen Operationen verwendet.

Überwachung AWS HealthScribe mit Verschlüsselungskontext

Wenn Sie einen symmetrischen, vom Kunden verwalteten Schlüssel verwenden, um Ihre gespeicherten Daten zu verschlüsseln AWS HealthScribe, können Sie den Verschlüsselungskontext auch in Prüfaufzeichnungen und Protokollen verwenden, um zu ermitteln, wie der vom Kunden verwaltete Schlüssel verwendet wird. Der Verschlüsselungskontext erscheint auch in Protokollen, die von AWS CloudTrail oder CloudWatch Logs generiert wurden.

Verwendung des Verschlüsselungskontextes zur Steuerung des Zugriffs auf den vom Kunden verwalteten Schlüssel

Sie können den Verschlüsselungskontext in Schlüsselrichtlinien und IAM-Richtlinien als Bedingungen verwenden, um den Zugriff auf Ihren symmetrischen, kundenseitig verwalteten Schlüssel zu kontrollieren.

Im Folgenden finden Sie Beispiele für Schlüsselrichtlinienanweisungen zur Gewährung des Zugriffs auf einen kundenseitig verwalteten Schlüssel für einen bestimmten Verschlüsselungskontext. Die Bedingung in dieser Richtlinienanweisung erfordert, dass für die Verwendung von KMS-Schlüsseln eine Einschränkung des Verschlüsselungskontextes gilt, die den Verschlüsselungskontext spezifiziert.

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:ECKey": "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"
  }
]
}

```

Überwachen Sie Ihre Verschlüsselungsschlüssel für AWS HealthScribe

Wenn Sie einen vom AWS Key Management Service Kunden verwalteten Schlüssel mit verwenden AWS HealthScribe, können Sie AWS CloudTrail Or CloudWatch Logs verwenden, um Anfragen nachzuverfolgen, die AWS HealthScribe an gesendet AWS KMS werden.

Bei den folgenden Beispielen handelt es sich um Ereignisse vom Typ CloudTrail Verschlüsseln und Entschlüsseln, mit denen Sie überwachen können, wie Ihr vom Kunden verwalteter Schlüssel AWS HealthScribe verwendet wird.

Encrypt

```

{
  "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"
}

```

Decrypt

```

{
  "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"  
}
```

Dokumenthistorie für Amazon Transcribe

- Letzte Aktualisierung der Dokumentation: 13. November 2023

In der folgenden Tabelle werden wichtige Änderungen in den einzelnen Versionen von Amazon Transcribe beschrieben. Um Benachrichtigungen über Aktualisierungen dieser Dokumentation zu erhalten, können Sie einen RSS-Feed abonnieren.

Änderung	Beschreibung	Datum
Feature-Update	Die PII-Schwärzung für Batch-Transkriptionsaufträge unterstützt jetzt zusätzliche Entitätstypen: AGE AWS_ACCESS_KEY_ID AWS_SECRET_KEY CA_HEALTH_CARE_NUMBER CA_SOCIAL_SECURITY_NUMBER DATE_TIME_DRIVER_ID INTERNATIONAL_BANK_ACCOUNT_NUMBER IP_ADDRESS LICENSE_PLATE MAC_ADDRESS PASSPORT_NUMBER PASSWORD SWIFT_CODE URLUS_INDIVIDUAL_TAX_IDENTIFICATION_NUMBER USERNAME und VEHICLE_IDENTIFICATION_NUMBER	22. Mai 2026
Feature-Update	Die Bearbeitung personenbezogener Daten für Batch-	1. Mai 2026

Transkriptionsaufträge unterstützt jetzt weitere Sprachen: Französisch (fr-FR), kanadisches Französisch (fr-CA), Deutsch (de-DE), Schweizerdeutsch (de-CH), Italienisch (it-IT), Portugiesisch (pt-BR) und brasilianisches Portugiesisch (pt-BR).

Neues Feature

AWS HealthScribe unterstützt jetzt neue Vorlagen für die Zusammenfassung klinischer Notizen: BIRP, SIRP, DAP, BH_SOAP und PH_SOAP. [Weitere Informationen finden Sie in der Datei mit der klinischen Dokumentation.AWS HealthScribe](#)

30. Mai 2025

Neues Feature

AWS HealthScribe unterstützt jetzt eine GIRPP-Vorlage für die Zusammenfassung klinischer Notizen. Weitere Informationen finden Sie in der [Datei mit der AWS HealthScribe klinischen Dokumentation](#).

4. Februar 2025

Neues Feature

AWS HealthScribe unterstützt jetzt Streaming für die Transkription von medizinischen Gesprächen in Echtzeit.

29. Januar 2025

Abschnitt-Update

Aktualisierung der Ar-SA-Sprachunterstützung für die Frequenz 8000.

16. Januar 2025

Abschnitt-Update	Aktualisierung der Ar-SA-Sprachunterstützung für die Frequenz 8000.	16. Januar 2025
Abschnitt-Update	Aktualisierung des Bereichs „Unterstützte Sprache“ mit nicht unterstütztem Sprachcode, den wir in den Regionen AWS GovCloud (US) (US-West, us-gov-west-1), (, us-gov-east-1) oder US-East Afrika AWS GovCloud (US) (Cape Town, af-south-1) nicht unterstützen.	14. Januar 2025
Abschnitt-Update	Erweitern Sie die JSON-Ausgabe von Transcribe Batch um ein neues Feld namens „audio_segments“.	15. Juli 2024
Feature-Update	Aktualisiere die maximale Anzahl an Lautsprechern in der Diarisierung auf 30 statt auf 10.	10. Mai 2024
Abschnitt-Update	Die generative Anrufzusammenfassung wurde aktualisiert und es wurden Details zur Fehlerausgabe hinzugefügt.	30. April 2024
Abschnitt-Update	Aktualisierungen der Spalten mit benutzerdefiniertem Wortschatz — IPA und SoundsLike	30. April 2024

Feature-Update	Amazon Transcribe Call Analytics unterstützt jetzt die generative Anrufzusammenfassung.	29. November 2023
Abschnitt-Update	Aktualisierung des neuen Ausgabeformats mit PII-Schwärzung und Sprachenidentifizierung.	13. November 2023
Feature-Update	Die Diarisierung kann nun mit der Kanalidentifizierung kombiniert werden.	6. März 2023
Feature-Update	Die Kanalidentifizierung kann nun mit der Diarisierung kombiniert werden.	6. März 2023
Abschnitt-Update	IAM Die bewährten Verfahren wurden aktualisiert.	13. Februar 2023
Neue Sprachen	Amazon Transcribe unterstützt jetzt Vietnamesisch und Schwedisch.	6. Dezember 2022
Neues Feature	Amazon Transcribe unterstützt jetzt Anrufanalysen in Echtzeit.	28. November 2022
Feature-Update	Streaming-Schwärzung und -Identifizierung ist jetzt auch in Hindi und Thai verfügbar.	11. November 2022
Abschnitt-Update	Für die Schwärzung und Identifizierung von Datenströmen sind neue PII-Kategorien verfügbar.	14. September 2022

Abschnitt-Update	Der Abschnitt über das benutzerdefinierte Sprachmodell wurde überarbeitet.	18. Juni 2022
Abschnitt-Update	Die Batch-Sprachidentifizierung kann nun mehrere Sprachen pro Audiodatei identifizieren.	31. Mai 2022
Leitfaden-Update	Die Amazon Transcribe API-Referenz ist jetzt ein eigenständiger Leitfaden.	1. April 2022
Neues Kapitel	Eine neue Vergleichstabelle für Amazon Transcribe Medical Analytics und Amazon Transcribe Call Analytics ist enthalten.	21. März 2022
Neues Kapitel	Ein neues Kapitel mit SDK-Codebeispielen wurde aufgenommen.	21. März 2022
Feature-Update	Call Analytics bietet jetzt eine Anrufzusammenfassung.	21. März 2022
Kapitel-Update	Im Einführungskapitel werden nun Amazon Transcribe Anwendungsfälle vorgestellt.	21. März 2022
Kapitel-Update	Das Kapitel „Erste Schritte“ wurde aktualisiert und ist nun methodenspezifisch.	21. März 2022
Kapitel-Update	Das Kapitel Streaming wurde aktualisiert und umstrukturiert.	21. März 2022

Feature-Update	Die Sprachidentifizierung unterstützt jetzt benutzerdefinierte Vokabulare und benutzerdefinierte Wortschatzfilter mit Streaming-Transkriptionen.	11. März 2022
Neues Ereignis	Es gibt einen neuen Ereignistyp: Vokabularereignisse.	7. Februar 2022
Abschnitt-Update	Der Bereich der benutzerdefinierten Vokabulare wurde aktualisiert.	20. Januar 2022
Neues Feature	Die Sprachidentifizierung kann jetzt auch bei Streaming-Transkriptionen verwendet werden.	23. November 2021
Neues Feature	Die Sprachidentifizierung kann jetzt mit benutzerdefinierten Sprachmodellen, benutzerdefinierten Vokabularen, Wortschatzfilterung und Inhaltsredaktion verwendet werden.	29. Oktober 2021
Neues Feature	Amazon Transcribe unterstützt jetzt benutzerdefinierte Sprachmodelle mit Streaming-Transkriptionen.	20. Oktober 2021
Neues Feature	Amazon Transcribe kann jetzt Untertitel für Ihre Videodateien generieren.	16. September 2021

Neues Feature	Amazon Transcribe unterstützt jetzt die Schwärzung und Identifizierung personenbezogener Daten für das Streaming.	14. September 2021
Neues Feature	Amazon Transcribe unterstützt jetzt den AWS KMS Verschlüsselungskontext für ein zusätzliches Maß an Sicherheit für Ihre AWS-Konto Ressourcen.	10 September 2021
Neue Sprachen	Amazon Transcribe unterstützt jetzt Afrikaans, Dänisch, Mandarin-Chinesisch (traditionell), Thailändisches, neuseeländisches Englisch und südafrikanisches Englisch.	26. August 2021
Neues Feature	Amazon Transcribe unterstützt jetzt Ressourcen-Tagging.	24. August 2021
Neues Feature	Amazon Transcribe unterstützt jetzt Call Analytics für Batch-Transkriptionsaufträge.	4. August 2021
Neues Feature	Amazon Transcribe unterstützt jetzt die Verwendung von benutzerdefinierten Vokabularen mit benutzerdefinierten Batch-Sprachmodellen.	12. Mai 2021
Neues Feature	Amazon Transcribe unterstützt jetzt die teilweise Stabilisierung der Ergebnisse bei der Streaming-Transkription.	11. Mai 2021

Neues Feature	Amazon Transcribe unterstützt jetzt australisches Englisch, britisches Englisch, Hindi und US-Spanisch für benutzerdefinierte Sprachmodelle.	19. März 2021
Neues Feature	Amazon Transcribe unterstützt OGG/OPUS jetzt auch FLAC-Codecs für die Streaming-Audiotranskription.	24. November 2020
Neue Sprachen	Amazon Transcribe fügt Unterstützung für Italienisch und Deutsch für die Streaming-Audiotranskription hinzu.	4. November 2020
AWS-Region Erweiterung	Amazon Transcribe ist jetzt in Frankfurt (eu-central-1) und London (eu-west-2) verfügbar.	4. November 2020
Neues Feature	Amazon Transcribe fügt Unterstützung für VPC-Schnittstellen-Endpunkte bei der Batch-Transkription hinzu.	9. Oktober 2020
Neues Feature	Amazon Transcribe fügt Unterstützung für die Kanalidentifikation beim Streaming hinzu.	17. September 2020
Neues Feature	Amazon Transcribe fügt Unterstützung für die automatische Sprachidentifikation bei der Batch-Transkription hinzu.	15. September 2020

Neues Feature	Amazon Transcribe fügt Unterstützung für die Lautsprecherpartitionierung beim Streaming hinzu.	19. August 2020
Neues Feature	Amazon Transcribe fügt Unterstützung für benutzerdefinierte Sprachmodelle hinzu.	5. August 2020
Neues Feature	Amazon Transcribe fügt Unterstützung für VPC-Schnittstellen-Endpunkte beim Streaming hinzu.	26. Juni 2020
Neues Feature	Amazon Transcribe fügt Unterstützung für die Vokabelfilterung beim Streaming hinzu.	20. Mai 2020
Neues Feature	Amazon Transcribe fügt Unterstützung für das automatische Redigieren personenbezogener Daten hinzu.	26. Februar 2020
Neues Feature	Amazon Transcribe fügt Unterstützung für die Erstellung eines benutzerdefinierten Wortschatzes hinzu, das aus einer Transkription herausgefiltert werden soll.	20. Dezember 2019
Neues Feature	Amazon Transcribe fügt Unterstützung für das Einreihen von Transkriptionsaufträgen in die Warteschlange hinzu.	19. Dezember 2019

Neue Sprachen	Amazon Transcribe fügt Unterstützung für Golfarabisch, Hebräisch, Japanisch, Malaiisch, Schweizerdeutsch, Telugu und Türkisch hinzu.	21. November 2019
AWS-Region Erweiterung	Amazon Transcribe ist jetzt im asiatisch-pazifischen Raum (Tokio) (ap-northeast-1) erhältlich.	21. November 2019
Neues Feature	Amazon Transcribe fügt Unterstützung für alternative Transkriptionen hinzu.	20. November 2019
Neue Sprachen	Amazon Transcribe fügt Unterstützung für Niederländisch, Farsi, Indonesisch, Irisches Englisch, Portugiesisch, Schottisches Englisch, Tamil und Walisisches Englisch hinzu.	12. November 2019
Neue Sprache	Amazon Transcribe unterstützt jetzt Streaming-Transkription für australisches Englisch (en-AU).	25. Oktober 2019
AWS-Region Erweiterung	Amazon Transcribe ist jetzt in China (Peking) (cn-north-1) und China (Ningxia) (cn-northwest-1) erhältlich.	9. Oktober 2019

Neues Feature	Amazon Transcribe ermöglicht es Ihnen, Ihre eigenen Dateien zur Verschlüsselung Ihrer Transkriptionsausgabedateien bereitzustellen. KMS key Weitere Informationen finden Sie im OutputEncryptionKMSKeyId API-Parameter. StartStreamTranscription	24. September 2019
Neue Sprachen	Amazon Transcribe fügt Unterstützung für Chinesisch (Mandarin), vereinfachtes Chinesisch, Festlandchina und Russisch hinzu.	23. August 2019
Neues Feature	Amazon Transcribe fügt Unterstützung für Streaming-Audiotranskription mithilfe des WebSocket Protokolls hinzu.	19. Juli 2019
Neues Feature	AWS CloudTrail zeichnet jetzt Ereignisse für die StartStreamTranscription API auf.	19. Juli 2019
AWS-Region Erweiterung	Amazon Transcribe ist jetzt in den USA West (Nordkalifornien) (us-west-1) verfügbar.	27. Juni 2019
Neue Sprache	Amazon Transcribe fügt Unterstützung für Modernes Standardarabisch hinzu.	28. Mai 2019
Neues Feature	Amazon Transcribe transkribiert jetzt numerische Wörter in Zahlen für US-Englisch. Beispiel: "zweiundvierzig" wird als "42" transkribiert.	23. Mai 2019

Neue Sprache	Amazon Transcribe fügt Unterstützung für Hindi und indisches Englisch hinzu.	15. Mai 2019
Neues SDK	Das AWS SDK for C++ unterstützt jetzt Amazon Transcribe.	8. Mai 2019
Neue Sprache	Amazon Transcribe fügt Unterstützung für Spanisch hinzu.	19. April 2019
AWS-Region Erweiterung	Amazon Transcribe ist jetzt in der EU (Frankfurt) (eu-central-1) und im asiatisch-pazifischen Raum (Seoul) (ap-northeast-2) verfügbar.	18. April 2019
Neue Sprache	Amazon Transcribe fügt Unterstützung für Streaming-Transkription in britischem Englisch, Französisch und kanadischem Französisch hinzu.	5. April 2019
Neues Feature	Das AWS SDK for Ruby V3 unterstützt jetzt Amazon Transcribe	25. März 2019
Neues Feature	Amazon Transcribe ermöglicht benutzerdefinierte Vokabulare, d. h. Listen mit bestimmten Wörtern, die Sie in Ihrer Audioeingabe wiedererkennen Amazon Transcribe möchten.	25. März 2019

Neue Sprachen	Amazon Transcribe fügt Unterstützung für Deutsch und Koreanisch hinzu.	22. März 2019
Neue Sprache	Amazon Transcribe unterstützt jetzt Streaming-Transkription für US-Spanisch (es-US).	7. Februar 2019
AWS-Region Erweiterung	Amazon Transcribe ist jetzt in Südamerika (São Paulo) (sa-east-1) erhältlich.	7. Februar 2019
AWS-Region Erweiterung	Amazon Transcribe ist jetzt in den Ländern Asien-Pazifik (Mumbai) (ap-south-1), Asien-Pazifik (Singapur) (ap-south-east-1), EU (London) (eu-west-2) und EU (Paris) (eu-west3) erhältlich.	24. Januar 2019
Neue Sprachen	Amazon Transcribe fügt Unterstützung für Französisch, Italienisch und brasilianisches Portugiesisch hinzu.	20. Dezember 2018
Neues Feature	Amazon Transcribe unterstützt jetzt die Transkription von Audiostreams.	19. November 2018
Neue Sprachen	Amazon Transcribe fügt Unterstützung für australisches Englisch, britisches Englisch und kanadisches Französisch hinzu.	15. November 2018

AWS-Region Erweiterung	Amazon Transcribe ist jetzt in Kanada (Zentral) (ca-central-1) und im asiatisch-pazifischen Raum (Sydney) (ap-south-east-2) erhältlich.	17. Juli 2018
Neues Feature	Sie können nun zum Speichern der Ausgabe eines Transkriptionsauftrags Ihren eigenen Standort angeben.	11. Juli 2018
Neues Feature	Hinzugefügt und integriert. AWS CloudTrail Amazon CloudWatch Events	28. Juni 2018
Neues Feature	Amazon Transcribe fügt Unterstützung für benutzerdefinierte Vokabulare hinzu.	4. April 2018
Neues Handbuch	Dies ist die erste Version des Amazon Transcribe -Entwicklerhandbuchs.	29. November 2017

AWS Glossar

Die neueste AWS Terminologie finden Sie im [AWS Glossar](#) in der AWS-Glossar Referenz.

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.