



Entwicklerhandbuch

# Amazon Transcribe



# Amazon Transcribe: Entwicklerhandbuch

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

Die Handelsmarken und 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
Verfügbarkeit und Kontingente in den Regionen .....	2
Verfügbare Features .....	4
Unterstützte Sprachen .....	7
Unterstützte Programmiersprachen .....	16
Zeichensätze .....	17
Abchasisch .....	20
Afrikaans .....	23
Arabisch .....	23
Asturisch .....	24
Aserbaidshaisch .....	25
Armenisch .....	26
Baschkirisch .....	27
Baskisch .....	29
Belarussisch .....	30
Bengalisch .....	31
Bosnisch .....	33
Bulgarisch .....	34
Katalanisch .....	35
Zentralkurdisch .....	35
Chinesisch – vereinfacht .....	37
Chinesisch – traditionell .....	38
Kroatisch .....	39
Tschechisch .....	39
Dänisch .....	40
Niederländisch .....	41
Englisch .....	42
Estnisch .....	42
Farsi .....	43
Finnisch .....	44
Französisch .....	45
Galizisch .....	46

---

Georgisch .....	46
Deutsch .....	48
Griechisch .....	48
Gujarati .....	50
Hausa .....	52
Hebräisch .....	52
Hindi .....	53
Ungarisch .....	55
Isländisch .....	56
Indonesisch .....	57
Italienisch .....	57
Japanisch .....	58
Kabylich .....	58
Kannada .....	59
Kasachisch .....	61
Kinyarwanda .....	62
Koreanisch .....	63
Kirgisisch .....	63
Lettisch .....	66
Litauisch .....	66
Luganda .....	67
Mazedonisch .....	67
Malaiisch .....	70
Malayalam .....	70
Maltesisch .....	72
Marathi .....	73
Meadow Mari .....	75
Mongolisch .....	77
Norwegisches Bokmål .....	80
Odia/Oriya .....	80
Paschto .....	82
Polnisch .....	84
Portugiesisch .....	85
Pandschabi .....	86
Rumänisch .....	88
Russisch .....	89

Serbisch .....	90
Singhalesisch .....	92
Slowakisch .....	94
Slowenisch .....	95
Somali .....	96
Spanisch .....	96
Sundanesisch .....	97
Swahili .....	98
Schwedisch .....	98
Tagalog/Filipino .....	99
Tamil .....	99
Tatarisch .....	101
Telugu .....	103
Thailändisch .....	105
Türkisch .....	107
Ukrainisch .....	108
Uigurisch .....	109
Usbekisch .....	112
Vietnamesisch .....	113
Walisisch .....	117
Wolof .....	118
Zulu .....	119
Funktionsweise .....	120
Ein- und Ausgabe von Daten .....	120
Medienformate .....	121
Audiokanäle .....	122
Stichprobenraten .....	123
Ausgabe .....	123
Zahlen transkribieren .....	126
Erste Schritte .....	132
Anmeldung fürAWS-Konto .....	133
Installation derAWS CLI und SDKs .....	133
IAMAnmeldeinformationen konfigurieren .....	134
EinenAmazon S3 Bucket erstellen .....	134
Erstellen einer IAM-Richtlinie .....	135
Transkribieren mit demAWS Management Console .....	137

Transkribieren mit dem AWS CLI .....	147
Einen neuen Transkriptionsjob starten .....	147
Abrufen des den Status eines Abtrags .....	149
Deine Transkriptionsjobs auflisten .....	150
Löschen Ihres Transkriptionsauftrags .....	150
Transkribieren mit den SDKs AWS .....	151
Mit AWS SDKs arbeiten .....	163
Transkribieren mit HTTP oder WebSockets .....	164
Streaming-Transkriptionen .....	167
Bewährte Methoden .....	168
Streaming und Teilergebnisse .....	169
Stabilisierung von Teilergebnissen .....	170
Einrichten einer Streaming-Transkription .....	175
Ereignis-Stream-Kodierung .....	189
Datenframes .....	192
Auftragswarteschlangen Warteschlange tesch .....	193
Aktivieren von Auftrags Warteschlange tesch Warteschlange tesch .....	193
Markieren von Ressourcen .....	199
Tag-basierte Zugriffskontrolle .....	200
Hinzufügen von Stichwörtern zu Ihren Amazon Transcribe Ressourcen .....	201
Sprecherpartitionierung (Diarisierung) .....	205
Aufteilung der Sprecher in einer Batch-Transkription .....	206
Partitionierung der Sprecher in einer Streaming-Transkription .....	209
Beispielausgabe .....	212
Transkription von Mehrkanal-Audio .....	218
Verwendung der Kanalidentifikation bei einer Batch-Transkription .....	219
Verwendung der Kanalidentifikation in einer Streaming-Transkription .....	223
Beispielausgabe .....	224
Identifizierung von Sprachen .....	232
Batch-Sprachidentifizierung .....	232
Identifizierung von Sprachen in mehrsprachigen Audiodateien .....	233
Verbesserung der Genauigkeit der Sprachidentifizierung .....	234
Kombination von Sprachidentifizierung mit anderen Amazon Transcribe -Features .....	235
Sprachidentifizierung mit Batch-Transkriptionen .....	236
Identifizierung von Streaming-Sprachen .....	243
Identifizierung von Sprachen in mehrsprachigen Audiodateien .....	244

Sprachidentifizierung mit Streaming-Medien nutzen .....	245
Alternative Transkriptionen .....	252
Alternative Transkriptionen anfordern .....	254
Verbesserung der Transkription .....	259
Benutzerdefinierte Vokabulare .....	260
Benutzerdefinierte Vokabulartabellen versus Listen .....	261
Erstellen eines benutzerdefinierten Vokabulars mithilfe einer Tabelle .....	262
Erstellen eines benutzerdefinierten Vokabulars mithilfe einer Liste .....	274
Verwenden eines benutzerdefinierten einen benutzerdefinierten Vokabular .....	277
Benutzerdefinierte Sprachmodelle .....	284
Datenquellen .....	285
Daten trainieren oder optimieren .....	285
Ein benutzerdefiniertes Sprachmodell erstellen .....	286
Verwenden eines benutzerdefinierten Sprachmodells .....	293
Wörter filtern .....	301
Einen Vokabelfilter erstellen .....	302
Benutzerdefinierte Vokabelfilter erstellen .....	303
Einen benutzerdefinierten Vokabelfilter verwenden .....	308
Verwendung eines benutzerdefinierten Vokabelfilters in einer Batch-Transkription .....	278
Verwenden eines benutzerdefinierten Vokabelfilters in einer Streaming-Transkription .....	281
Erkennung toxischer Sprache .....	318
Verwendung der Erkennung toxischer Sprache .....	319
Verwendung der Erkennung toxischer Sprache in einer Batch-Transkription .....	319
Beispielausgabe .....	324
Transkripte redigieren .....	326
Schwärzen von PII in Ihrem Batch-Job .....	327
Schwärzen oder Identifizieren von PII in einem Echtzeit-Datenstrom .....	334
Beispielausgabe .....	340
Beispiel für eine redigierte Ausgabe (Batch) .....	340
Beispiel für eine redigierte Streaming-Ausgabe .....	343
Beispiel für eine PII-Identifikationsausgabe .....	344
Erstellen von Untertitel .....	347
Generieren von Untertiteldateien .....	348
Analyse von Callcenter-Audio .....	352
Häufige Anwendungsfälle .....	352
Überlegungen und zusätzliche Informationen .....	354

Regionale Verfügbarkeit und Quotas .....	355
Analyse nach Anrufen .....	356
Anruferkenntnisse nach dem Gespräch .....	357
Kategorien erstellen .....	360
Beginn einer Transkription .....	372
Ausgabe für Analyse nach Anrufen .....	382
Aktivieren der generativen Anrufzusammenfassung .....	395
Echtzeit-Call-Analytics .....	400
Einblicke in Echtzeit .....	401
Kategorien erstellen .....	403
Analyse nach Anrufen mit Echtzeit-Transkriptionen .....	412
Beginn einer Transkription .....	419
Echtzeit-Call-Analytics-Ausgabe .....	428
Transkribieren von Amazon Chime Anrufen .....	434
Codebeispiele .....	436
Aktionen .....	437
CreateVocabulary .....	438
DeleteMedicalTranscriptionJob .....	441
DeleteTranscriptionJob .....	444
DeleteVocabulary .....	448
GetTranscriptionJob .....	450
GetVocabulary .....	453
ListMedicalTranscriptionJobs .....	456
ListTranscriptionJobs .....	461
ListVocabularies .....	467
StartMedicalTranscriptionJob .....	471
StartStreamTranscriptionAsync .....	483
StartTranscriptionJob .....	487
UpdateVocabulary .....	506
Szenarien .....	509
Erstellen und Verfeinern eines benutzerdefinierten Vokabulars .....	510
Audio transkribieren und Auftragsdaten abrufen .....	520
Serviceübergreifende Beispiele .....	531
Eine Amazon-Transcribe-App entwickeln .....	531
Erstellen einer Amazon-Transcribe-Streaming-App .....	532
Text in Sprache und zurück in Text konvertieren .....	533

Sicherheit .....	535
Identitäts- und Zugriffsverwaltung .....	536
Zielgruppe .....	536
Authentifizierung mit Identitäten .....	537
Verwalten des Zugriffs mit Richtlinien .....	541
Wie Amazon Transcribe funktioniert mit IAM .....	544
Confused-Deputy-Prävention .....	552
Beispiele für identitätsbasierte Richtlinien .....	553
Fehlerbehebung .....	562
Datenschutz .....	564
Datenschutz für den Datenverkehr zwischen Netzwerken .....	565
Datenverschlüsselung .....	566
Abmeldung von der Verwendung Ihrer Daten zur Serviceverbesserung .....	569
Überwachung Amazon Transcribe .....	569
Überwachung mit CloudWatch .....	570
Überwachung Amazon Transcribe mit CloudTrail .....	571
Verwenden Amazon EventBridge mit Amazon Transcribe .....	575
Compliance-Validierung .....	583
Ausfallsicherheit .....	584
Sicherheit der Infrastruktur .....	584
Schwachstellenanalyse und -management .....	585
VPC-Endpunkte (AWS PrivateLink) .....	585
Gemeinsame Subnetze .....	587
Bewährte Methoden für die Gewährleistung der Sicherheit .....	588
Amazon Transcribe Medical .....	590
Verfügbarkeit und Kontingente in der Region .....	591
Medizinische Fachgebiete .....	593
Transkription medizinischer Begriffe und Messungen .....	594
Zahlen transkribieren .....	596
Transkription eines medizinischen Gesprächs .....	598
Eine Audiodatei transkribieren .....	599
Transkribieren eines Echtzeit-Streams .....	604
Aktivieren der Lautsprecherpartitionierung .....	607
Transkription von Mehrkanal-Audio .....	618
Ein medizinisches Diktat transkribieren .....	625
Eine Audiodatei transkribieren .....	626

Transkription eines medizinischen Streaming-Diktats .....	631
Erstellung und Verwendung von benutzerdefinierten medizinischen Vokabeln .....	634
Erstellen einer Textdatei für Ihr medizinisches benutzerdefiniertes Vokabular .....	635
Verwenden einer Textdatei, um ein medizinisches benutzerdefiniertes Vokabular zu erstellen .....	640
Transkription einer Audiodatei mit einem medizinischen benutzerdefinierten Vokabular .....	642
Transkription eines Echtzeit-Streams mit einem medizinischen benutzerdefinierten Vokabular .....	644
Zeichensatz für Amazon Transcribe Medical .....	647
PHI in einem Transkript identifizieren .....	648
PHI in einer Audiodatei identifizieren .....	650
Identifizierung von PHI in einem Echtzeit-Stream .....	654
Generierung alternativer Transkriptionen .....	656
VPC-Endpunkte (AWS PrivateLink) .....	659
Überlegungen für Amazon Transcribe VPC-Endpunkte für den medizinischen Bereich .....	659
Erstellen eines VPC-Schnittstellen-Endpunkts für Amazon Transcribe Medizinisch .....	660
Erstellen einer VPC-Endpunktrichtlinie für Amazon Transcribe Medizinisches Streaming .....	660
Gemeinsame Subnetze .....	662
AWS-HealthScribe .....	663
Transkriptionsdatei .....	665
Datei mit klinischer Dokumentation .....	666
Start eines AWS-HealthScribe-Auftrags .....	667
Beispielausgabe .....	669
Datenverschlüsselung im Ruhezustand für AWS HealthScribe .....	682
Einen kundenverwalteten Schlüssel erstellen .....	683
Angaben eines vom Kunden verwalteten Schlüssels für AWS HealthScribe .....	685
AWS KMS-Verschlüsselungskontext .....	685
Dokumentverlauf .....	687
AWS-Glossar .....	699
.....	dcc

# Was ist Amazon Transcribe?

Amazon Transcribe ist ein automatischer Spracherkennungsdienst, der maschinelle Lernmodelle verwendet, um Audio in Text umzuwandeln. Sie können Amazon Transcribe es als eigenständigen Transkriptionsdienst verwenden oder um speech-to-text Funktionen zu einer beliebigen Anwendung hinzuzufügen.

Mit Amazon Transcribe können Sie die Genauigkeit für Ihren speziellen Anwendungsfall durch Sprachanpassung verbessern, Inhalte filtern, um den Datenschutz für Kunden 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 Sie können Mediendateien transkribieren, die sich in einem Amazon S3 Bucket befinden (Stapel). In der [Unterstützte Sprachen und sprachspezifische Funktionen](#) Tabelle finden Sie, welche Sprachen für die einzelnen Transkriptionstypen unterstützt werden.

## Themen

- [Amazon Transcribe und HIPAA-Eignung](#)
- [Preisgestaltung](#)
- [Verfügbarkeit und Kontingente in den Regionen](#)

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

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

### Tip

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

## Amazon Transcribe und HIPAA-Eignung

Amazon Transcribe fällt unter AWS die HIPAA-Zulassung und BAA, nach der BAA-Kunden alle gespeicherten und übertragenen PHI verschlüsseln müssen, wenn sie verwendet werden. Die

automatische PHI-Identifizierung ist ohne Aufpreis und in allen Regionen, in denen Amazon Transcribe tätig ist, verfügbar. Weitere Informationen finden Sie unter [HIPAA-Eignung und BAA](#).

## Preisgestaltung

Amazon Transcribe ist ein pay-as-you-go Service. Die Preisgestaltung basiert auf Sekunden transkribierter Audiowiedergabe, die monatlich abgerechnet wird.

Die Nutzung wird pro Sekunde berechnet, wobei pro Sekunde berechnet wird, wobei mindestens für 15 Sekunden zu zahlen ist. Beachten Sie, dass zusätzliche Gebühren für Funktionen wie das Redigieren von PII-Inhalten und benutzerdefinierte Sprachmodelle anfallen.

Informationen zu den jeweiligen AWS-Region Kosten finden Sie unter [Amazon Transcribe Preisgestaltung](#).

## Verfügbarkeit und Kontingente in den Regionen

Amazon Transcribe wird im Folgenden unterstützt AWS-Regionen:

Region	Art der Transkription
af-south-1 (Kapstadt)	Stapeinstellungen, Stream-Stapeinstellungen
ap-east-1 (Hongkong)	Batch
ap-northeast-1 (Tokyo)	Stapeinstellungen, Stream-Stapeinstellungen
ap-northeast-2 (Seoul)	Stapeinstellungen, Stream-Stapeinstellungen
ap-south-1 (Mumbai)	Stapeinstellungen, Stream-Stapeinstellungen
ap-southeast-1 (Singapur)	Stapeinstellungen, Stream-Stapeinstellungen
ap-southeast-2 (Sydney)	Stapeinstellungen, Stream-Stapeinstellungen
ca-central-1 (Kanada, Zentral)	Stapeinstellungen, Stream-Stapeinstellungen
eu-central-1 (Frankfurt)	Stapeinstellungen, Stream-Stapeinstellungen
eu-north-1 (Stockholm)	Batch

Region	Art der Transkription
eu-west-1 (Irland)	Stapeinstellungen, Stream-Stapeinstellungen
eu-west-2 (Londres)	Stapeinstellungen, Stream-Stapeinstellungen
eu-west-3 (Paris)	Batch
me-south-1 (Bahrain)	Batch
sa-east-1 (São Paulo)	Stapeinstellungen, Stream-Stapeinstellungen
us-east-1 (Nordkalifornien)	Stapeinstellungen, Stream-Stapeinstellungen
us-east-2 (Ohio)	Stapeinstellungen, Stream-Stapeinstellungen
us-gov-east-1 (GovCloud, US-Ost)	Stapeinstellungen, Stream-Stapeinstellungen
us-gov-west-1 (GovCloud, US-West)	Stapeinstellungen, Stream-Stapeinstellungen
us-west-1 (San Francisco)	Batch
us-west-2 (Oregon)	Stapeinstellungen, Stream-Stapeinstellungen

 **Important**

Die regionale Unterstützung unterscheidet sich für Amazon Transcribe [Amazon Transcribe Medical](#), und [Call Analytics](#).

Informationen zu den Endpunkten für jede unterstützte Region finden Sie unter [Dienstendpunkte](#) in der AWSAllgemeinen Referenz.

Eine Liste der Kontingente, die sich auf Ihre Transkriptionen beziehen, finden Sie in den [Servicekontingenten](#) in der Allgemeinen Referenz. AWS Einige Kontingente können auf Anfrage geändert werden. Wenn die Spalte „Einstellbar“ den Wert „Ja“ enthält, können Sie eine Erhöhung beantragen. Wählen Sie dafür den bereitgestellten Link aus.

# 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.

Funktion	Amazon Transcribe	<a href="#">Amazon Transcribe Medical</a> <sup>1</sup>	<a href="#">Amazon Transcribe Rufen Sie Analytics an</a>
----------	-------------------	--	--

## Konfigurationsoptionen

<a href="#">Alternative Transkriptionen</a>	Batch, Streaming	Batch, Streaming	Nein
<a href="#">Kanalidentifizierung</a>	Batch, Streaming	Batch, Streaming	post-call, real-time
<a href="#">Auftragswarteschlangen</a>	Batch	Nein	post-call
<a href="#">Sprachidentifizierung</a>	Batch, Streaming	Nein	post-call
<a href="#">Mehrsprachige-Identifizierung</a>	Batch, Streaming	Nein	Nein
<a href="#">Sprecherdiarisierung</a>	Batch, Streaming	Batch, Streaming	post-call
<a href="#">Transkribieren von Ziffern</a> <sup>2</sup>	Batch, Streaming	Batch, Streaming	post-call, real-time

## Gesprächsanalytik

<a href="#">Anrufmerkmale</a>	Nein	Nein	post-call
<a href="#">Anrufzusammenfassung</a> <sup>2</sup>	Nein	Nein	post-call

Funktion	Amazon Transcribe	<a href="#">Amazon Transcribe Medical</a> <sup>1</sup>	<a href="#">Amazon Transcribe Rufen Sie Analytics an</a>
<a href="#">Benutzerdefinierte Kategorisierung</a>	Nein	Nein	post-call
<a href="#">Kategorie-Ereignisse in Echtzeit</a>	Nein	Nein	real-time
<a href="#">Erkennung von Problemen in Echtzeit</a> <sup>2</sup>	Nein	Nein	real-time
<a href="#">Echtzeit-Sprechers timmung</a>	Nein	Nein	real-time
<a href="#">Sprecherstimmung</a>	Nein	Nein	post-call
Sprachanpassung			
<a href="#">Benutzerdefinierte Sprachmodelle</a> <sup>2</sup>	Batch, Streaming	Nein	post-call, real-time
<a href="#">Benutzerdefinierte Vokabulare</a>	Batch, Streaming	Batch, Streaming	post-call, real-time
Organisation der Ressourcen			
<a href="#">Tagging</a>	Batch	Batch	post-call
Sensible Daten			
<a href="#">Identifizierung persönlicher Gesundheitsinformationen</a> <sup>2</sup>	Nein	Batch, Streaming	Nein

Funktion	Amazon Transcribe	<a href="#">Amazon Transcribe Medical</a> <sup>1</sup>	<a href="#">Amazon Transcribe Rufen Sie Analytics an</a>
<a href="#">Identifizierung persönlich identifizierbarer Informationen</a> <sup>2</sup>	Streamen	Nein	real-time
<a href="#">Audio schwärzen</a> <sup>2</sup>	Nein	Nein	post-call, real-time
<a href="#">Schwärzung von Transkripten</a> <sup>2</sup>	Batch, Streaming	Nein	post-call, real-time
<a href="#">Wortschatzfilter</a>	Batch, Streaming	Nein	post-call, real-time
Video			
<a href="#">Untertitel</a>	Batch	Nein	Nein

 <sup>1</sup> Amazon Transcribe Medical ist nur in US-Englisch verfügbar.

<sup>2</sup> 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	<a href="#">Dateneingabe</a>	<a href="#">Zahlen transkribieren</a>	<a href="#">Akronyme</a>	<a href="#">Benutzerdefinierte Sprachmodelle</a>	<a href="#">Schwärzen</a>	<a href="#">Call Analytics</a> * -
<a href="#">Abchasisch</a>	ab-GE	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Afrikaans</a>	af-ZA	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Arabisch, Golf</a>	ar-AE	Batch	Nein	Nein	Nein	Nein	post-call
<a href="#">Arabisch, Moderner Standard</a>	ar-SA	Batch	Nein	Nein	Nein	Nein	Nein
<a href="#">Armenisch</a>	hy-AM	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Asturisch</a>	ast-ES	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Aserbaidschanisch</a>	az-AZ	Batch	Nein	Batch	Nein	Nein	Nein

Sprache	Sprachcode	<a href="#">Dateneingabe</a>	<a href="#">Zahlen transkribieren</a>	<a href="#">Akronyme</a>	<a href="#">Benutzerdefinierte Sprachmodelle</a>	<a href="#">Schwärzen</a>	<a href="#">Call Analytics</a> *-
<a href="#">Baschkirisch</a>	ba-RU	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Baskisch</a>	eu-ES	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Belarussisch</a>	be-BY	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Bengalisch</a>	bn-IN	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Bosnisch</a>	bs-BA	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Bulgarisch</a>	bg-BG	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Katalanisch</a>	ca-ES	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Zentralkurdisch, Iran</a>	ckb-IR	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Zentralkurdisch, Irak</a>	ckb-IQ	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Chinesisch – vereinfacht</a>	zh-CN	Batch, Streaming	Nein	Nein	Nein	Nein	post-call

Sprache	Sprachcode	<a href="#">Dateneingabe</a>	<a href="#">Zahlen transkribieren</a>	<a href="#">Akronyme</a>	<a href="#">Benutzerdefinierte Sprachmodelle</a>	<a href="#">Schwärzen</a>	<a href="#">Call Analytics</a> *-
<a href="#">Chinesisch – traditionell</a>	zh-TW	Batch	Nein	Nein	Nein	Nein	Nein
<a href="#">Kroatisch</a>	hr-HR	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Tschechisch</a>	cs-CZ	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Dänisch</a>	da-DK	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Niederländisch</a>	nl-NL	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Englisch, Australisch</a>	en-AU	Batch, Streaming	Batch, Streaming	Batch, Streaming	Batch, Streaming	Streamen	post-call, real-time
<a href="#">Englisch, Britisch</a>	en-GB	Batch, Streaming	Batch, Streaming	Batch, Streaming	Batch, Streaming	Streamen	post-call, real-time
<a href="#">Englisch, Indisch</a>	en-IN	Batch	Batch	Batch	Nein	Nein	post-call
<a href="#">Englisch, Irisch</a>	en-IE	Batch	Batch	Batch	Nein	Nein	post-call
<a href="#">Englisch, Neuseeland</a>	en-NZ	Batch	Batch	Batch	Nein	Nein	Nein

Sprache	Sprachcode	<a href="#">Dateneingabe</a>	<a href="#">Zahlen transkribieren</a>	<a href="#">Akronyme</a>	<a href="#">Benutzerdefinierte Sprachmodelle</a>	<a href="#">Schwärzen</a>	<a href="#">Call Analytics</a> *-
<a href="#">Englisch</a> , Schottisch	en-AB	Batch	Batch	Batch	Nein	Nein	post-call
<a href="#">Englisch</a> , Südafrika	en-ZA	Batch	Batch	Batch	Nein	Nein	Nein
<a href="#">Englisch</a> , US	en-US	Batch, Streaming	Batch, Streaming	Batch, Streaming	Batch, Streaming	Batch, Streaming	post-call, real-time
<a href="#">Englisch</a> , Walisisch	en-WL	Batch	Batch	Batch	Nein	Nein	post-call
<a href="#">Estnisch</a>	et-ET	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Farsi</a>	fa-IR	Batch	Nein	Nein	Nein	Nein	Nein
<a href="#">Finnisch</a>	fi-FI	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Französisch</a>	fr-FR	Batch, Streaming	Nein	Batch, Streaming	Nein	Nein	post-call, real-time
<a href="#">Französisch</a> , Kanadisch	fr-CA	Batch, Streaming	Nein	Batch, Streaming	Nein	Nein	post-call, real-time
<a href="#">Galizisch</a>	gl-ES	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Georgisch</a>	ka-GE	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Deutsch</a>	de-DE	Batch, Streaming	Batch, Streaming	Batch, Streaming	Batch, Streaming	Nein	post-call, real-time

Sprache	Sprachcode	<a href="#">Dateneingabe</a>	<a href="#">Zahlen transkribieren</a>	<a href="#">Akronyme</a>	<a href="#">Benutzerdefinierte Sprachmodelle</a>	<a href="#">Schwärzen</a>	<a href="#">Call Analytics</a> *-
<a href="#">Deutsch</a> , Schweizerisch	de-CH	Batch	Batch	Batch	Nein	Nein	post-call
<a href="#">Griechisch</a>	el-GR	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Gujarati</a>	gu-IN	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Hausa</a>	ha-NG	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Hebräisch</a>	he-IL	Batch	Nein	Nein	Nein	Nein	Nein
<a href="#">Hindi</a> , Indisch	hi-IN	Batch, Streaming	Nein	Batch, Streaming	Batch	Nein	post-call
<a href="#">Ungarisch</a>	hu-HU	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Isländisch</a>	is-IS	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Indonesisch</a>	id-ID	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Italienisch</a>	it-IT	Batch, Streaming	Nein	Batch, Streaming	Nein	Nein	post-call, real-time
<a href="#">Japanisch</a>	ja-JP	Batch, Streaming	Nein	Nein	Batch, Streaming	Nein	post-call
<a href="#">Kabylich</a>	kab-DZ	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Kannada</a>	kn-IN	Batch	Nein	Batch	Nein	Nein	Nein

<a href="#">Sprache</a>	<a href="#">Sprachcode</a>	<a href="#">Dateneingabe</a>	<a href="#">Zahlen transkribieren</a>	<a href="#">Akronyme</a>	<a href="#">Benutzerdefinierte Sprachmodelle</a>	<a href="#">Schwärzen</a>	<a href="#">Call Analytics</a> *-
<a href="#">Kasachisch</a>	kk-KZ	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Kinyarwanda</a>	rw-RW	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Koreanisch</a>	ko-KR	Batch, Streaming	Nein	Nein	Nein	Nein	post-call
<a href="#">Kirgisisch</a>	ky-KG	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Lettisch</a>	lv-LV	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Litauisch</a>	lt-LT	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Luganda</a>	lg-IN	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Mazedonisch</a>	mk-MK	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Malaiisch</a>	ms-MY	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Malayalam</a>	ml-IN	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Maltesisch</a>	mt-MT	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Marathi</a>	mr-IN	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Meadow Mari</a>	mhr-RU	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Mongolisch</a>	mn-MN	Batch	Nein	Batch	Nein	Nein	Nein

<a href="#">Sprache</a>	<a href="#">Sprachcode</a>	<a href="#">Dateneingabe</a>	<a href="#">Zahlen transkribieren</a>	<a href="#">Akronyme</a>	<a href="#">Benutzerdefinierte Sprachmodelle</a>	<a href="#">Schwärzen</a>	<a href="#">Call Analytics</a> *-
<a href="#">Norwegisches Bokmål</a>	no-NO	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Odia/Oriya</a>	or-IN	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Paschto</a>	ps-AF	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Polnisch</a>	pl-PL	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Portugiesisch</a>	pt-PT	Batch	Nein	Batch	Nein	Nein	post-call
<a href="#">Portugiesisch, Brasilianisch</a>	pt-BR	Batch, Streaming	Nein	Batch, Streaming	Nein	Nein	post-call, real-time
<a href="#">Pandschabi</a>	pa-IN	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Rumänisch</a>	ro-RO	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Russisch</a>	ru-RU	Batch	Nein	Nein	Nein	Nein	Nein
<a href="#">Serbisch</a>	sr-RS	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Singhalesisch</a>	si-LK	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Slowakisch</a>	sk-SK	Batch	Nein	Batch	Nein	Nein	Nein

Sprache	Sprachcode	<a href="#">Dateneingabe</a>	<a href="#">Zahlen transkribieren</a>	<a href="#">Akronyme</a>	<a href="#">Benutzerdefinierte Sprachmodelle</a>	<a href="#">Schwärzen</a>	<a href="#">Call Analytics</a> * -
<a href="#">Slowenisch</a>	sl-SI	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Somali</a>	so-SO	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Spanisch</a>	es-ES	Batch	Nein	Batch	Nein	Nein	post-call
<a href="#">Spanisch, US</a>	es-US	Batch, Streaming	Nein	Batch, Streaming	Batch, Streaming	Batch, Streaming	post-call, real-time
<a href="#">Sundanesisch</a>	su-ID	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Swahili, Kenia</a>	sw-KE	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Swahili, Burundi</a>	sw-BI	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Swahili, Ruanda</a>	sw-RW	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Swahili, Tansania</a>	sw-TZ	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Swahili, Uganda</a>	sw-UG	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Schwedisch</a>	sv-SE	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Tagalog/Filipino</a>	tl-PH	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Tamil</a>	ta-IN	Batch	Nein	Nein	Nein	Nein	Nein

<a href="#">Sprache</a>	<a href="#">Sprachcode</a>	<a href="#">Dateneingabe</a>	<a href="#">Zahlen transkribieren</a>	<a href="#">Akronyme</a>	<a href="#">Benutzerdefinierte Sprachmodelle</a>	<a href="#">Schwärzen</a>	<a href="#">Call Analytics</a> *-
<a href="#">Tatarisch</a>	tt-RU	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Telugu</a>	te-IN	Batch	Nein	Nein	Nein	Nein	Nein
<a href="#">Thailändisch</a>	th-TH	Batch, Streaming	Nein	Batch, Streaming	Nein	Nein	Nein
<a href="#">Türkisch</a>	tr-TR	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Ukrainisch</a>	uk-UA	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Uigurisch</a>	ug-CN	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Usbekisch</a>	uz-UZ	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Vietnamesisch</a>	vi-VN	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Walisisch</a>	cy-WL	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Wolof</a>	wo-SN	Batch	Nein	Batch	Nein	Nein	Nein
<a href="#">Zulu</a>	zu-ZA	Batch	Nein	Batch	Nein	Nein	Nein

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

- [Anruferzusammenfassung](#): 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
<a href="#">.NET</a>	.NET wird für Streaming nicht unterstützt.
<a href="#">AWS Befehlszeilenschnittstelle (CLI)</a>	Die CLI wird für das Streaming nicht unterstützt.
<a href="#">C++</a>	<a href="#">C++</a>
<a href="#">Go</a>	<a href="#">Go</a>
<a href="#">Java V2</a>	<a href="#">Java V2</a>
<a href="#">JavaScript</a>	<a href="#">JavaScript V3</a>
<a href="#">PHP V3</a>	<a href="#">PHP V3</a>
<a href="#">Python Boto3</a>	<a href="#">Python-Streaming-SDK für Amazon Transcribe</a>
<a href="#">Rubin V3</a>	<a href="#">Rubin V3</a>
<a href="#">Rust</a>	<a href="#">Rust</a>

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

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 Sprache, die Amazon Transcribe unterstützt, gibt es einen bestimmten Satz von Zeichen, die Amazon Transcribe erkennen kann. 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](#)
- [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
a	0430	лъ	0459
б	0431	ь	045A

Zeichen	Code	Zeichen	Code
в	0432	ħ	045B
г	0433	ќ	045C
д	0434	#	045D
е	0435	ŷ	045E
ж	0436	ц	045F
з	0437	ѓ	0491
и	0438	ƒ	0493
й	0439	жҕ	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	œ	04A9
с	0441	ç	04AB
т	0442	т	04AD
у	0443	ү	04AF
ф	0444	ұ	04B1
х	0445	х	04B3

Zeichen	Code	Zeichen	Code
ц	0446	ц	04B5
ч	0447	ч	04B7
ш	0448	h	04BB
щ	0449	є	04BD
ъ	044A	є	04BF
ы	044B	#	04CA
ь	044C	ă	04D1
э	044D	ă	04D3
ю	044E	ě	04D7
я	044F	ə	04D9
#	0450	з	04E1
ë	0451	й	04E3
ђ	0452	ö	04E7
í	0453	ө	04E9
є	0454	ÿ	04EF
s	0455	ÿ	04F1
i	0456	ÿ	04F3
ï	0457	#	04F7
j	0458	Ы	04F9
#	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
ع	0621	س	0633
آ	0622	ش	0634

Zeichen	Code	Zeichen	Code
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		

## Aserbaidsschanischer Zeichensatz

Für ein aserbaidsschanisches 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
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
ա	0561	օ	0574
բ	0562	լ	0575
գ	0563	հ	0576
դ	0564	Շ	0577
ե	0565	ն	0578
զ	0566	Ն	0579
է	0567	ոյ	057A
ը	0568	ջ	057B
թ	0569	ւ	057C
ժ	056A	ս	057D
ի	056B	վ	057E
լ	056C	տ	057F
խ	056D	ր	0580
ծ	056E	գ	0581

Zeichen	Code	Zeichen	Code
l̡	056F	ɫ	0582
h	0570	ɥ	0583
ɑ	0571	f	0584
ɯ	0572	o	0585
ɸ	0573	ɕ	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	Ӏ	0459
б	0431	ӑ	045A
в	0432	ӓ	045B
г	0433	ӕ	045C
д	0434	ӗ	045D
е	0435	ә	045E
ж	0436	ӛ	045F

Zeichen	Code	Zeichen	Code
з	0437	ғ	0491
и	0438	ҕ	0493
й	0439	җ	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	с	04AB
т	0442	т	04AD
у	0443	ү	04AF
ф	0444	ү	04B1
х	0445	х	04B3
ц	0446	ц	04B5
ч	0447	ч	04B7
ш	0448	һ	04BB
щ	0449	е	04BD
ъ	044A	е	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	ÿ	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	с	0441
б	0431	т	0442
в	0432	у	0443
г	0433	ф	0444
д	0434	х	0445
е	0435	ц	0446
ж	0436	ч	0447

Zeichen	Code	Zeichen	Code
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

Zeichen	Code	Zeichen	Code
অ	0985	প	09AA
আ	0986	ফ	09AB
ঈ	0987	ব	09AC
ই	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
চ	099A	†	09C0
ছ	099B	~	09C1
জ	099C	~	09C2
ঝ	099D	~	09C3

Zeichen	Code	Zeichen	Code
ঞ	099E	<	09C4
ট	099F	ৗ	09C7
ঠ	09A0	ঠ	09C8
ড	09A1	ঢ	09CB
ঢ	09A2	ঢ়	09CC
ণ	09A3	,	09CD
ত	09A4	#	09CE
থ	09A5	৳	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
й	0439	ш	0448
к	043A	щ	0449
л	043B	ъ	044A
м	043C	ь	044C
н	043D	ю	044E

Zeichen	Code	Zeichen	Code
o	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
ı	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	ف	06A4
ر	0631	ک	06A9
ز	0632	گ	06AF
س	0633	ژ	06B5
ش	0634	ھ	06BE
ع	0639	ۆ	06C6
غ	063A	ؤ	06C7
ف	0641	ى	06CC
ق	0642	ئ	06CE
ل	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 stellt die vier Töne im Chinesischen (vereinfacht) durch Zahlen dar. 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 stellt die vier Töne im Chinesischen (traditionell) durch Zahlen dar. Die folgende Tabelle zeigt, wie die Tonzeichen für das Wort ㄇㄩ̇ ㄩˊ zugeordnet werden.

Ton	Tonmarkierung
Ton 1	ㄇㄩ̇
Ton 2	ㄇㄩˊ
Ton 3	ㄇㄩˇ
Ton 4	ㄇㄩˋ

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

## 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	đ	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
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

Zeichen	Code	Zeichen	Code
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	ظ	0638
آ	0622	ع	0639
أ	0623	غ	063A
ؤ	0624	ف	0641
ئ	0626	ق	0642
ا	0627	ل	0644
ب	0628	م	0645
ت	062A	ن	0646
ث	062B	ه	0647
ج	062C	و	0648
ح	062D	ـ	064E
خ	062E	ـ	064F
د	062F	ـ	0650

Zeichen	Code	Zeichen	Code
ذ	0630	س	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
Î	00CE	î	00EE
Ï	00CF	ï	00EF
Ô	00D4	ô	00F4
Ö	00D6	ö	00F6
Ù	00D9	ù	00F9

Zeichen	Code	Zeichen	Code
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	ბ	10E0
ბ	10D1	გ	10E1
გ	10D2	დ	10E2
დ	10D3	ე	10E3
ე	10D4	ვ	10E4
ვ	10D5	ზ	10E5
ზ	10D6	ყ	10E6
ყ	10D7	შ	10E7
შ	10D8	ჩ	10E8
ჩ	10D9	ც	10E9
ც	10DA	ძ	10EA
ძ	10DB	წ	10EB
წ	10DC	ჭ	10EC
ჭ	10DD	ხ	10ED
ხ	10DE	ბ	10EE
ბ	10DF	ჯ	10EF
ჯ	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
ά	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	ï	03CA
κ	03BA	ü	03CB
λ	03BB	ó	03CC
μ	03BC	ώ	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
આ	0A86	ફ	0AAB
ઇ	0A87	બ	0AAC
ઈ	0A88	ભ	0AAD
ઉ	0A89	મ	0AAE
ઊ	0A8A	ય	0AAF
ઋ	0A8B	ર	0AB0
ઌ	0A8D	લ	0AB2
એ	0A8F	ળ	0AB3
ઐ	0A90	વ	0AB5
ઔ	0A91	શ	0AB6

Zeichen	Code	Zeichen	Code
ઓ	0A93	૫	0AB7
ઔ	0A94	સ	0AB8
ક	0A95	હ	0AB9
ખ	0A96	.	0ABC
ગ	0A97	૧	0ABE
ઘ	0A98	૨	0ABF
ઙ	0A99	૩	0AC0
ચ	0A9A	૪	0AC1
છ	0A9B	૫	0AC2
જ	0A9C	૬	0AC3
ઝ	0A9D	૭	0AC5
ઞ	0A9E	૮	0AC7
ટ	0A9F	૯	0AC8
ઠ	0AA0	૧૦	0AC9
ડ	0AA1	૧૧	0ACB
ઢ	0AA2	૧૨	0ACC
ણ	0AA3	૧૩	0ACD
ત	0AA4	૩૦	0AD0
થ	0AA5	૩૧	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	ḅ	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
κ	05D0	◌	05DE
⸀	05D1	◌	05DF
λ	05D2	◌	05E0
τ	05D3	◌	05E1
◌	05D4	◌	05E2

Zeichen	Code	Zeichen	Code
א	05D5	ך	05E3
ב	05D6	פ	05E4
ג	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
ॆ	0901	ध	0927
ॆ	0902	न	0928
:	0903	प	092A
अ	0905	फ	092B
आ	0906	ब	092C
इ	0907	भ	092D

Zeichen	Code	Zeichen	Code
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	ठ	093F
ग	0917	ड	0940
घ	0918	ढ	0941
ङ	0919	ण	0942
च	091A	त	0943
छ	091B	थ	0945
ज	091C	द	0947
झ	091D	न	0948
ञ	091E	प	0949
ट	091F	फ	094B

Zeichen	Code	Zeichen	Code
ठ	0920	ौ	094C
ड	0921	्	094D
ढ	0922	ज़	095B
ण	0923	ड़	095C
त	0924	ढ	095D

Amazon Transcribe ordnet die folgenden Zeichen zu:

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

## 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	õ	0151
ǿ	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

Zeichen	Code	Zeichen	Code
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

Zeichen	Code	Zeichen	Code
č	010D	ħ	1E25
ř	0159	ŗ	1E5B
ǧ	01E7	ş	1E63
ε	025B	ţ	1E6D
γ	0263	ẓ	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
೦	0C82	೧	0CA7
೨	0C83	೨	0CA8
೩	0C85	೩	0CAA
೪	0C86	೪	0CAB
೫	0C87	೫	0CAC
೬	0C88	೬	0CAD
೭	0C89	೭	0CAE

Zeichen	Code	Zeichen	Code
ಊ	0C8A	ಯ	0CAF
ಋ	0C8B	ರ	0CB0
ಎ	0C8E	ಲ	0CB2
ಏ	0C8F	ಳ	0CB3
ಐ	0C90	ಱ	0CB5
ಒ	0C92	ಱ	0CB6
ಓ	0C93	ಱ	0CB7
ಔ	0C94	ಱ	0CB8
ಕ	0C95	ಱ	0CB9
ಖ	0C96	#	0CBC
ಗ	0C97	#	0CBD
ಘ	0C98	ಁ	0CBE
ಙ	0C99	ಂ	0CBF
ಚ	0C9A	ಃ	0CC0
ಛ	0C9B	಄	0CC1
ಜ	0C9C	ಅ	0CC2
ಝ	0C9D	ಆ	0CC3
ಞ	0C9E	ಇ	0CC6
ಟ	0C9F	ಈ	0CC7
ಠ	0CA0	ಉ	0CC8

Zeichen	Code	Zeichen	Code
Ә	0CA1	Қ	0CCA
Ғ	0CA2	Қа	0CCB
Ң	0CA3	Ғ	0CCC
Ң	0CA4	Ғ	0CCD
Ә	0CA5	Ғ	0CD5
Ғ	0CA6	Ғ	0CD6
Ә	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
т	0442	ы	044B
б	0431	я	044F
о	043E	с	0441
п	043F	h	04BB
ш	0448	д	0434

Zeichen	Code	Zeichen	Code
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 ein koreanisches benutzerdefiniertes Vokabular können Sie jede der Hangeul-Silben im Feld `Phrase` verwenden. Weitere Informationen finden Sie unter [Hangeul-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	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	ж	0497
к	043A	з	0499
л	043B	ќ	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	т	04AD

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

Zeichen	Code	Zeichen	Code
ī	0457	#	04F7
j	0458	ī	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
ģ	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	ı	012F
č	010D	š	0161
ę	0119	ı	0173
è	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	η	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	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	ж	0497
к	043A	ѕ	0499
л	043B	ќ	049B
м	043C	к	049F
н	043D	т	04A1
о	043E	џ	04A3
п	043F	џ	04A5

Zeichen	Code	Zeichen	Code
ρ	0440	ϱ	04A9
с	0441	ç	04AB
т	0442	ṭ	04AD
у	0443	γ	04AF
φ	0444	ϣ	04B1
х	0445	χ	04B3
ц	0446	ц	04B5
ч	0447	ч	04B7
ш	0448	h	04BB
щ	0449	є	04BD
ъ	044A	є	04BF
ы	044B	#	04CA
ь	044C	ă	04D1
э	044D	ä	04D3
ю	044E	ě	04D7
я	044F	ə	04D9
#	0450	з	04E1
ë	0451	ñ	04E3
ђ	0452	ö	04E7
ѓ	0453	ø	04E9

Zeichen	Code	Zeichen	Code
€	0454	ȳ	04EF
s	0455	ÿ	04F1
i	0456	ÿ	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
◌	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	ഠ	0D31
എ	0D0E	ല	0D32
ഏ	0D0F	ള	0D33
ഐ	0D10	ഴ	0D34
ഒ	0D12	വ	0D35
ഓ	0D13	ശ	0D36
ഔ	0D14	ഷ	0D37
ക	0D15	സ	0D38
ഖ	0D16	ഹ	0D39
ഗ	0D17	ഘ	0D3E
ഘ	0D18	ങ	0D3F
ങ	0D19	ച	0D40

Zeichen	Code	Zeichen	Code
ⵏ	0D1A	ⵐ	0D41
ⵐ	0D1B	ⵎ	0D42
ⵑ	0D1C	ⵏ	0D43
ⵒ	0D1D	ⵐ	0D46
ⵓ	0D1E	ⵑ	0D47
ⵔ	0D1F	ⵒ	0D48
ⵕ	0D20	ⵓ	0D4A
ⵖ	0D21	ⵔ	0D4B
ⵗ	0D22	ⵕ	0D4C
ⵘ	0D23	ⵖ	0D4D
ⵙ	0D24	#	0D7A
ⵚ	0D25	#	0D7B
ⵛ	0D26	#	0D7C
ⵜ	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
ž	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
ॆ	0901	थ	0925
ॆ	0902	द	0926
:	0903	ध	0927
अ	0905	न	0928
आ	0906	प	092A
इ	0907	फ	092B

Zeichen	Code	Zeichen	Code
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
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

Zeichen	Code	Zeichen	Code
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	љ	0459
б	0431	њ	045A
в	0432	ћ	045B
г	0433	ќ	045C
д	0434	#	045D

Zeichen	Code	Zeichen	Code
е	0435	ÿ	045E
ж	0436	ц	045F
з	0437	г	0491
и	0438	ғ	0493
й	0439	ж	0497
к	043A	з	0499
л	043B	қ	049B
м	043C	к	049F
н	043D	т	04A1
о	043E	ң	04A3
п	043F	н	04A5
р	0440	ъ	04A9
с	0441	с	04AB
т	0442	т	04AD
у	0443	ү	04AF
ф	0444	ұ	04B1
х	0445	х	04B3
ц	0446	ц	04B5
ч	0447	ч	04B7
ш	0448	h	04BB

Zeichen	Code	Zeichen	Code
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
ѓ	0453	ө	04E9
є	0454	ÿ	04EF
s	0455	ÿ	04F1
i	0456	ÿ	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	ћ	045B
г	0433	ќ	045C
д	0434	#	045D
е	0435	ђ	045E
ж	0436	џ	045F
з	0437	ѓ	0491
и	0438	Ѡ	0493
й	0439	ж	0497
к	043A	з	0499
л	043B	қ	049B
м	043C	к	049F
н	043D	т	04A1
о	043E	ң	04A3
п	043F	т	04A5
р	0440	ѡ	04A9
с	0441	ѣ	04AB

Zeichen	Code	Zeichen	Code
т	0442	ᠲ	04AD
у	0443	ᠮ	04AF
ф	0444	ᠮ	04B1
х	0445	ᠬ	04B3
ц	0446	ᠴ	04B5
ч	0447	ᠴ	04B7
ш	0448	ᠰ	04BB
щ	0449	ᠰ	04BD
ъ	044A	ᠰ	04BF
ы	044B	#	04CA
ь	044C	ᠠ	04D1
э	044D	ᠠ	04D3
ю	044E	ᠡ	04D7
я	044F	ᠡ	04D9
#	0450	ᠰ	04E1
ë	0451	ᠠ	04E3
ñ	0452	ᠣ	04E7
í	0453	ᠡ	04E9
ε	0454	ᠡ	04EF
s	0455	ᠡ	04F1

Zeichen	Code	Zeichen	Code
i	0456	ÿ	04F3
ï	0457	#	04F7
j	0458	ÿi	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 ein benutzerdefiniertes Odia-/Oriya-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
ୂ	0B01	୩	0B26
୦	0B02	୪	0B27
୧	0B03	୫	0B28
ଅ	0B05	ଢ	0B2A
ଆ	0B06	ଢ଼	0B2B
ଇ	0B07	ଢ଼	0B2C
ଈ	0B08	ଢ଼	0B2D
ଉ	0B09	ଢ଼	0B2E
ଊ	0B0A	ଢ଼	0B2F
ଋ	0B0B	ର	0B30
ଏ	0B0F	ଲ	0B32
ଏଫ	0B10	ଳ	0B33
ଓ	0B13	ଶ	0B36
ଓଫ	0B14	ଷ	0B37
କ	0B15	ସ	0B38
ଖ	0B16	ହ	0B39
ଗ	0B17	.	0B3C
ଘ	0B18	।	0B3E
ଙ	0B19	୿	0B3F
ଚ	0B1A	୧	0B40

Zeichen	Code	Zeichen	Code
ḥ	0B1B	ḥ	0B41
ḡ	0B1C	ḡ	0B42
ḏ	0B1D	ḏ	0B43
ḙ	0B1E	ḙ	0B47
ḟ	0B1F	ḟ	0B48
Ḑ	0B20	Ḑ#	0B4B
ḑ	0B21	ḑ	0B4C
Ḓ	0B22	Ḓ	0B4D
ḓ	0B23	ḓ	0B56
Ḕ	0B24	Ḕ	0B5F
ḕ	0B25	ḕ	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
آ	0622	و	0648
أ	0623	ې	064A
ؤ	0624	="	064B
ئ	0626	ۀ	064C
ا	0627	="	064D
ب	0628	ـ	064E
ت	062A	ۀ	064F
ث	062B	ـ	0650
ج	062C	ۀ	0651
ح	062D	°	0652
خ	062E	#	0654
د	062F	ـ	0670
ذ	0630	ډ	067C
ر	0631	ړ	067E
ز	0632	ځ	0681
س	0633	څ	0685
ش	0634	چ	0686
ص	0635	ډ	0689
ض	0636	ړ	0693
ط	0637	ړ	0696

Zeichen	Code	Zeichen	Code
ظ	0638	ژ	0698
ع	0639	بن	069A
غ	063A	ک	06A9
ف	0641	گ	06AB
ق	0642	گ	06AF
ل	0644	ن	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

Zeichen	Code	Zeichen	Code
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

Zeichen	Code	Zeichen	Code
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	ੳ	0A27
ਆ	0A06	ਠ	0A28
ੲ	0A07	ੲ	0A2A

Zeichen	Code	Zeichen	Code
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
ਙ	0A19	ਾ	0A3E
ਚ	0A1A	ਿ	0A3F
ਛ	0A1B	ੀ	0A40
ਜ	0A1C	ੁ	0A41
ਝ	0A1D	ੂ	0A42
ਞ	0A1E	ੇ	0A47
ਟ	0A1F	ੈ	0A48
ਠ	0A20	ੋ	0A4B
ਡ	0A21	ਾ	0A4C

Zeichen	Code	Zeichen	Code
ੴ	0A22	ੴ	0A4D
ੴ	0A23	ੴ	0A5C
ੴ	0A24	ੴ	0A70
ੴ	0A25	ੴ	0A71
ੴ	0A26	ੴ	0A72
ੴ	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	ș	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
и	0438	ъ	044A
й	0439	ы	044B
к	043A	ь	044C
л	043B	э	044D
м	043C	ю	044E
н	043D	я	044F
о	043E	ё	0451

## 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	i	0456
č	010D	ï	0457
đ	0111	j	0458
š	0161	љ	0459
ž	017E	њ	045A
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

Zeichen	Code	Zeichen	Code
й	0439	ќ	049B
к	043A	к	049F
л	043B	ќ	04A1
м	043C	ћ	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	џ	04B3
ф	0444	џ	04B5
х	0445	џ	04B7
ц	0446	h	04BB
ч	0447	е	04BD
ш	0448	е	04BF
щ	0449	#	04CA
ъ	044A	ă	04D1
ы	044B	ă	04D3
ь	044C	ě	04D7

Zeichen	Code	Zeichen	Code
э	044D	ө	04D9
ю	044E	з	04E1
я	044F	й	04E3
#	0450	ö	04E7
ë	0451	е	04E9
ђ	0452	ÿ	04EF
í	0453	ÿ	04F1
є	0454	ÿ	04F3
s	0455	#	04F7
Ы	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

Zeichen	Code	Zeichen	Code
#	0D85	#	0DB1
#	0D86	#	0DB3
#	0D87	#	0DB4
#	0D88	#	0DB5
#	0D89	#	0DB6
#	0D8A	#	0DB7
#	0D8B	#	0DB8
#	0D8C	#	0DB9
#	0D8D	#	0DBA
#	0D91	#	0DBB
#	0D92	#	0DBD
#	0D93	#	0DC0
#	0D94	#	0DC1
#	0D95	#	0DC2
#	0D96	#	0DC3
#	0D9A	#	0DC4
#	0D9B	#	0DC5
#	0D9C	#	0DC6
#	0D9D	#	0DCA
#	0D9E	#	0DCF

Zeichen	Code	Zeichen	Code
#	0D9F	#	0DD0
#	0DA0	#	0DD1
#	0DA1	#	0DD2
#	0DA2	#	0DD3
#	0DA3	#	0DD4
#	0DA4	#	0DD6
#	0DA5	#	0DD8
#	0DA7	#	0DD9
#	0DA8	##	0DDA
#	0DA9	#	0ddb
#	0DAA	##	0DDC
#	0DAB	###	0DDD
#	0DAC	##	0DDE
#	0DAD	#	0DDF
#	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
d'	010F	í	0155
é	00E9	š	0161
í	00ED	t'	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
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
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 ein benutzerdefiniertes Tagalog-/Filipino-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		
ñ	00F1		

## Tamilischer Zeichensatz

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

Zeichen	Code	Zeichen	Code
அ	0B85	ஈ	0BB0
ஆ	0B86	ஊ	0BB2
இ	0B87	஋	0BB5

Zeichen	Code	Zeichen	Code
ஈ	0B88	ழ	0BB4
உ	0B89	ள	0BB3
ஊ	0B8A	ற	0BB1
எ	0B8E	ன	0BA9
ஏ	0B8F	ஐ	0B9C
ஐ	0B90	#	0BB6
ஒ	0B92	ஷ	0BB7
ஓ	0B93	ஸ	0BB8
ஔ	0B94	ஹ	0BB9
ஃ	0B83	.	0BCD
க	0B95	ஈ	0BBE
ங	0B99	ஐ	0BBF
ச	0B9A	ஃ	0BC0
ஞ	0B9E	ஃ	0BC1
ட	0B9F	ஃ	0BC2
ண	0BA3	ஃ	0BC6
த	0BA4	ஃ	0BC7
ந	0BA8	ஃ	0BC8
ப	0BAA	ஃ	0BCA
ம	0BAE	ஃ	0BCB

Zeichen	Code	Zeichen	Code
ш	0BAF	ᄀ	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	Ӏ	0459
б	0431	ӑ	045A
в	0432	ӕ	045B
г	0433	ӗ	045C
д	0434	ӝ	045D
е	0435	ӡ	045E
ж	0436	Ӣ	045F
з	0437	Ӥ	0491
и	0438	Ӧ	0493
й	0439	ӧ	0497
к	043A	Ә	0499

Zeichen	Code	Zeichen	Code
л	043B	қ	049B
м	043C	к	049F
н	043D	т	04A1
о	043E	ң	04A3
п	043F	ф	04A5
р	0440	Ә	04A9
с	0441	Ҙ	04AB
т	0442	җ	04AD
у	0443	ү	04AF
ф	0444	ұ	04B1
х	0445	х	04B3
ц	0446	ц	04B5
ч	0447	ч	04B7
ш	0448	h	04BB
щ	0449	є	04BD
ъ	044A	ё	04BF
ы	044B	#	04CA
ь	044C	ă	04D1
э	044D	ä	04D3
ю	044E	ě	04D7

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

## Telugu-Zeichensatz

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

Zeichen	Code	Zeichen	Code
-	002D	ఱ	0C24
ఁ	0C01	ఢ	0C25
ఌ	0C02	ణ	0C26
఍	0C03	డ	0C27
ఞ	0C05	ఙ	0C28
ఠ	0C06	ఞ	0C2A

Zeichen	Code	Zeichen	Code
ఇ	0C07	ఫ	0C2B
ఈ	0C08	బ	0C2C
ఉ	0C09	భ	0C2D
ఊ	0C0A	మ	0C2E
ఋ	0C0B	య	0C2F
ఠ	0C30	ఎ	0C0E
ఙ	0C31	ఏ	0C0F
ఠ	0C32	ఐ	0C10
ఠ	0C33	ఒ	0C12
ఠ	0C35	ఓ	0C13
ఠ	0C36	ఔ	0C14
ఠ	0C37	క	0C15
ఠ	0C38	ఖ	0C16
ఠ	0C39	గ	0C17
ఠ	0C3E	ఘ	0C18
ఠ	0C3F	ఙ	0C19
ఠ	0C40	చ	0C1A
ఠ	0C41	ఛ	0C1B
ఠ	0C42	జ	0C1C
ఠ	0C43	ఝ	0C1D

Zeichen	Code	Zeichen	Code
๒	0C44	๓	0C1E
๔	0C47	๕	0C1F
๖	0C48	๗	0C20
๘	0C4A	๙	0C21
๑๐	0C4B	๑๑	0C22
๑๒	0C4C	๑๓	0C23
๑๔	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

Zeichen	Code	Zeichen	Code
ง	0E07	ห	0E2B
จ	0E08	ฬ	0E2C
ฉ	0E09	อ	0E2D
ช	0E0A	ฮ	0E2E
ซ	0E0B	๓	0E2F
ฌ	0E0C	๕	0E30
ญ	0E0D	๖	0E31
ฎ	0E0E	๗	0E32
ฏ	0E0F	๘	0E34
ฐ	0E10	๙	0E35
ฑ	0E11	๐	0E36
ฒ	0E12	๑	0E37
ณ	0E13	๒	0E38
ด	0E14	๓	0E39
ต	0E15	๔	0E3A
ถ	0E16	๕	0E40
ท	0E17	๖	0E41
ธ	0E18	๗	0E42
น	0E19	๘	0E43
บ	0E1A	๙	0E44

Zeichen	Code	Zeichen	Code
๑	0E1B	๓	0E45
๒	0E1C	๔	0E46
๓	0E1D	๕	0E47
๔	0E1E	'	0E48
๕	0E1F	๖	0E49
๖	0E20	๗	0E4A
๗	0E21	๘	0E4B
๘	0E22	๙	0E4C
๙	0E23	๐	0E4D
๐	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

Zeichen	Code	Zeichen	Code
Ö	00D6	û	00FB
Ü	00DC	ü	00FC
â	00E2	Ǧ	011E
ä	00E4	ǧ	011F
ç	00E7	ı	0130
è	00E8	ı	0131
é	00E9	Ş	015E
ê	00EA	ş	015F
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
a	0430	p	0440

Zeichen	Code	Zeichen	Code
б	0431	с	0441
в	0432	т	0442
г	0433	у	0443
д	0434	ф	0444
е	0435	х	0445
ж	0436	ц	0446
з	0437	ч	0447
и	0438	ш	0448
й	0439	щ	0449
к	043A	ь	044C
л	043B	ю	044E
м	043C	я	044F
н	043D	ё	0454
о	043E	і	0456
п	043F	ï	0457
ғ	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
ء	0621	="	064B
آ	0622	ء	064C
أ	0623	="	064D
ؤ	0624	´	064E
إ	0625	ء	064F
ئ	0626	´	0650
ا	0627	ء	0651
ب	0628	°	0652
ة	0629	#	0653
ت	062A	#	0654
ث	062B	#	0657
ج	062C	´	0670
ح	062D	ط	0679
خ	062E	ن	067A
د	062F	ب	067B

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

Zeichen	Code	Zeichen	Code
м	043C	у	0443
ь	044C	ж	0436
ъ	044A	ф	0493
а	0430		

## Vietnamesischer Zeichensatz

Amazon Transcribe stellt die sechs Töne im Vietnamesischen durch Zahlen dar. 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
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

Zeichen	Code	Zeichen	Code
ă	0103	Ă	0102
đ	0111	Đ	0110
ĩ	0129	Ĩ	0128
ũ	0169	Ũ	0168
ơ	01A1	Ơ	01A0
ư	01B0	Ư	01AF
ạ	1EA1	Ạ	1EA0
à	1EA3	À	1EA2
ã	1EA5	Ã	1EA4
á	1EA7	Á	1EA6
ă	1EA9	Â	1EA8
â	1EAB	Ã	1EAA
â	1EAD	Â	1EAC
á	1EAF	Á	1EAE
ă	1EB1	À	1EB0
â	1EB3	Ã	1EB2
ã	1EB5	Ã	1EB4
ă	1EB7	Ă	1EB6
ê	1EB9	Ê	1EB8
è	1EBB	È	1EBA

Zeichen	Code	Zeichen	Code
ẽ	1EBD	ẽ	1EBC
ế	1EBF	Ế	1EBE
ề	1EC1	Ề	1EC0
ễ	1EC3	Ễ	1EC2
ễ	1EC5	Ễ	1EC4
ệ	1EC7	Ệ	1EC6
ị	1EC9	Ị	1EC8
ị	1ECB	Ị	1ECA
ọ	1ECD	Ọ	1ECC
ỏ	1ECF	Ỏ	1ECE
ố	1ED1	Ố	1ED0
ồ	1ED3	Ồ	1ED2
ỗ	1ED5	Ỗ	1ED4
ỗ	1ED7	Ỗ	1ED6
ộ	1ED9	Ộ	1ED8
ớ	1EDB	Ớ	1EDA
ờ	1EDD	Ờ	1EDC
ở	1EDF	Ở	1EDE
ỡ	1EE1	Ỡ	1EE0
ợ	1EE3	Ợ	1EE2

Zeichen	Code	Zeichen	Code
ұ	1EE5	Ў	1EE4
ů	1EE7	Ў	1EE6
ú	1EE9	Ў	1EE8
ù	1EEB	Ў	1EEA
ǔ	1EED	Ў	1EEC
ũ	1EEF	Ў	1EEE
ұ	1EF1	Ў	1EF0
ỳ	1EF3	Ỳ	1EF2
ʏ	1EF5	Ỳ	1EF4
ỷ	1EF7	Ỳ	1EF6
ỹ	1EF9	Ỳ	1EF8

## Walisischer Zeichensatz

Für ein walisisches 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

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

# Funktionsweise von Amazon Transcribe

Amazon Transcribe verwendet Modelle des maschinellen Lernens, um Sprache in Text umzuwandeln.

Zusätzlich zum transkribierten Text enthalten Transkripte Daten über den transkribierten Inhalt, einschließlich Konfidenzwerten und Zeitstempeln für jedes Wort oder jedes Satzzeichen. Ein Ausgabebeispiel finden Sie im Abschnitt [Dateneingabe und -ausgabe](#). Eine vollständige Liste der Funktionen, die Sie auf Ihre Transkription anwenden können, finden Sie in der [Funktionsübersicht](#).

Transkriptionsmethoden lassen sich in zwei Hauptkategorien einteilen:

- **Batch-Transkriptionen:** Transcribe Sie Mediendateien, die in einen Amazon S3 Bucket hochgeladen wurden. Sie können die [AWS CLI](#), [AWS Management Console](#), und verschiedene [AWS SDKs](#) für Batch-Transkriptionen verwenden.
- **Streaming-Transkriptionen:** Transcribe Sie Medienstreams in Echtzeit. Sie können das [AWS Management Console](#), [WebSocketsHTTP/2](#) und verschiedene [AWS SDKs](#) für Streaming-Transkriptionen verwenden.

Beachten Sie, dass die Funktions- und Sprachunterstützung für Batch- und Streaming-Transkriptionen unterschiedlich ist. Weitere Informationen finden Sie unter [Amazon Transcribe features](#) und [Unterstützte Sprachen](#).

Themen

- [Ein- und Ausgabe von Daten](#)
- [Zahlen und Interpunktion transkribieren](#)

## API-Operationen für den Einstieg

Batch: [StartTranscriptionJob](#)

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

## Ein- und Ausgabe von Daten

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

Wenn Sie in einem Amazon S3 Bucket gespeicherte Mediendateien transkribieren, führen Sie Batch-Transkriptionen durch. Wenn Sie Medienstreams transkribieren, führen Sie Streaming-Transkriptionen durch. Diese beiden Prozesse haben unterschiedliche Regeln und Anforderungen.

Batch-Transkriptionen können Sie verwenden, [Auftragswarteschlangen](#) [Warteschlange](#) [tesch](#) wenn Sie nicht alle Ihre Transkriptionsaufträge gleichzeitig bearbeiten müssen. So behalten Sie Amazon Transcribe den Überblick über Ihre Transkriptionsaufträge und können diese bearbeiten, wenn Plätze 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 [AWS Support](#).

## Themen

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

## Medienformate

Die unterstützten Medientypen unterscheiden sich zwischen Batch-Transkriptionen und Streaming-Transkriptionen, obwohl für beide Formate verlustfreie Formate empfohlen werden. Einzelheiten finden Sie in der folgenden Tabelle:

	Stapel	Streaming
Unterstützte Formate	<ul style="list-style-type: none"> <li>• AMR</li> <li>• FLAC</li> <li>• M4A</li> </ul>	<ul style="list-style-type: none"> <li>• FLAC</li> <li>• Ogg Opus</li> <li>• PCM-Codierung</li> </ul>

	Stapel	Streaming
	<ul style="list-style-type: none"> <li>• MP3</li> <li>• MP4</li> <li>• Ogg</li> <li>• WebM</li> <li>• WAV</li> </ul>	
Empfohlene Formate	<ul style="list-style-type: none"> <li>• FLAC</li> <li>• WAV mit PCM-16-Bit-Codierung</li> </ul>	<ul style="list-style-type: none"> <li>• FLAC</li> <li>• PCM-signiertes 16-Bit-Little-Endian-Audio (beachten Sie, dass WAV nicht enthalten ist)</li> </ul>

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

#### Note

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

## Audiokanäle

Amazon Transcribe unterstützt einkanalige und zweikanalige Medien. Medien mit mehr als zwei Kanälen werden derzeit nicht unterstützt.

Wenn Ihr Audio mehrere Lautsprecher auf einem Kanal enthält und Sie jeden Lautsprecher in Ihrer Transkriptionsausgabe partitionieren und beschriften möchten, können Sie die [Lautsprecherpartitionierung \(Diarisierung\)](#) verwenden.

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

Beide Optionen erzeugen eine Transkriptdatei.

**Note**

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

## Stichprobenraten

Bei Batch-Transkriptionsaufträgen können Sie wählen, ob Sie eine Samplerate angeben möchten, obwohl dieser Parameter optional ist. Wenn Sie es in Ihre Anfrage aufnehmen, stellen Sie sicher, dass der von Ihnen angegebene Wert mit der tatsächlichen Samplerate in Ihrem Audio übereinstimmt. Wenn Sie eine Samplerate angeben, die nicht mit Ihrem Audio übereinstimmt, kann Ihr Job fehlschlagen.

Bei Streaming-Transkriptionen müssen Sie in Ihrer Anfrage eine Samplerate angeben. Stellen Sie wie bei Batch-Transkriptionsaufträgen sicher, dass der von Ihnen angegebene Wert mit der tatsächlichen Samplerate in Ihrem Audio übereinstimmt.

Bei den Abtastraten für Audio mit niedriger Klangqualität, wie z. B. Telefonaufzeichnungen, liegt der Wert in der Regel bei 8.000 Hz. Amazon Transcribe unterstützt für Hi-Fidelity-Audio Werte zwischen 16.000 Hz und 48.000 Hz.

## Ausgabe

Die Transkriptionsausgabe erfolgt im JSON-Format. Der erste Teil Ihres Transkripts enthält das Transkript selbst in Absatzform, gefolgt von zusätzlichen Daten für jedes Wort und jedes Satzzeichen. Die bereitgestellten Daten hängen von den Funktionen 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 eine Beispielausgabe einer einfachen Transkriptionsanforderung, die keine zusätzlichen Optionen oder Funktionen enthielt.

Alle Batch-Transkripte werden in Amazon S3 Buckets gespeichert. 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 Ihr Transkript in einem Amazon S3 Bucket speichern möchten, der Ihnen gehört, geben Sie die URI des Buckets in Ihrer Transkriptionsanfrage an. Stellen Sie sicher, dass Sie Amazon Transcribe Schreibberechtigungen für diesen Bucket erteilen, bevor Sie Ihren Batch-Transkriptionsjob

starten. Wenn Sie Ihren eigenen Bucket angeben, verbleibt Ihr Transkript in diesem Bucket, bis Sie es entfernen.

Wenn Sie keinen Amazon S3 Bucket angeben, verwendet Amazon Transcribe einen sicheren, vom Service verwalteten Bucket und stellt Ihnen eine temporäre URI zur Verfügung, mit der Sie Ihr Transkript herunterladen können. Beachten Sie, dass temporäre URIs für 15 Minuten gültig sind. Wenn Sie bei der Verwendung der angegebenen URI eine `AccessDenied` Fehlermeldung erhalten, fordern Sie eine neue temporäre URI für Ihr Transkript an.

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

Streaming-Transkripte werden mit derselben 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 Analyseprotokollen nach einem Anruf und Standard-Batch-Transkripten mit aktivierter Tagebuchfunktion.

## Beispielausgabe

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

Das folgende Beispiel stammt aus einem einfachen Batch-Transkriptionsjob, der keine [zusätzlichen Funktionen beinhaltet](#). Mit jeder zusätzlichen Funktion, die Sie auf Ihre Transkriptionsanfrage anwenden, erhalten Sie zusätzliche Daten in Ihrer Transkript-Ausgabedatei.

Die grundlegenden Batch-Transkripte bestehen aus zwei Hauptabschnitten:

1. `transcripts`: Enthält das gesamte Transkript in einem Textblock.
2. `items`: Enthält Informationen zu jedem Wort und jedem Satzzeichen aus dem `transcripts` Abschnitt.

Jede zusätzliche Funktion, die 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": [
      {
        "start_time": "0.64",
        "end_time": "1.09",
        "alternatives": [
          {
            "confidence": "1.0",
            "content": "Welcome"
          }
        ],
        "type": "pronunciation"
      },
      {
        "start_time": "1.09",
        "end_time": "1.21",
        "alternatives": [
          {
            "confidence": "1.0",
            "content": "to"
          }
        ],
        "type": "pronunciation"
      },
      {
        "start_time": "1.21",
        "end_time": "1.74",
        "alternatives": [
          {
            "confidence": "1.0",
            "content": "Amazon"
          }
        ]
      }
    ]
  }
}
```

```

    ],
    "type": "pronunciation"
  },
  {
    "start_time": "1.74",
    "end_time": "2.56",
    "alternatives": [
      {
        "confidence": "1.0",
        "content": "Transcribe"
      }
    ],
    "type": "pronunciation"
  },
  {
    "alternatives": [
      {
        "confidence": "0.0",
        "content": "."
      }
    ],
    "type": "punctuation"
  }
]
},
"status": "COMPLETED"
}

```

## Zahlen und Interpunktion transkribieren

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

In den meisten Sprachen werden Zahlen in ihre Wortformen transkribiert. Wenn Ihre Medien jedoch auf Englisch oder Deutsch sind, Amazon Transcribe behandeln Sie Zahlen je nach Kontext, in dem sie verwendet werden, unterschiedlich.

Wenn ein Sprecher beispielsweise "" sagt 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, wird dies wie folgt transkribiert:

- Englische und deutsche Dialekte: 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

In der folgenden Tabelle finden Sie alle Regeln für gesprochene Zahlen in Englisch und Deutsch.

Regeln	Englische Dialekte (Audio eingeben → Text ausgeben)	Deutsche Dialekte (Audio eingeben → Text ausgeben)
Wandle Kardinalzahlen größer als zehn in Zahlen um.	<ul style="list-style-type: none"> <li>• "Fifty five" → 55</li> <li>• "a hundred" → 100</li> <li>• "One thousand and thirty one" → 1031</li> <li>• "One hundred twenty-three million four hundred fifty six thousand seven hundred eight nine" → 123,456,789</li> </ul>	<ul style="list-style-type: none"> <li>• "fünfundfünfzig" → 55</li> <li>• "vier tausend sechs hundert einundachtzig" → 4681</li> <li>• "eine Sache" → "eine Sache"</li> </ul>
Konvertieren von Kardinalzahlen gefolgt von „Million“ oder „Milliarde“ in Zahlen, gefolgt von einem Wort, wenn auf „Million“ oder „Milliarde“ keine Zahl folgt.	<ul style="list-style-type: none"> <li>• "one hundred million" → 100 million</li> <li>• "one billion" → 1 billion</li> <li>• "two point three million" → 2.3 million</li> </ul>	<ul style="list-style-type: none"> <li>• "zehn Millionen Menschen" → 10 Millionen Menschen</li> <li>• "zehn Millionen fünf hundert tausend" → 10.500.000</li> </ul>
Wandelt Ordinalzahlen größer als zehn in Zahlen um.	<ul style="list-style-type: none"> <li>• "Forty third" → 43rd</li> <li>• "twenty sixth avenue" → 26th avenue</li> </ul>	<ul style="list-style-type: none"> <li>• "dreiundzwanzigste" → 23</li> <li>• "vierzigster" → 40</li> <li>• "ich war Erster" → "ich war Erster"</li> </ul>
Konvertieren von Bruchzahlen in ihr numerisches Format.	<ul style="list-style-type: none"> <li>• "a quarter" → 1/4</li> <li>• "three sixteenths" → 3/16</li> <li>• "a half" → 1/2</li> </ul>	Brüche werden nicht in ein numerisches Format umgewandelt.

Regeln	Englische Dialekte (Audio eingeben → Text ausgegeben)	Deutsche Dialekte (Audio eingeben → Text ausgegeben)
	<ul style="list-style-type: none"> <li>• "a hundredth" → 1/100</li> </ul>	<ul style="list-style-type: none"> <li>• "ein Drittel" → "ein Drittel"</li> </ul>
Wandelt Zahlen unter zehn in Ziffern um, wenn mehr als eine in einer Reihe stehen.	<ul style="list-style-type: none"> <li>• "three four five" → 345</li> <li>• "My phone number is four two five five five five one two one two" → My phone number is 4255551212</li> </ul>	<ul style="list-style-type: none"> <li>• "eins zwei drei" → 123</li> <li>• "plus vier neun zwei vier eins" → +49241</li> </ul>
Die Wörter „Punkt“ oder „Punkt“ werden als Dezimalzahl angezeigt.	<ul style="list-style-type: none"> <li>• "three hundred and three dot five" → 303.5</li> <li>• "three point twenty three" → 3.23</li> <li>• "zero point four" → 0.4</li> <li>• "point three" → 0.3</li> </ul>	Dezimalzahlen werden mit „,“ gekennzeichnet. <ul style="list-style-type: none"> <li>• "zweiundzwanzig komma drei" → 22,3</li> </ul>
Konvertieren des Wortes „Prozent“ nach einer Zahl zum Prozentzeichen (%).	<ul style="list-style-type: none"> <li>• "twenty three percent" → 23%</li> <li>• "twenty three point four five percent" → 23.45%</li> </ul>	<ul style="list-style-type: none"> <li>• "fünf Prozent Hürde" → 5% Hürde</li> <li>• "dreiundzwanzig komma vier Prozent" → 23,4%</li> </ul>

Regeln	Englische Dialekte (Audio eingeben → Text ausgeben)	Deutsche Dialekte (Audio eingeben → Text ausgeben)
Wandelt monetäre Wörter in Symbole um.	<p>Wandle die Wörter „Dollar“, „US-Dollar“, „Australischer Dollar“, „AUD“ oder „USD“ nach einer Zahl in ein Dollarzeichen (\$) vor der Zahl um.</p> <ul style="list-style-type: none"> <li>• "one dollar and fifteen cents" → \$1.15</li> <li>• "twenty three USD" → \$23</li> <li>• "twenty three Australian dollars" → \$23</li> </ul> <p>Konvertieren Sie der Wörter „Pfund“, „Britische Pfund“ oder „GDB“ nach einer Zahl in das Pfundzeichen (£) vor der Zahl.</p> <ul style="list-style-type: none"> <li>• "twenty three pounds" → £23</li> <li>• "I have two thousand pounds" → I have £2,000</li> <li>• "five pounds thirty three pence" → £5.33</li> </ul> <p>Konvertieren der Wörter „Rupien“, „Indische Rupien“ oder „INR“ nach einer Zahl in das Rupien-Zeichen (#) vor der Zahl.</p> <ul style="list-style-type: none"> <li>• "twenty three rupees" → #23</li> </ul>	<p>Wandle die Wörter „Euro“ in ein Euro-Zeichen um.</p> <ul style="list-style-type: none"> <li>• "ein euro" → 1 €</li> <li>• "ein Euro vierzig" → 1,40 €</li> <li>• "ein Euro vierzig Cent" → 1,40 €</li> </ul>

Regeln	Englische Dialekte (Audio eingeben → Text ausgeben)	Deutsche Dialekte (Audio eingeben → Text ausgeben)
	<ul style="list-style-type: none"> <li>• "fifty rupees thirty paise" → #50.30</li> </ul>	
Konvertieren von Zeiten in Zahlen.	<ul style="list-style-type: none"> <li>• "seven a m eastern standard time" → 7 a.m. eastern standard time</li> <li>• "twelve thirty p m" → 12:30 p.m.</li> </ul>	<ul style="list-style-type: none"> <li>• "vierzehn Uhr fünfzehn" → 14:15 Uhr</li> </ul>
Umwandeln von Datumsangaben in Zahlen.	<ul style="list-style-type: none"> <li>• "May fifth twenty twelve" → May 5th 2012</li> <li>• "May five twenty twelve" → May 5 2012</li> <li>• "five May twenty twelve" → 5 May 2012</li> </ul>	<ul style="list-style-type: none"> <li>• "dritter Dezember neunzehn hundert sechundfünfzig" → 3. Dezember 1956</li> </ul>
Trennen Sie Zahlenbereiche durch das Wort „bis“.	<ul style="list-style-type: none"> <li>• "twenty three to thirty seven" → 23 to 37</li> </ul>	Nicht zutreffend
Jahre werden vierstellig dargestellt. Dies gilt nur für Jahre im 20., 21. und 22. Jahrhundert.	<ul style="list-style-type: none"> <li>• "nineteen sixty two" → 1962</li> <li>• „the year is twenty twelve" → das Jahr ist 2012</li> <li>• "twenty nineteen" → 2019</li> <li>• "twenty one thirty" → 2130</li> </ul>	Nicht zutreffend

Regeln	Englische Dialekte (Audio eingeben → Text ausgeben)	Deutsche Dialekte (Audio eingeben → Text ausgeben)
Zeigt Schrägstriche und Bindestriche an.	<ul style="list-style-type: none"> <li>• "fifty-five dash thirteen" → 55-13</li> </ul> <p>Schrägstriche werden nicht angezeigt.</p> <ul style="list-style-type: none"> <li>• "fifty-five slash thirteen" → 55 slash 13</li> </ul>	<ul style="list-style-type: none"> <li>• "fünfundfünfzig Schrägstrich dreizehn" → 55/13</li> <li>• "fünfundfünfzig Strich dreizehn" → 55-13</li> </ul>
Nummerierte Absätze anzeigen.	<p>Nummerierte Absätze werden nicht mit dem Absatzsymbol (§) angezeigt.</p> <ul style="list-style-type: none"> <li>• "paragraph seventeen" → paragraph 17</li> </ul>	<ul style="list-style-type: none"> <li>• "Paragraf siebzehn" → § 17</li> </ul>

# Erste Schritte mit Amazon Transcribe

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

- [Registrieren für AWS-Konto](#)
- [Installieren Sie die AWS CLI und SDKs](#) (wenn Sie die AWS Management Console für Ihre Transkriptionen verwenden, können Sie diesen Schritt überspringen)
- [IAM Anmeldeinformationen konfigurieren](#)
- [Richten Sie einen Amazon S3 Bucket ein](#)
- [Erstellen Sie eine IAM Richtlinie](#)

Sobald Sie diese Voraussetzungen erfüllt haben, können Sie transkribieren. Wählen Sie Ihre bevorzugte Transkriptionsmethode aus der folgenden Liste aus, um loszulegen.

- [AWS CLI](#)
- [AWS Management Console](#)
- [AWS SDK](#)
- [HTTP](#)
- [WebSockets](#)

## Tip

Wenn Sie mit unseren Funktionen noch nicht vertraut sind Amazon Transcribe oder sie erkunden möchten, empfehlen wir die Verwendung von [AWS Management Console](#). Dies ist auch die einfachste Option, wenn Sie einen Stream mit Ihrem Computermikrofon starten möchten.

Da das Streamen mit HTTP/2 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.

## Anmeldung fürAWS-Konto

Sie können sich für ein [kostenloses Kontingent](#) oder ein [kostenpflichtiges Konto anmelden](#). Mit beiden Optionen haben Sie Zugriff auf alleAWS-Services. Das kostenlose Kontingent umfasst einen Testzeitraum, in dem Sie Ihre Nutzung untersuchenAWS-Services und abschätzen können. Sobald Ihre Testphase abgelaufen ist, können Sie auf ein kostenpflichtiges Konto migrieren. Die Gebühren fallen auf einer bestimmten pay-as-you-use Basis an; Einzelheiten finden Sie unter [Amazon TranscribePreise](#).

### Tip

Notieren Sie sich bei der Einrichtung Ihres Kontos IhreAWS-Konto ID, da Sie sie benötigen, umIAM Entitäten zu erstellen.

## Installation derAWS CLI und SDKs

Um dieAmazon Transcribe API verwenden zu können, müssen Sie zuerst die installierenAWS CLI. Die aktuelle VersionAWS CLI ist Version 2. Installationsanweisungen für [Linux](#), [Mac](#), [Windows](#) und [Docker](#) finden Sie im [AWS Command Line InterfaceBenutzerhandbuch](#).

Sobald Sie dasAWS CLI installiert haben, müssen Sie es für Ihre Sicherheitsanmeldeinformationen [konfigurieren](#) undAWS-Region.

Wenn Sie esAmazon 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 for Python \(Boto3\)](#)(Batch-Transkriptionen)
- [Python](#) (Streaming-Transkriptionen)
- [Ruby V3](#)

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

## IAMAnmeldeinformationen konfigurieren

Wenn Sie ein erstellenAWS-Konto, enthält es zunächst nur eine einzelne Anmeldeidentität, die über kompletten Zugriff auf alleAWS Dienste und Ressourcen in Ihrem Konto verfügt. Diese Identität wird alsAWS-Konto -Stammbenutzer bezeichnet. Um auf den Stammbenutzer zuzugreifen, müssen Sie sich mit der E-Mail-Adresse und dem Passwort anmelden, die zur Erstellung des Kontos verwendet wurden.

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.

Fordern Sie als bewährte Methode Benutzer, die Administratorzugriff benötigen, aufzufordern, den Verbund mit einem Identitätsanbieter zu verwenden, um auf mit temporären Anmeldeinformationen zuzugreifenAWS.

Eine Verbundidentität ist ein Benutzer, der aufAWS Dienste mit Anmeldeinformationen zuzugreifen, die über eine Identitätsquelle bereitgestellt wurden. 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 inIAM Identity Center. Sie können auch eine Verbindung mit einer Gruppe von Benutzern und Gruppen in Ihrer eigenen Identitätsquelle herstellen und synchronisieren, um sie in allenAWS-Konten und Anwendungen zu verwenden. Weitere Informationen finden Sie unter [Identity and Access Management für Amazon Transcribe](#).

Weitere Informationen zuIAM bewährten Methoden finden Sie unter [Bewährte Sicherheitsmethoden unterIAM](#).

## EinenAmazon S3 Bucket erstellen

Amazon S3ist ein sicherer Objektspeicherdienst. Amazon S3speichert Ihre Dateien (Objekte genannt) in Containern (sogenannten Buckets).

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

legen Sie Amazon Transcribe Ihr Transkript in einen temporär AWS verwalteten Amazon S3 Bucket. 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 einer IAM-Richtlinie

Um den Zugriff in zu verwalten AWS, müssen Sie Richtlinien erstellen und diese an IAM -Identitäten (Gruppen oder Rollen) oder AWS Ressourcen anfügen. Eine Richtlinie definiert die Berechtigungen der Entität, der sie zugeordnet ist. Beispielsweise kann eine Rolle nur dann auf eine Mediendatei in Ihrem Amazon S3 Bucket zugreifen, wenn Sie dieser Rolle eine Richtlinie angefügt 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 Genehmigungen in IAM](#)
- [IAM Richtlinien erstellen](#)
- [Wie Amazon Transcribe funktioniert mit IAM](#)

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

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

Richtlinien haben das Format:

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

```

    ],
    "Resource": [
        "amazon-resource-name"
    ]
  }
]
}

```

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

```

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

```

Hier ist ein Beispiel für eine Richtlinie `GetObject, ListBucket` die einem Amazon S3 Bucket und seinen Unterordnern Amazon Transcribe Lese- (`DOC-EXAMPLE-BUCKET, PutObject`) - und Schreibrechte () gewährt:

```

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

```

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

## Transkribieren mit dem AWS Management Console

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

Bevor Sie mit einer Batch-Transkription beginnen, müssen Sie zuerst Ihre Mediendatei in einen Amazon S3 Bucket hochladen. Für Streaming-Transkriptionen mit dem AWS Management Console müssen Sie Ihr Computermikrofon verwenden.

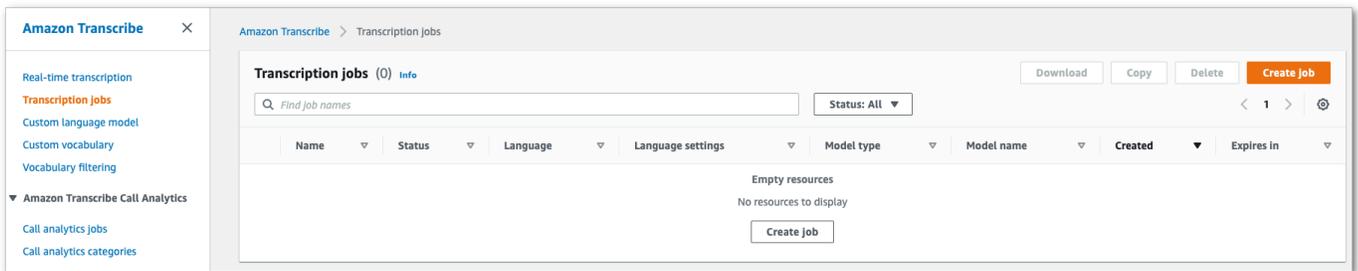
Informationen zu unterstützten Medienformaten und anderen Medienanforderungen und Einschränkungen finden Sie unter [Ein- und Ausgabe von Daten](#).

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: [Laden Sie ein Objekt in Ihren Bucket](#) hoch.

1. Wählen Sie im linken Navigationsbereich Transkriptionsaufträge aus. [AWS Management Console](#) Dadurch gelangen Sie zu einer Liste Ihrer Transkriptionsjobs.



Wählen Sie **Job erstellen** aus.

2. Füllen Sie die Felder auf der Seite „Jobdetails angeben“ 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 innerhalb eines Amazon S3 Buckets sein. Als Ausgabeort können Sie einen sicheren, vom Amazon S3 Service verwalteten Bucket wählen oder Ihren eigenen Amazon S3 Bucket angeben.

Wenn Sie sich für einen vom Service verwalteten Bucket entscheiden, können Sie sich eine Transkriptvorschau im [Ansehen AWS Management Console](#) und Ihr Transkript von der Seite mit den Jobdetails herunterladen (siehe unten).

Wenn Sie Ihren eigenen Amazon S3 Bucket auswählen, können Sie im Bucket keine Vorschau [sehen AWS Management Console](#) und 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.

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

---

### Output data

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

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

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

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

SRT (SubRip)

VTT (WebVTT)

---

### Tags - optional

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

No tags associated with the resource.

You can add up to 50 more tags.

Klicken Sie auf Next (Weiter).

3. Wählen Sie auf der Seite „Job konfigurieren“ die gewünschten Optionen aus. Wenn Sie [Benutzerdefinierte Vokabulare](#) oder [Benutzerdefinierte Sprachmodelle](#) mit Ihrer Transkription verwenden möchten, müssen Sie diese erstellen, bevor Sie mit Ihrem Transkriptionsjob beginnen.

## 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 Job erstellen aus.

4. Sie befinden sich jetzt auf der Seite mit Transkriptionsjobs. Hier können Sie den Status des Transkriptionsauftrags. Wählen Sie nach Abschluss Ihre Transkription aus.

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

5. Sie sehen jetzt die Seite mit den Stellendetails 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 verknüpften Dateipfad in der rechten Spalte unter Speicherort der Ausgabedaten aus. Dadurch gelangen Sie zu dem von Ihnen angegebenen Amazon S3 Ausgabeordner. Wählen Sie Ihre Ausgabedatei aus, die jetzt die Erweiterung .json hat.

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	2/4/2022, 12:11:31 PM	Disabled	Service-managed S3 bucket
Language	Started	Custom vocabulary	
English, US (en-US)	2/4/2022, 12:11:31 PM	None	
Language settings	Ended	PII redaction	
Specific language	2/4/2022, 12:12:50 PM	Disabled	
Expiration <a href="#">Info</a>	Input file format	Vocabulary filter	
The transcription is available for 87 more days.	flac	-	
	Audio sampling rate		
	44100 Hz		

6. Wie Sie Ihr Transkript herunterladen, hängt davon ab, ob Sie einen vom Service verwalteten Amazon S3 Bucket oder Ihren eigenen Amazon S3 Bucket ausgewählt haben.
- Wenn Sie sich für einen vom Service verwalteten Bucket entschieden haben, wird auf der Informationsseite Ihres Transkriptionsauftrags ein Vorschaufenster für die Transkription sowie eine Schaltfläche zum Herunterladen angezeigt.

**my-first-transcription-job**

---

**Job details**

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

---

**Transcription preview**  ▼

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

**Text**
Audio identification
Subtitles

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

Wählen Sie Herunterladen und dann Transkript herunterladen.

- b. Wenn Sie Ihren eigenen Amazon S3 Bucket ausgewählt haben, wird im Bereich 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**

---

**Job details**

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

---

**Transcription preview**

Select download to save a local copy of the transcription.

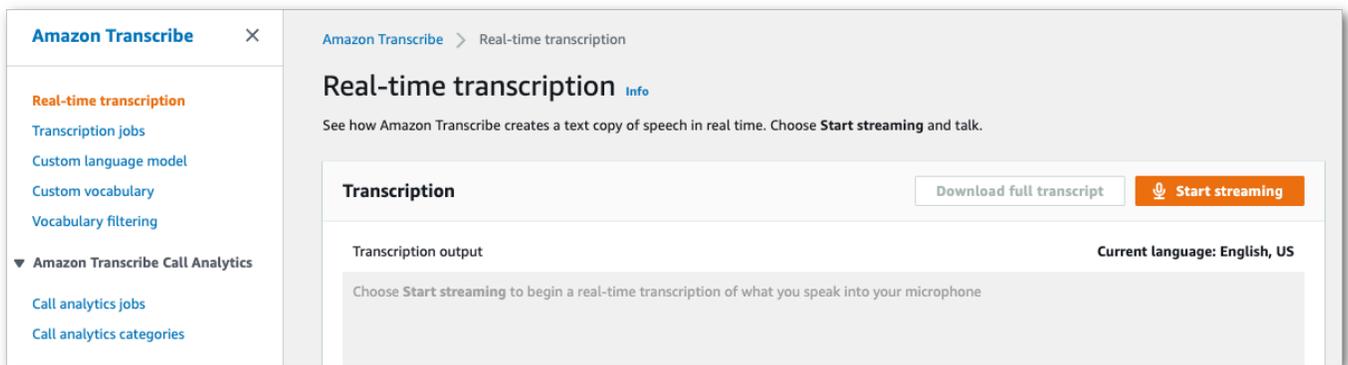
Text
Audio identification
Subtitles

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

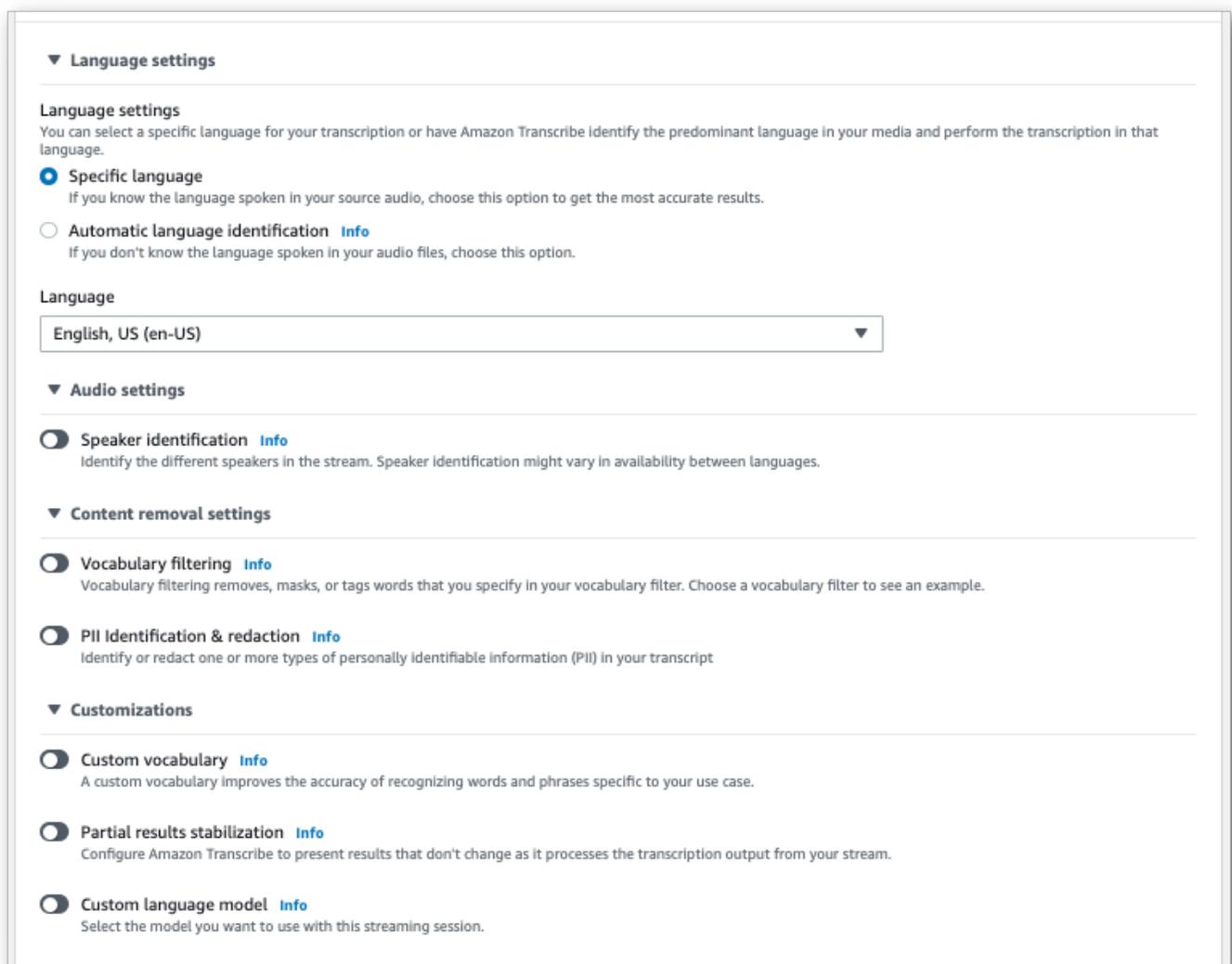
Um auf Ihr Transkript zuzugreifen, wechseln Sie zum angegebenen Amazon S3 Bucket. Verwenden Sie dazu den Link unter Speicherort der Ausgabedaten im Bereich Auftragsdetails oder den Link S3-Bucket im blauen Informationsfeld im Bereich Transkriptionsvorschau.

## Streaming-Transkriptionen

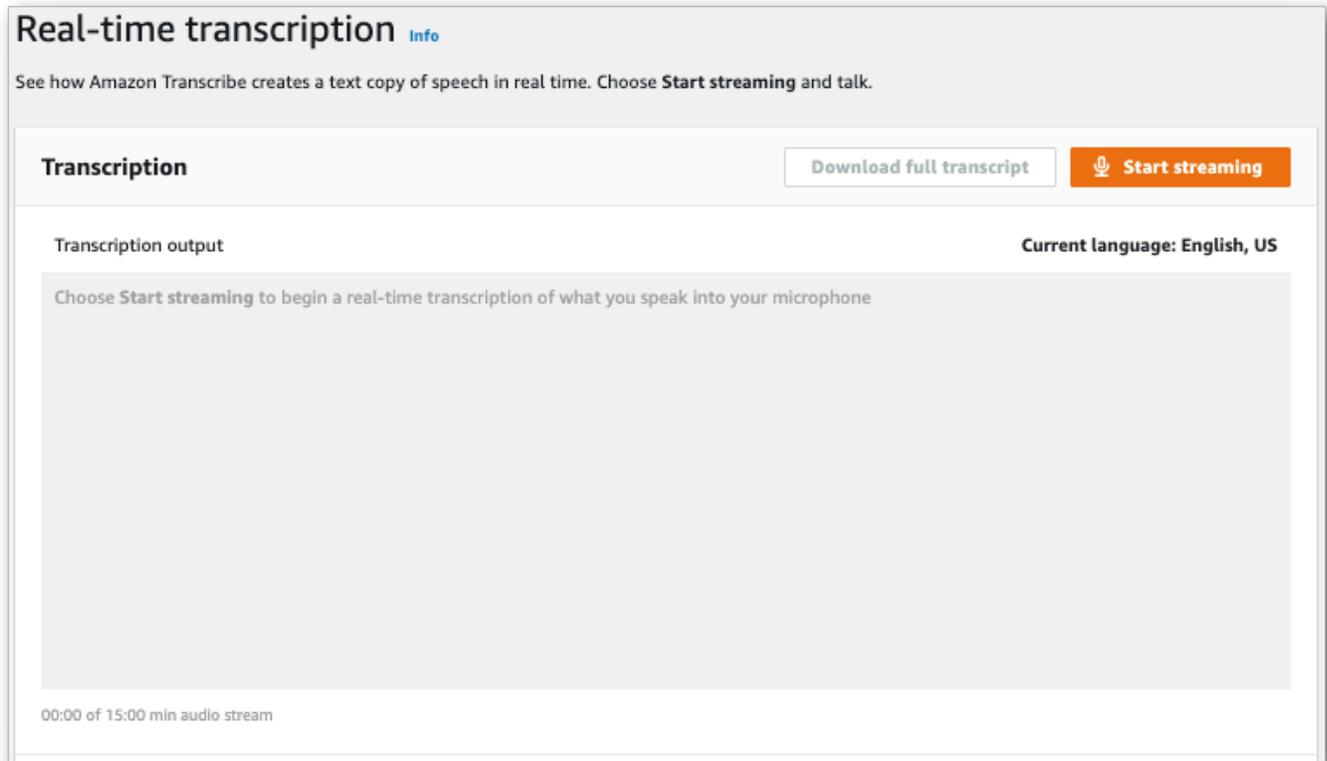
1. Wählen Sie im [AWS Management Console](#) linken Navigationsbereich die Option Transkription in Echtzeit aus. Dadurch gelangen Sie zur Streaming-Hauptseite, auf der Sie vor dem Start Ihres Streams Optionen auswählen können.



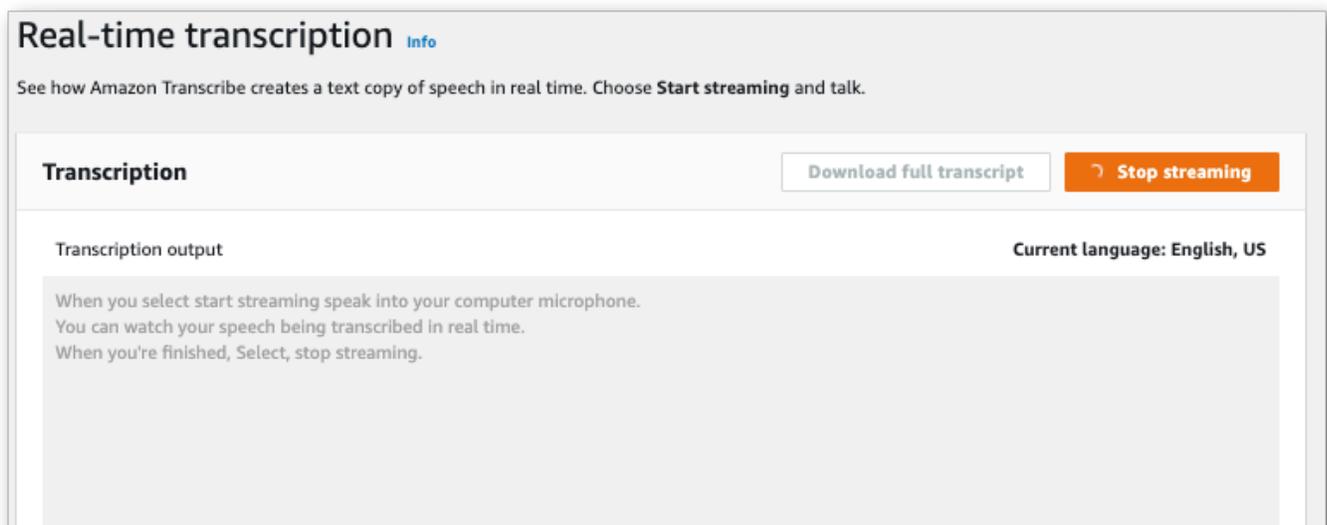
2. Unter dem Ausgabefeld für die Transkription haben Sie die Möglichkeit, verschiedene Sprach- und Audioeinstellungen auszuwählen.



3. Nachdem Sie die entsprechenden Einstellungen ausgewählt haben, scrollen Sie zum Seitenanfang und wählen Sie „Streaming starten“ und beginnen Sie dann, in Ihr Computermikrofon zu sprechen. Sie können Ihre Rede in Echtzeit transkribiert sehen.



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



Sie können Ihr Transkript jetzt herunterladen, indem Sie Vollständiges Transkript herunterladen auswählen.

# Transkribieren mit demAWS CLI

Wenn Sie die verwenden,AWS CLI um eine Transkription zu starten, können Sie alle Befehle auf CLI-Ebene ausführen. Oder Sie können den Befehl ausführen, den Sie verwenden möchten, gefolgt von demAWS-Region und dem Speicherort einer JSON-Datei, die einen Anforderungstext enthält. In diesem Handbuch finden Sie Beispiele für beide Methoden. Dieser Abschnitt konzentriert sich jedoch auf die erstere Methode.

Das unterstütztAWS CLI keine Streaming-Transkriptionen.

Bevor Sie fortfahren, stellen Sie sicher, dass Sie:

- Deine Mediendatei wurde in einenAmazon S3 Bucket hochgeladen. Wenn Sie sich nicht sicher sind, wie Sie einenAmazon S3 Bucket erstellen oder Ihre Datei hochladen, finden Sie weitere Informationen unter [Erstellen Sie Ihren erstenAmazon S3 Bucket](#) und [Laden Sie ein Objekt in Ihren Bucket](#) hoch.
- Installierte die [AWS CLI](#).

Sie finden alleAWS CLI Befehle fürAmazon Transcribe in der [AWS CLIBefehlsreferenz](#).

## Einen neuen Transkriptionsjob starten

Verwenden Sie den`start-transcription-job` Befehl, um eine neue Transkription zu starten.

1. Geben Sie in einem Terminalfenster den folgenden Befehl ein:

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

In der nächsten Zeile erscheint ein`>`, und Sie können nun mit dem Hinzufügen der erforderlichen Parameter fortfahren, wie im nächsten Schritt beschrieben.

Sie können das `\` auch weglassen und alle Parameter anhängen, indem Sie sie jeweils durch ein Leerzeichen trennen.

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

Wenn Sie einen Ausgabeort angeben möchten, fügen Sie dies`output-bucket-name` in Ihre Anfrage ein. Wenn Sie einen Unterordner des angegebenen Ausgabe-Buckets angeben möchten, schließen Sie auch ein`output-key`.

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

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://DOC-EXAMPLE-BUCKET/my-  
input-files/my-media-file.flac --language-code en-US
```

Wenn Sie sich dafür entscheiden, keinen Ausgabe-Bucket mit anzugeben `output-bucket-name`, Amazon Transcribe platziert Ihre Transkriptionsausgabe in einem vom Service verwalteten Bucket. In einem vom Service verwalteten Bucket gespeicherte 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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-  
file.flac"  
    },  
    "StartTime": "2022-03-07T15:03:44.246000-08:00",  
    "CreationTime": "2022-03-07T15:03:44.229000-08:00"  
  }  
}
```

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

## Abrufen des den Status eines Abtrags

Verwenden Sie den `get-transcription-job` Befehl, um Informationen über Ihren Transkriptionsjob zu erhalten.

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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"  
    },  
    "Transcript": {  
      "TranscriptFileUri": "https://s3.the-URI-where-your-job-is-located.json"  
    },  
    "StartTime": "2022-03-07T15:03:44.246000-08:00",  
    "CreationTime": "2022-03-07T15:03:44.229000-08:00",  
    "CompletionTime": "2022-03-07T15:04:01.158000-08:00",  
    "Settings": {  
      "ChannelIdentification": false,  
      "ShowAlternatives": false  
    }  
  }  
}
```

Wenn Sie Ihren eigenen Amazon S3 Bucket für Ihre Transkriptionsausgabe ausgewählt haben, wird dieser Bucket mit `TranscriptFileUri` aufgeführt. Wenn Sie einen vom Service verwalteten Bucket ausgewählt haben, wird eine temporäre URI bereitgestellt. Verwenden Sie diese 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 `get-transcription-job` Anfrage erneut aus, um eine neue temporäre URI zu erhalten.

## Deine Transkriptionsjobs auflisten

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

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

Verwenden Sie den `delete-transcription-job` Befehl, um Ihren Transkriptionsauftrag zu löschen.

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 `list-transcription-jobs` Befehl ausführen. Ihr Job sollte nun nicht mehr in der Liste angezeigt werden.

## Transkribieren mit den SDKs AWS

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 [Ein- und Ausgabe von Daten](#).

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 SDKs AWS](#)

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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-
file.flac")
            .build();

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

```
        .outputBucketName(outputS3BucketName)
        .build();

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

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

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

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

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

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

## JavaScript

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

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

const input = {
    TranscriptionJobName: "my-first-transcription-job",
```

```
    LanguageCode: "en-US",
    Media: {
      MediaFileUri: "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
    },
    OutputBucketName: "DOC-EXAMPLE-BUCKET",
  };

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

startTranscriptionRequest();
```

## Python

```
import time
import boto3

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

    max_tries = 60
    while max_tries > 0:
        max_tries -= 1
```

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

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

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

## Streaming-Transkriptionen

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

Beachten Sie, dass der Standard AWS SDK for 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 SDK AWS

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

SDK-Dokumentation	Codebeispiele
<a href="#">AWS SDK for C++</a>	<a href="#">AWS SDK for C++ Codebeispiele</a>
<a href="#">AWS CLI</a>	<a href="#">AWS CLI Codebeispiele</a>
<a href="#">AWS SDK for Go</a>	<a href="#">AWS SDK for Go Codebeispiele</a>
<a href="#">AWS SDK for Java</a>	<a href="#">AWS SDK for Java Codebeispiele</a>
<a href="#">AWS SDK for JavaScript</a>	<a href="#">AWS SDK for JavaScript Codebeispiele</a>
<a href="#">AWS SDK for Kotlin</a>	<a href="#">AWS SDK for Kotlin Codebeispiele</a>
<a href="#">AWS SDK for .NET</a>	<a href="#">AWS SDK for .NET Codebeispiele</a>
<a href="#">AWS SDK for PHP</a>	<a href="#">AWS SDK for PHP Codebeispiele</a>
<a href="#">AWS Tools for PowerShell</a>	<a href="#">Tools für PowerShell Codebeispiele</a>
<a href="#">AWS SDK for Python (Boto3)</a>	<a href="#">AWS SDK for Python (Boto3) Codebeispiele</a>
<a href="#">AWS SDK for Ruby</a>	<a href="#">AWS SDK for Ruby Codebeispiele</a>

SDK-Dokumentation	Codebeispiele
<a href="#">AWS SDK for Rust</a>	<a href="#">AWS SDK for Rust Codebeispiele</a>
<a href="#">AWS SDK für SAP ABAP</a>	<a href="#">AWS SDK für SAP ABAP Codebeispiele</a>
<a href="#">AWS SDK for Swift</a>	<a href="#">AWS SDK for Swift Codebeispiele</a>

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

#### 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-Transkriptionen (HTTP/2). WebSockets werden für Streaming-Transkriptionen unterstützt.

Wenn Sie eine Mediendatei in einem Amazon S3 Bucket transkribieren, führen Sie eine Batch-Transkription durch. Wenn Sie einen Echtzeit-Stream 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 [AWSAPI-Anfragen signieren](#).

### Batch-Transkriptionen

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

- Host
- x-amz-target
- Inhaltstyp

- x-amz-content-sha256
- x-amz-date
- Autorisierung

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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
  },
  "OutputBucketName": "DOC-EXAMPLE-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 sind unter [Elemente einer AWS Signature-Version 4-Anfrage](#) detailliert beschrieben.

## Streaming-Transkriptionen

Streaming-Transkriptionen verwenden HTTP/2 und WebSockets sind aufwendiger als die Verwendung von SDKs. Wir empfehlen, den [Einrichten einer Streaming-Transkription](#) Abschnitt zu lesen, bevor du deinen ersten Stream einrichtest.

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

 Note

Wir empfehlen dringend, ein SDK für Streaming-Transkriptionen zu verwenden. 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 der Verwendung von Streaming gegenüber Batch gehören speech-to-text Echtzeitfunktionen 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 Management Console](#)

Um Streaming-Audio im zu transkribieren AWS Management Console, 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-kodiertes 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 Amazon Transcribe regionalen Verfügbarkeit für Streaming-Transkriptionen finden Sie unter: [Amazon Transcribe Endpunkte](#) und Kontingente.

## Bewährte Methoden

Die folgenden Empfehlungen verbessern die Effizienz der Streaming-Transkription:

- Verwenden Sie nach Möglichkeit PCM-kodiertes Audio.
- 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.

## Streaming und Teilergebnisse

Da das Streaming in Echtzeit funktioniert, werden die Transkripte nur teilweise 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 Prozess in Aktion sehen, indem Sie sich auf der Website [AWS Management Console](#) anmelden, Echtzeit-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.

## Stabilisierung von Teilergebnissen

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

Durch die Stabilisierung teilweiser Ergebnisse ändert sich 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":
Stabilisierung von Teilergebnissen nicht aktiviert		
The 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.	0.545 1.045 1.545 2.045 2.545 3.045 3.545 4.045 4.545 5.045 5.545 6.025 6.025	true true true true true true true true true true true true false

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

### Stabilisierung von Teilergebnissen 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 largest	3.015	true
The Amazon is the largest rainfall.	3.515	true
The Amazon is the largest rain forest.	4.015	true
The Amazon is the largest rain forest on	4.515	true
The Amazon is the largest rain forest on	5.015	true
The Amazon is the largest rain forest on	5.515	true
The Amazon is the largest rain forest on	6.015	true
The Amazon is the largest rain forest on	6.335	true
The Amazon is the largest rain forest on	6.335	false

Wenn Sie die Stabilisierung teilweiser Ergebnisse aktivieren, gibt Amazon Transcribe diese Option in einem `Stable` Feld an, ob ein Element stabil ist, wobei sich „Element“ auf ein transkribiertes Wort oder ein Satzzeichen 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 mit dem Blog zum AWS Machine Learning tiefer ein

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

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

## Beispiel für die Stabilisierung von Teilergebnissen

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 nicht mit einem SDK, sondern mit HTTP/2 oder WebSockets direkt einrichten möchten. AWS Die Informationen in diesem Abschnitt können auch zur Erstellung eines eigenen SDKs verwendet werden.

### Important

Wir empfehlen dringend, SDKs anstelle von HTTP/2 und direkt zu verwenden. WebSockets 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 den SDKs AWS](#)

## Einrichten eines HTTP/2-Streams

Die wichtigsten Komponenten eines [HTTP/2-Protokolls](#) 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. Hängen Sie die folgende Richtlinie an die IAM Rolle an, die die Anfrage stellt. Weitere Informationen finden Sie unter [IAM Richtlinien hinzufügen](#).

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

2. Senden Sie eine HTTP/2-Anfrage an Amazon Transcribe, um die Sitzung zu starten:

```
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	Wertzeichenfolge (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"
Uk1GRjzxPQBxQVZFZm10IBAAAAABAAEAgD4AAAB9AAACABAAZGF0YVVTwPQAAAAAAAAAAAAAAAAAAD//wIA/
f8EAA==
```

4. Erstellen Sie eine Audionachricht, die Ihre Audiodaten enthält.
  - a. 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
```

```
AAAA0gAAAIKVoRFcTTcjb250ZW50LXR5cGUHABhhcHBsaWNhdGlvbi9vY3RldC1zdHJlYW0L0mV2ZW50LXR5cGUHAApBdWRpb0V2ZW50DTptZXNzYwd1LXR5cGUHAAV1dmVudAxDb256ZW50LVR5cGUHABphcHBsaWNhdGlvbi94LWFtei1qc29uLTEuMVJJRkY88T0AV0FWRWZtdCAQAAAAAQABAIA
+AAAAfQAAAQAQAGRhdGFU8D0AAAAA
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`. z. B. `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 eine Signatur unter Verwendung der SHA-256-Hash-Funktion erstellt.
- abgeleitet SigningKey: Der Signature Version 4-Signaturschlüssel.
- 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
```

## Behandlung von 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	BadRequestAusnahme
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](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "my-transcribe-websocket-policy",
      "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. z. B. `20220127/us-west-2/transcribe/aws4_request`.

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

- f. Erstellen Sie die kanonische Abfragezeichenfolge. Die Werte der Abfragezeichenfolge müssen URI-kodiert und 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.
  - Fügen Sie keine URI-Kodierung für die nicht reservierten Zeichen durch, die von RFC 3986 definiert sind: A–Z, a–z, 0–9, Bindestrich (-), Unterstrich (\_), Punkt (.) und Tilde (~).
  - Versehen Sie alle anderen Zeichen mit Prozentcode (`%XY`), wobei X und Y für Hexadezimalzeichen, d. h. 0-9 und die Großbuchstaben A-F, stehen. Zum Beispiel muss das Leerzeichen als `%20` kodiert werden (schließen Sie kein „+“ ein, wie es einige

Kodierungsschemata tun); erweiterte UTF-8-Zeichen müssen in der Form %XY%ZA%BC sein.

- Doppelcodieren Sie alle gleich (=)-Zeichen 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. Leiten Sie dazu 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 liefert.

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 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	Wertzeichenfolge (UTF-8)
13	:content-type	7	24	application/octet-stream
11	:event-type	7	10	AudioEvent
13	:message-type	7	5	event

9. 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	Wertzeichenfolge (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 Zugriffsschlüssel oder geheimen Schlüssel 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 Ereignisstreamkodierung 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

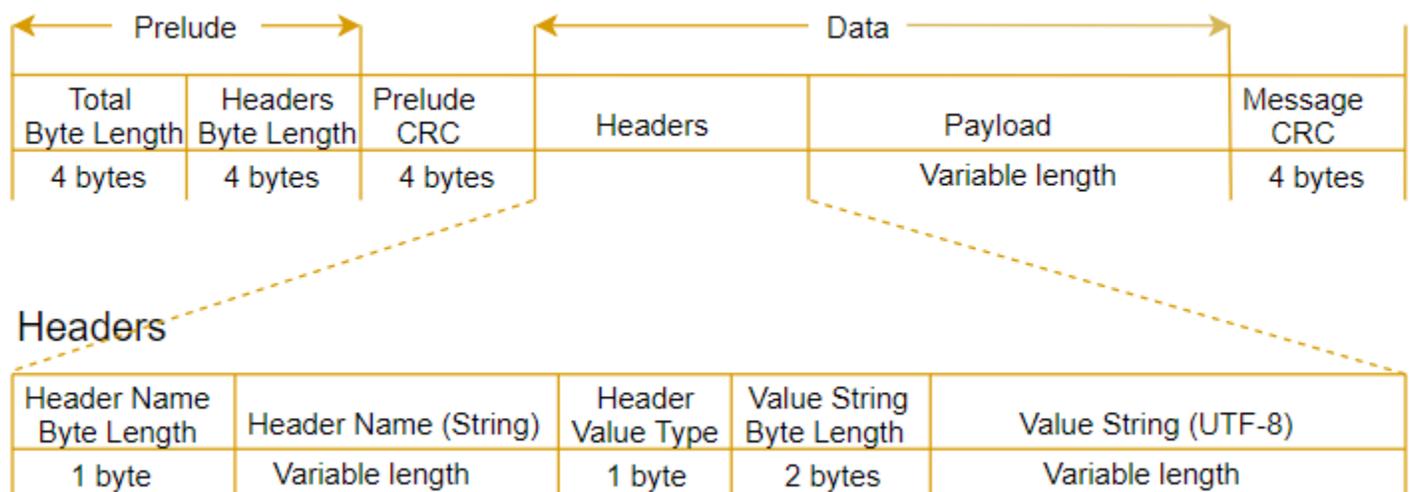
Der Datenabschnitt setzt sich folgendermaßen zusammen:

1. Überschriften
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, die Schlüssel:Wert-Paare sind, wobei der Schlüssel eine UTF-8-Zeichenfolge ist. 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

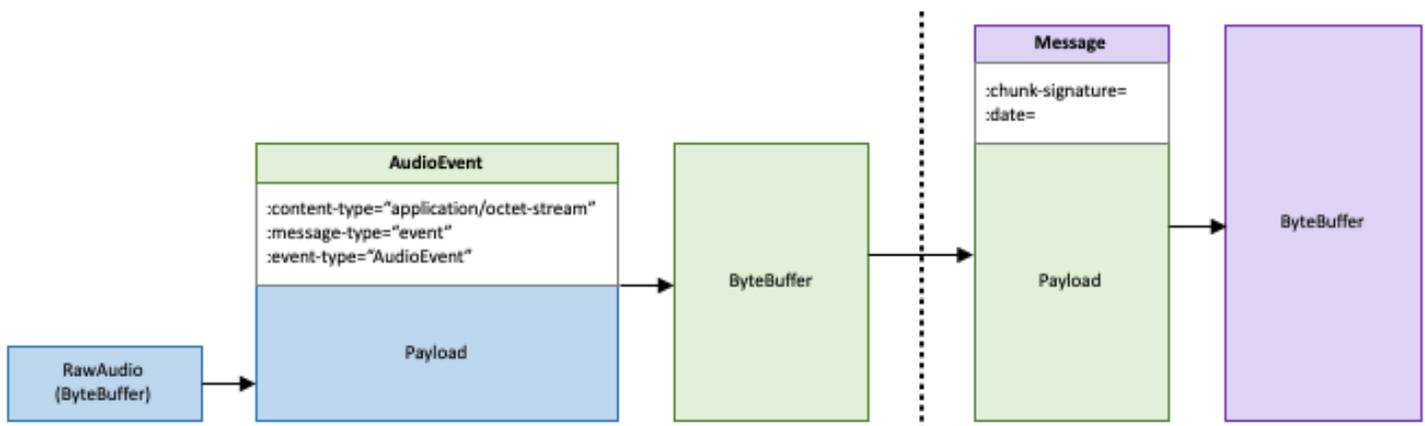
- 7 – Zeichenfolge
- 8 – ZEITSTEMPEL
- 9 – UUID
- Byte-Länge der Wertzeichenfolge: die Byte-Länge der Header-Wertzeichenfolge.
- Header-Wert: der Wert der Header-Zeichenfolge. Gültige Werte für dieses Feld sind vom Header-Typ abhängig. Weitere Informationen finden Sie unter [Einrichten eines HTTP/2-Streams](#) 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.



## AuftragswarteschlangenWarteschlange tesc

Mithilfe von Job Queueing können Sie mehr Transkriptionsanfragen einreichen, als gleichzeitig bearbeitet werden können. Ohne Job Queueing müssen Sie, sobald Sie das Kontingent der zulässigen gleichzeitigen Anforderungen erreicht haben, warten, bis eine oder mehrere Anforderungen abgeschlossen sind, bevor Sie eine neue Anfrage senden.

Das Einreihen von Job in die Warteschlange ist für Transkriptionsanfragen optional. Bei Analyseanfragen nach einem Anruf ist die Auftragswarteschlange automatisch aktiviert.

Wenn Sie Job Queueing 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 abgerufen und bearbeitet. Anfragen in der Warteschlange werden in einer FIFO-Reihenfolge (First In, First Out) bearbeitet.

Sie können bis zu 10.000 Auftragswarteschlangenlimit zu Ihrer Warteschlange zu bis zu 10.000 AuftragsWarteschlange zu Wenn Sie dieses Limit überschreiten, erhalten Sie eineLimitExceededConcurrentJobException Fehlermeldung. Um eine optimale Leistung zu gewährleisten, werdenAmazon 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 fürAmazon Transcribe Ressourcen finden Sie in der [AWSAllgemeinen Referenz](#). Einige dieser Standardwerte können auf Anfrage erhöht werden.

Wenn Sie Job Queueing aktivieren, aber das Kontingent für gleichzeitige Anfragen nicht überschreiten, werden alle Anfragen gleichzeitig verarbeitet.

## Aktivieren von AuftragsWarteschlange tescWarteschlange tesc

Sie können Job Queueing mithilfe der AWSSDKs AWS Management ConsoleAWS CLI, oder aktivieren. Beispiele finden Sie im Folgenden; Beispiele finden Sie im Folgenden:

## AWS Management Console

1. Melden Sie sich an der [AWS Management Console](#) an.
2. Wählen Sie im Navigationsbereich Transkriptionsaufträge und dann Job erstellen (oben rechts) aus. Dadurch wird die Seite „Jobdetails angeben“ geöffnet.
3. Im Feld „Jobeinstellungen“ befindet sich ein Bereich mit zusätzlichen Einstellungen. Wenn Sie diesen Bereich erweitern, können Sie das Feld Zur Jobwarteschlange hinzufügen auswählen, 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 auf der Seite „Jobdetails angeben“ alle weiteren Felder aus, die Sie einbeziehen möchten, und wählen Sie dann Weiter aus. Dadurch gelangen Sie zur Seite Job konfigurieren — optional.
5. Wählen Sie Job erstellen aus, um Ihren Transkriptionsjob auszuführen.

## AWS CLI

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

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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac \  
--output-bucket-name DOC-EXAMPLE-BUCKET \  
--output-key my-output-files/ \  
--language-code en-US \  
--job-execution-settings  
  AllowDeferredExecution=true,DataAccessRoleArn=arn:aws:iam::111122223333:role/  
ExampleRole
```

Hier ist ein weiteres Beispiel für die Verwendung des [start-transcription-job](#) Befehls und eines Anforderungstexts, der die Warteschlange ermöglicht.

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

}

## AWS SDK for Python (Boto3)

In diesem Beispiel wird das verwendete AWS SDK for Python (Boto3), um die Auftragswarteschlange mithilfe des `AllowDeferredExecution` Arguments für die Methode [start\\_transcription\\_job](#) zu aktivieren. Beachten Sie, dass Sie `AllowDeferredExecution` in Ihrer Anfrage auch angeben müssen `DataAccessRoleArn`. Weitere Informationen erhalten Sie unter [StartTranscriptionJob](#) und [JobExecutionSettings](#).

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

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

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

Sie können den Fortschritt eines Jobs in der Warteschlange über den AWS Management Console oder durch Absenden einer [GetTranscriptionJob](#) Anfrage verfolgen. Wenn ein Job in die Warteschlange gestellt wird, Status ist `QUEUED`. Der Status ändert sich zu „`IN_PROGRESS`“ sobald Ihr Job mit der Verarbeitung beginnt“ und dann zu „`COMPLETED`“ oder `FAILED` wenn die Verarbeitung abgeschlossen ist“.

# Markieren von Ressourcen

Ein Tag ist ein benutzerdefiniertes Metadatenlabel, das Sie einer Ressource hinzufügen können, um sie bei einer Suche leichter identifizieren, organisieren und 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. Zum Beispiel könnten Sie das Tag `key=Color` und das Tag `value=blue` haben, wodurch das Schlüssel:Value-Paar entsteht `Color:Blue`. Beachten Sie, dass Sie den Wert eines Tags auf eine leere Zeichenfolge setzen können, aber Sie können den Wert eines Tags nicht auf null setzen. Ein nicht angegebener Tag-Wert entspricht einer leeren Zeichenfolge.

## Tip

AWS Billing and Cost Management kann Tags verwenden, um Ihre Rechnungen in dynamische Kategorien zu unterteilen. Wenn Sie beispielsweise Stichwörter hinzufügen, um verschiedene Abteilungen innerhalb Ihres Unternehmens zu repräsentieren, wie z. B. `Department:Sales` oder `Department:Legal`, AWS können Sie Ihre Kostenverteilung pro Abteilung ermitteln.

In Amazon Transcribe können Sie die folgenden Ressourcen mit Tags versehen:

- Transkriptionsjobs
- Jobs im Bereich der medizinischen Transkription
- Transkriptionsaufträge nach dem Anruf von Call Analytics
- Benutzerdefinierte definierte Vokabieren
- Benutzerdefiniertes medizinisches Vokabular
- Benutzerdefinierte Vokabelfilter
- Benutzerdefinierte Sprachmodelle

Die Tag-Schlüssel können bis zu 128 Zeichen lang sein, und die Tag-Werte können bis zu 256 Zeichen lang sein. Bei beiden wird zwischen Groß- und Kleinschreibung unterschieden. Amazon Transcribe unterstützt bis zu 50 Tags pro Ressource. Für eine bestimmte Ressource muss jeder

einzelne Tag mit nur einem Wert sein. Beachten Sie, dass Ihre Tags nicht mit `aws:` beginnen können, da dieses Präfix für vom System generierte Tags von AWS reserviert ist. Sie können keine `aws:*` Tags hinzufügen, ändern oder löschen, und sie werden nicht auf Ihr `tags-per-resource` Limit angerechnet.

 API-Operationen, die für das Ressourcen-Tagging spezifisch sind

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

Um die Tagging-APIs verwenden zu können, müssen Sie Ihrer Anfrage einen Amazon Resource Name (ARN) hinzufügen. ARNs haben das Format `arn:partition:service:region:account-id:resource-type/resource-id`. Der ARN, der einem Transkriptionsjob zugeordnet ist, kann beispielsweise wie folgt aussehen: `arn:aws:transcribe:us-west-2:111122223333:transcription-job/my-transcription-job-name`.

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

## Tag-basierte Zugriffskontrolle

Sie können Tags verwenden, um den Zugriff innerhalb Ihres AWS-Kontos zu steuern. Bei der tag-basierten Zugriffskontrolle geben Sie im Bedingungelement einer IAM Richtlinie Tag-Informationen an. Sie können dann Tags und den zugehörigen Tag-Bedingungsschlüssel verwenden, um den Zugriff auf Folgendes zu steuern:

- Markieren Sie den Zugriff auf Ihre Amazon Transcribe-Ressourcen, basierend auf den Tags, die Sie diesen Ressourcen zugewiesen haben.
  - Verwenden Sie den `aws:ResourceTag/key-name` Bedingungs-schlüssel, um anzugeben, welches Tag-Schlüsselwert-Paar an die Ressource angefügt werden muss.
- Anfragen: Steuern, welche Tags an eine Anforderung weitergeleitet werden können.
  - Verwenden Sie den `aws:RequestTag/key-name` Bedingungs-schlüssel, um anzugeben, welche Tags hinzugefügt, geändert oder aus IAM-Benutzern oder -Rollen entfernt werden können.
- Autorisierungsprozesse: Steuern Sie den tag-basierten Zugriff für jeden Teil Ihres Autorisierungsprozesses.

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

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

Weitere Informationen zur Tag-basierter Zugriffskontrolle finden Sie unter [Steuern des Zugriffs auf AWS -ieren mit Tags](#).

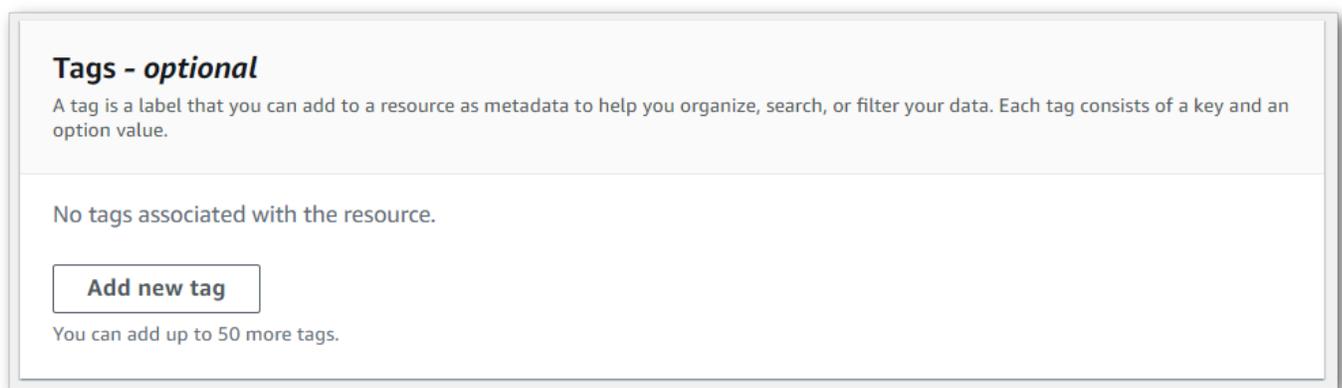
## Hinzufügen von Stichwörtern zu Ihren Amazon Transcribe Ressourcen

Sie können Tags hinzufügen, bevor oder nachdem Sie Ihren Amazon Transcribe Job ausgeführt haben. Mithilfe der vorhandenen `Create*` - und `Start*`-APIs können Sie Ihrer Transkriptionsanfrage Tags hinzufügen.

Mit den AWSSDKs, oder können Sie Tags hinzufügen AWS Management Console AWS CLI, ändern oder löschen. Beispiele finden Sie im Folgenden:

### AWS Management Console

1. Melden Sie sich an der [AWS Management Console](#) an.
2. Wählen Sie im Navigationsbereich Transkriptionsaufträge und dann Job erstellen (oben rechts) aus. Dadurch wird die Seite „Jobdetails angeben“ geöffnet.
3. Scrollen Sie auf der Seite „Stellendetails angeben“ zum Ende, um das optionale Feld Schlagworte zu finden, und wählen Sie Neues Schlagwort hinzufügen aus.



4. Geben Sie Informationen für das Schlüsselfeld und optional für das Wertfeld 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

Value - optional

You can add up to 49 more tags.

5. Füllen Sie auf der Seite „Jobdetails angeben“ alle weiteren Felder aus, die Sie einbeziehen möchten, und wählen Sie dann Weiter aus. Dadurch gelangen Sie zur Seite Job konfigurieren — optional.

Wählen Sie Job erstellen aus, um Ihren Transkriptionsjob auszuführen.

6. Sie können die mit einem Transkriptionsauftrag verknüpften Tags einsehen, indem Sie zur Seite mit den Transkriptionsaufträgen navigieren, einen Transkriptionsauftrag auswählen und auf der Informationsseite dieses Jobs zum Ende scrollen. Wenn Sie Ihre Ihre Tags bearbeiten möchten, können Sie dies tun, indem Sie auf Tags verwalten klicken.

**Tags (2)**

Key	▼	Value
color		blue

## AWS CLI

In diesem Beispiel werden der [start-transcription-job](#) Befehl und der Tags Parameter 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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac \
--output-bucket-name DOC-EXAMPLE-BUCKET \
```

```
--output-key my-output-files/ \  
--language-code en-US \  
--tags Key=color,Value=blue Key=shape,Value=square
```

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

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

## AWS SDK for Python (Boto3)

Im folgenden Beispiel wird das verwendete AWS SDK for Python (Boto3), um mithilfe des `Tags` Arguments für die Methode [start\\_transcription\\_job](#) ein Tag hinzuzufügen. Weitere Informationen erhalten Sie unter [StartTranscriptionJob](#) und [Tag](#).

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

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

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

## 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"
        }
      ],
      ...
    }
  ]
}
```

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

# Aufteilung der Sprecher in einer Batch-Transkription

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

## AWS Management Console

1. Melden Sie sich an der [AWS Management Console](#) 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 in den Audioeinstellungen die Option Sprecherpartitionierung (unter der Überschrift „Audioidentifikationstyp“). Sie können optional die Anzahl der Sprecher angeben, die Sie partitionieren möchten, bis zu einem Maximum von 10.

### 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 verwenden wir den [start-transcription-job](#). 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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac \
--output-bucket-name DOC-EXAMPLE-BUCKET \
--output-key my-output-files/ \
--language-code en-US \
--show-speaker-labels TRUE \
```

```
--max-speaker-labels 3
```

Hier ist ein weiteres Beispiel mit dem [start-transcription-job](#) Befehl und einem Anforderungstext, der die Lautsprecherpartitionierung bei diesem Job ermöglicht.

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

Die Datei `my-first-transcription-job.json` enthält den folgenden Anfragetext.

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

## AWS SDK for Python (Boto3)

In diesem Beispiel werden Kanäle mithilfe der AWS SDK for 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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"  
transcribe.start_transcription_job(  
    TranscriptionJobName = job_name,  
    Media = {  
        'MediaFileUri': job_uri  
    },  
    OutputBucketName = 'DOC-EXAMPLE-BUCKET',  
    OutputKey = 'my-output-files/',
```

```
LanguageCode = 'en-US',
Settings = {
    'ShowSpeakerLabels': True,
    'MaxSpeakerLabels': 3
}
)

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

## 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 Management Console](#) an.
2. Wählen Sie im Navigationsbereich Echtzeit-Streaming aus. Blättern Sie nach unten zu den Audioeinstellungen und erweitern Sie dieses Feld, falls es minimiert ist.

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

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

**Transcription** Download full transcript  Start streaming

Transcription output Current language: English, US

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

00:00 of 15:00 min audio stream

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

### 3. Aktivieren Sie die Sprecherpartitionierung.

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

### 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-Stream

In diesem Beispiel wird eine HTTP/2-Anfrage erstellt, die Sprecher in Ihrer Transkriptionsausgabe trennt. Weitere Informationen zur Verwendung von HTTP/2-Streaming mit finden Sie unter [Amazon Transcribe Einrichten eines HTTP/2-Streams](#). 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-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 eine Tagebuchausgabe (Batch)

Hier ist ein Ausgabebeispiel für eine Batch-Transkription mit aktivierter Tagebuchfunktion.

```
{
  "jobName": "my-first-transcription-job",
  "accountId": "111122223333",
  "results": {
    "transcripts": [
      {
        "transcript": "I've been on hold for an hour. Sorry about that."
      }
    ],
    "speaker_labels": {
      "channel_label": "ch_0",
      "speakers": 2,
      "segments": [
        {
          "start_time": "4.87",
          "speaker_label": "spk_0",
          "end_time": "6.88",
          "items": [
            {
              "start_time": "4.87",
              "speaker_label": "spk_0",
              "end_time": "5.02"
            },
            {
              "start_time": "5.02",
              "speaker_label": "spk_0",
              "end_time": "5.17"
            }
          ]
        }
      ]
    }
  }
}
```

```
        {
            "start_time": "5.17",
            "speaker_label": "spk_0",
            "end_time": "5.29"
        },
        {
            "start_time": "5.29",
            "speaker_label": "spk_0",
            "end_time": "5.64"
        },
        {
            "start_time": "5.64",
            "speaker_label": "spk_0",
            "end_time": "5.84"
        },
        {
            "start_time": "6.11",
            "speaker_label": "spk_0",
            "end_time": "6.26"
        },
        {
            "start_time": "6.26",
            "speaker_label": "spk_0",
            "end_time": "6.88"
        }
    ]
},
{
    "start_time": "8.49",
    "speaker_label": "spk_1",
    "end_time": "9.24",
    "items": [
        {
            "start_time": "8.49",
            "speaker_label": "spk_1",
            "end_time": "8.88"
        },
        {
            "start_time": "8.88",
            "speaker_label": "spk_1",
            "end_time": "9.05"
        },
        {
            "start_time": "9.05",
```

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

```
{
  "start_time": "5.29",
  "speaker_label": "spk_0",
  "end_time": "5.64",
  "alternatives": [
    {
      "confidence": "1.0",
      "content": "hold"
    }
  ],
  "type": "pronunciation"
},
{
  "start_time": "5.64",
  "speaker_label": "spk_0",
  "end_time": "5.84",
  "alternatives": [
    {
      "confidence": "1.0",
      "content": "for"
    }
  ],
  "type": "pronunciation"
},
{
  "start_time": "6.11",
  "speaker_label": "spk_0",
  "end_time": "6.26",
  "alternatives": [
    {
      "confidence": "1.0",
      "content": "an"
    }
  ],
  "type": "pronunciation"
},
{
  "start_time": "6.26",
  "speaker_label": "spk_0",
  "end_time": "6.88",
  "alternatives": [
    {
      "confidence": "1.0",
      "content": "hour"
    }
  ]
}
```

```
    }
  ],
  "type": "pronunciation"
},
{
  "speaker_label": "spk_0",
  "alternatives": [
    {
      "confidence": "0.0",
      "content": "."
    }
  ],
  "type": "punctuation"
},
{
  "start_time": "8.49",
  "speaker_label": "spk_1",
  "end_time": "8.88",
  "alternatives": [
    {
      "confidence": "1.0",
      "content": "Sorry"
    }
  ],
  "type": "pronunciation"
},
{
  "start_time": "8.88",
  "speaker_label": "spk_1",
  "end_time": "9.05",
  "alternatives": [
    {
      "confidence": "0.902",
      "content": "about"
    }
  ],
  "type": "pronunciation"
},
{
  "start_time": "9.05",
  "speaker_label": "spk_1",
  "end_time": "9.24",
  "alternatives": [
    {
```

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

# Transkription von Mehrkanal-Audio

Wenn Ihr Audio zwei Kanäle hat, 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 werden den Kanälen die Bezeichnungen `ch_0` und `ch_1` zugewiesen.

Zusätzlich zu den [standardmäßigen Transkriptabschnitten](#) (`transcriptsunditems`) enthalten Anfragen mit aktivierter Kanalidentifikation einen `channel_labels` Abschnitt. Dieser Abschnitt enthält jede Äußerung oder jedes Satzzeichen, gruppiert nach Kanälen, sowie die zugehörige Kanalbezeichnung, 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
  },
```

Beachten Sie, dass, wenn eine Person auf einem Kanal gleichzeitig mit einer Person auf einem separaten Kanal spricht, sich die Zeitstempel für jeden Kanal überschneiden, während die Personen übereinander sprechen.

Ein vollständiges Beispieltranskript mit Kanalidentifikation finden Sie unter [Beispiel für eine Ausgabe zur Kanalidentifikation \(Batch\)](#).

## Verwendung der Kanalidentifikation bei einer Batch-Transkription

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

### AWS Management Console

1. Melden Sie sich an der [AWS Management Console](#) an.
2. Wählen Sie im Navigationsbereich Transkriptionsaufträge und dann Job erstellen (oben rechts) aus. Dadurch wird die Seite „Jobdetails angeben“ geöffnet.

## 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 auf der Seite „Jobdetails angeben“ alle Felder aus, die Sie einbeziehen möchten, und wählen Sie dann Weiter aus. Dadurch gelangen Sie zur Seite Job konfigurieren — optional.

Wählen Sie im Bereich Audioeinstellungen die Option Kanalidentifikation aus (unter der Überschrift „Audioidentifizierungstyp“).

**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 Job erstellen aus, um Ihren Transkriptionsjob auszuführen.

## AWS CLI

In diesem Beispiel verwenden wir den [start-transcription-job](#). 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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac \  
--output-bucket-name DOC-EXAMPLE-BUCKET \  
--output-key my-output-files/ \  
--language-code en-US \  
--settings ChannelIdentification=true
```

Hier ist ein weiteres Beispiel mit dem [start-transcription-job](#) Befehl und einem Anforderungstext, der die Kanalidentifikation mit diesem Job ermöglicht.

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

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

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

## AWS SDK for Python (Boto3)

In diesem Beispiel wird der verwendet AWS SDK for Python (Boto3), um Kanäle mithilfe der Methode [start\\_transcription\\_job](#) zu identifizieren. 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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
transcribe.start_transcription_job(
    TranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'DOC-EXAMPLE-BUCKET',
    OutputKey = 'my-output-files/',
    LanguageCode = 'en-US',
    Settings = {
        'ChannelIdentification': True
    }
)

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

```
print(status)
```

## Verwendung der Kanalidentifikation in einer Streaming-Transkription

Um Kanäle in einer Streaming-Transkription zu identifizieren, können Sie HTTP/2 verwenden oder WebSockets; Beispiele finden Sie im Folgenden:

### HTTP/2-Stream HTTP/2-

In diesem Beispiel wird eine HTTP/2-Anfrage erstellt, die Kanäle in Ihrer Transkriptionsausgabe trennt. Weitere Informationen zur Verwendung von HTTP/2-Streaming mit Amazon Transcribe finden Sie unter [Einrichten eines HTTP/2-Streams](#). Weitere Informationen zu Parametern und Headern, die spezifisch für sind Amazon Transcribe, 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 vorsignierte URL erstellt, die die Kanäle in Ihrer Transkriptionsausgabe voneinander trennt. Für eine bessere Lesbarkeit werden Zeilenumbrüche hinzugefügt. Weitere Informationen zur Verwendung von WebSocket Streams mit Amazon Transcribe finden Sie unter [Einen WebSocket Stream einrichten](#). Weitere Informationen zu 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 für eine Ausgabe zur Kanalidentifikation (Batch)

Hier ist ein Ausgabebeispiel für eine Batch-Transkription mit aktivierter Kanalidentifikation.

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

```
        "alternatives": [  
            {  
                "confidence": "1.0",  
                "content": "I've"  
            }  
        ],  
        "type": "pronunciation"  
    },  
    {  
        "channel_label": "ch_0",  
        "start_time": "5.01",  
        "end_time": "5.16",  
        "alternatives": [  
            {  
                "confidence": "1.0",  
                "content": "been"  
            }  
        ],  
        "type": "pronunciation"  
    },  
    {  
        "channel_label": "ch_0",  
        "start_time": "5.16",  
        "end_time": "5.28",  
        "alternatives": [  
            {  
                "confidence": "1.0",  
                "content": "on"  
            }  
        ],  
        "type": "pronunciation"  
    },  
    {  
        "channel_label": "ch_0",  
        "start_time": "5.28",  
        "end_time": "5.62",  
        "alternatives": [  
            {  
                "confidence": "1.0",  
                "content": "hold"  
            }  
        ],  
        "type": "pronunciation"  
    },  
    },
```

```
{
  "channel_label": "ch_0",
  "start_time": "5.62",
  "end_time": "5.83",
  "alternatives": [
    {
      "confidence": "1.0",
      "content": "for"
    }
  ],
  "type": "pronunciation"
},
{
  "channel_label": "ch_0",
  "start_time": "6.1",
  "end_time": "6.25",
  "alternatives": [
    {
      "confidence": "1.0",
      "content": "an"
    }
  ],
  "type": "pronunciation"
},
{
  "channel_label": "ch_0",
  "start_time": "6.25",
  "end_time": "6.87",
  "alternatives": [
    {
      "confidence": "1.0",
      "content": "hour"
    }
  ],
  "type": "pronunciation"
},
{
  "channel_label": "ch_0",
  "language_code": "en-US",
  "alternatives": [
    {
      "confidence": "0.0",
      "content": "."
    }
  ]
}
```

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

```
        {
            "channel_label": "ch_1",
            "alternatives": [
                {
                    "confidence": "0.0",
                    "content": "."
                }
            ],
            "type": "punctuation"
        }
    ]
},
"number_of_channels": 2
},
"items": [
    {
        "channel_label": "ch_0",
        "start_time": "4.86",
        "end_time": "5.01",
        "alternatives": [
            {
                "confidence": "1.0",
                "content": "I've"
            }
        ],
        "type": "pronunciation"
    },
    {
        "channel_label": "ch_0",
        "start_time": "5.01",
        "end_time": "5.16",
        "alternatives": [
            {
                "confidence": "1.0",
                "content": "been"
            }
        ],
        "type": "pronunciation"
    },
    {
        "channel_label": "ch_0",
        "start_time": "5.16",
        "end_time": "5.28",
```

```
    "alternatives": [
      {
        "confidence": "1.0",
        "content": "on"
      }
    ],
    "type": "pronunciation"
  },
  {
    "channel_label": "ch_0",
    "start_time": "5.28",
    "end_time": "5.62",
    "alternatives": [
      {
        "confidence": "1.0",
        "content": "hold"
      }
    ],
    "type": "pronunciation"
  },
  {
    "channel_label": "ch_0",
    "start_time": "5.62",
    "end_time": "5.83",
    "alternatives": [
      {
        "confidence": "1.0",
        "content": "for"
      }
    ],
    "type": "pronunciation"
  },
  {
    "channel_label": "ch_0",
    "start_time": "6.1",
    "end_time": "6.25",
    "alternatives": [
      {
        "confidence": "1.0",
        "content": "an"
      }
    ],
    "type": "pronunciation"
  },
  },
```

```
{
  "channel_label": "ch_0",
  "start_time": "6.25",
  "end_time": "6.87",
  "alternatives": [
    {
      "confidence": "1.0",
      "content": "hour"
    }
  ],
  "type": "pronunciation"
},
{
  "channel_label": "ch_0",
  "alternatives": [
    {
      "confidence": "0.0",
      "content": "."
    }
  ],
  "type": "punctuation"
},
{
  "channel_label": "ch_1",
  "start_time": "8.5",
  "end_time": "8.89",
  "alternatives": [
    {
      "confidence": "1.0",
      "content": "Sorry"
    }
  ],
  "type": "pronunciation"
},
{
  "channel_label": "ch_1",
  "start_time": "8.89",
  "end_time": "9.06",
  "alternatives": [
    {
      "confidence": "0.9176",
      "content": "about"
    }
  ]
},
```

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

# 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 in der Lage ist, die Sprache zu identifizieren, desto geringer ist die Gefahr eines Datenverlustes in den ersten Sekunden des Streams.

## Important

Batch- und Streaming-Transkriptionen unterstützen verschiedene Sprachen. Einzelheiten finden Sie in der Spalte Dateneingabe in der [Tabelle der unterstützten 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 Management Console, AWS CLI, und dem AWS Python-SDK finden Sie unter [Sprachidentifizierung mit Batch-Transkriptionen](#).

## Identifizierung von Sprachen in mehrsprachigen Audiodateien

Die mehrsprachige Identifizierung ist für mehrsprachige Mediendateien gedacht und bietet Ihnen ein Transkript, das alle [unterstützten Sprachen](#) widerspiegelt, 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

Mehrsprachige Transkripte bieten eine Zusammenfassung der erkannten Sprachen und der Gesamtzeit, in der jede Sprache in Ihren Medien gesprochen wird. 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) werden nur die Sprachen verwendet, die Sie angeben, wenn Sie Ihr Audio der richtigen Sprache zuordnen. Dadurch kann die Sprachenerkennung beschleunigt und die Genauigkeit bei der Zuweisung des richtigen Sprachdialekts verbessert werden. Amazon Transcribe

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, Amazon Transcribe ist es wahrscheinlich zh-CNfr-FR, dass Ihre Medien Amazon Transcribe mit Deutsch (de-DE) übereinstimmen und eine deutschsprachige 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 von Sprachidentifizierung 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 Sprachenidentifikation bei aktivierter Inhaltsredaktion verwenden und Ihr Audio andere Sprachen als US-Englisch (en-US) oder US-Spanisch (es-US) enthält,

wird in Ihrem Transkript nur der Inhalt 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 handeln-`en-AU` handelt, 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 Management Console- oder AWS CLI-SDKs verwenden. Beispiele finden Sie im Folgenden:

### AWS Management Console

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

- 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)

- 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 [start-transcription-job](#) Befehl und `IdentifyLanguage` der Parameter verwendet. Weitere Informationen finden 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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac \
--output-bucket-name DOC-EXAMPLE-BUCKET \
--output-key my-output-files/ \
--identify-language \ (or --identify-multiple-languages) \
--language-options "en-US" "hi-IN"
```

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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac \
--output-bucket-name DOC-EXAMPLE-BUCKET \
--output-key my-output-files/ \
--identify-language \ (or --identify-multiple-languages) \
--language-options "en-US" "hi-IN" \
--language-id-settings en-US=VocabularyName=my-en-US-vocabulary,en-US=VocabularyFilterName=my-en-US-vocabulary-filter,en-US=LanguageModelName=my-en-US-language-model,hi-IN=VocabularyName=my-hi-IN-vocabulary,hi-IN=VocabularyFilterName=my-hi-IN-vocabulary-filter
```

Hier ist ein weiteres Beispiel mit dem [start-transcription-job](#) Befehl und einem Anforderungstext, der die Sprache identifiziert.

```
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 Anfragetext.

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

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

```
}  
}
```

## AWS SDK for Python (Boto3)

In diesem Beispiel wird die AWS SDK for Python (Boto3) Sprache Ihrer Datei mithilfe des [IdentifyLanguage](#) Arguments für die Methode `start_transcription_job` identifiziert. Weitere Informationen finden 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 SDKs AWS](#)

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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"  
transcribe.start_transcription_job(  
    TranscriptionJobName = job_name,  
    Media = {  
        'MediaFileUri': job_uri  
    },  
    OutputBucketName = 'DOC-EXAMPLE-BUCKET',  
    OutputKey = 'my-output-files/',  
    MediaFormat = 'flac',  
    IdentifyLanguage = True, (or IdentifyMultipleLanguages = True),  
    LanguageOptions = [  
        'en-US', 'hi-IN'  
    ]  
)  
  
while True:  
    status = transcribe.get_transcription_job(TranscriptionJobName = job_name)  
    if status['TranscriptionJob']['TranscriptionJobStatus'] in ['COMPLETED', 'FAILED']:  
        break  
    print("Not ready yet...")
```

```
time.sleep(5)
print(status)
```

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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
transcribe.start_transcription_job(
    TranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'DOC-EXAMPLE-BUCKET',
    OutputKey = 'my-output-files/',
    MediaFormat='flac',
    IdentifyLanguage=True, (or IdentifyMultipleLanguages=True)
    LanguageOptions = [
        'en-US', 'hi-IN'
    ],
    LanguageIdSettings={
        'en-US': {
            'VocabularyName': 'my-en-US-vocabulary',
            'VocabularyFilterName': 'my-en-US-vocabulary-filter',
            'LanguageModelName': 'my-en-US-language-model'
        },
        'hi-IN': {
            'VocabularyName': 'my-hi-IN-vocabulary',
            'VocabularyFilterName': 'my-hi-IN-vocabulary-filter'
        }
    }
)

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

```
time.sleep(5)
print(status)
```

## Sprachidentifizierung mit Streaming-Transkriptionen

Die Streaming-Sprachidentifizierung kann die in Ihrem Mediendatenstrom vorherrschende Sprache identifizieren. Amazon Transcribe benötigt mindestens drei Sekunden Sprache, 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 Services wie Amazon Transcribenutzen, 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, wählt Amazon Transcribe die Sprache aus, die den von Ihnen angegebenen Sprachcodes am nächsten kommt. Es wird dann eine Transkription in dieser Sprache erstellt. Wenn Ihre Medien beispielsweise in US-Englisch (en-US) vorliegen und Sie Amazon Transcribe mit den Sprachcodes zh-CN, fr-FR und de-DE versehen, wird Amazon Transcribe Ihre Medien wahrscheinlich mit Deutsch (de-DE) abgleichen und eine deutschsprachige Transkription erstellen. 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, kann Amazon Transcribe die vorherrschende 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.

## Identifizierung von Sprachen in mehrsprachigen Audiodateien

Die Identifizierung mehrerer Sprachen ist für mehrsprachige Streams gedacht 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

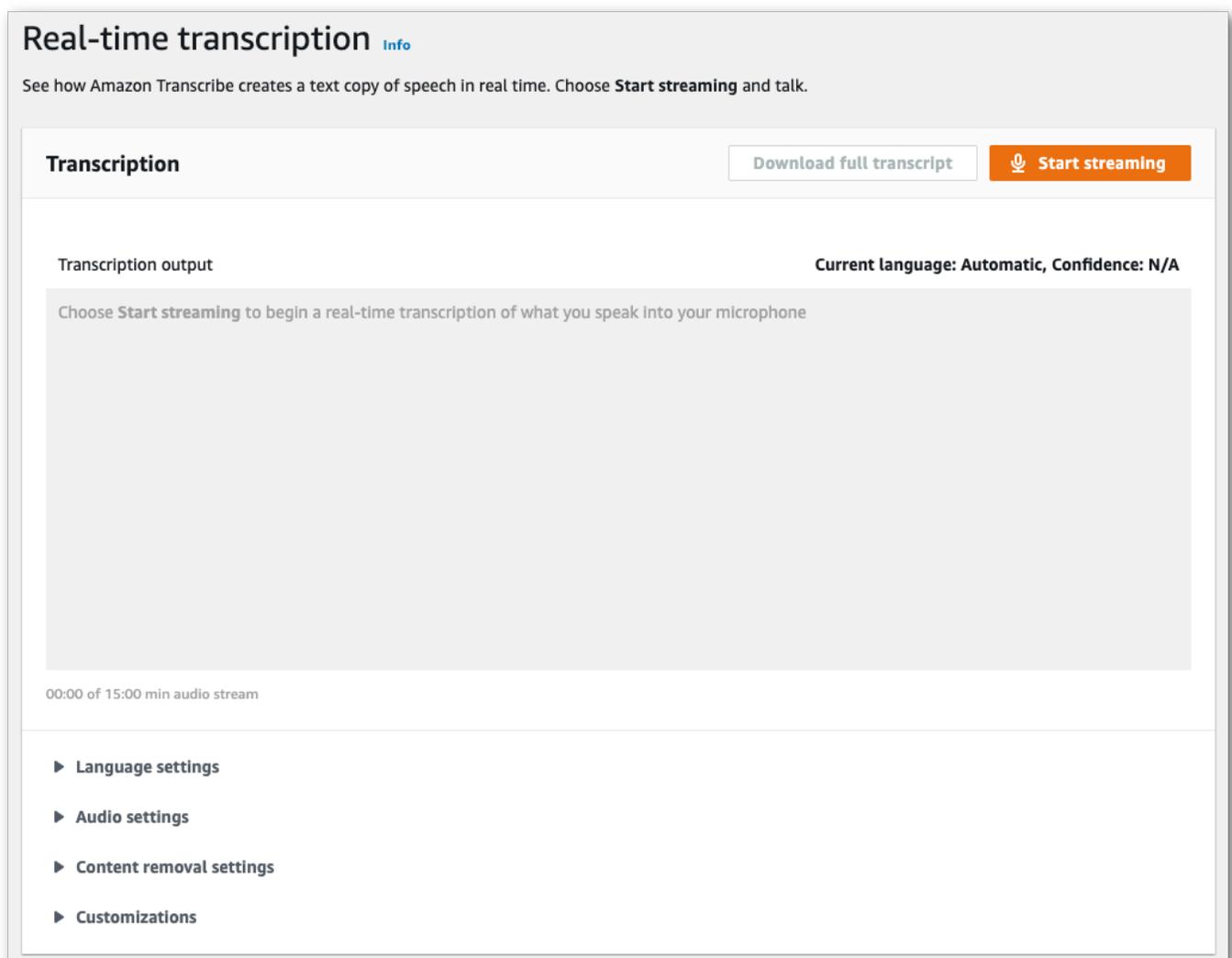
Schwärzen und benutzerdefinierte Sprachmodelle werden derzeit bei mehrsprachiger Identifizierung nicht unterstützt.

## Sprachidentifizierung mit Streaming-Medien nutzen

Sie können die automatische Sprachidentifizierung in einer Streaming-Transkription verwenden, indem Sie AWS Management Console, HTTP/2 oder WebSockets verwenden. Beispiele finden Sie im Folgenden:

### AWS Management Console

1. Melden Sie sich an der [AWS Management Console](#) an.
2. Wählen Sie im Navigationsbereich Echtzeit-Streaming 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 fr-CA 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) X    French, CA (fr-CA) X

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

Choose language ▲

None
English, US (en-US)
French, CA (fr-CA)

► **Customizations**

- 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-Stream

In diesem Beispiel wird eine HTTP/2-Anfrage mit aktivierter Sprachidentifizierung erstellt. Weitere Informationen zur Verwendung von HTTP/2-Streaming mit Amazon Transcribe finden Sie unter [Einrichten eines HTTP/2-Streams](#). 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-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 mit aktivierter Identifizierung mehrerer Sprachen erstellt. Weitere Informationen zur Verwendung von HTTP/2-Streaming mit Amazon Transcribe finden Sie unter [Einrichten eines HTTP/2-Streams](#). 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-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, werden im Abschnitt [Allgemeine Parameter](#) aufgeführt.

## WebSocket-Stream

In diesem Beispiel wird eine vordefinierte URL erstellt, die die Sprachidentifikation in einem WebSocket-Stream verwendet. Für eine bessere Lesbarkeit werden Zeilenumbrüche hinzugefügt. Weitere Informationen 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  
&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 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  
&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, werden im Abschnitt [Allgemeine Parameter](#) aufgeführt.

# Alternative Transkriptionen

Beim Amazon Transcribe Transkribieren von Audio werden verschiedene Versionen desselben Transkripts erstellt und jeder Version wird ein Konfidenzwert zugewiesen. In 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 wählen, ob Sie bis zu 10 alternative Transkriptionen zurückgeben möchten. Wenn Sie eine größere Anzahl von Alternativen als die angegebene Anzahl Amazon Transcribe angeben, wird nur die tatsächliche Anzahl der Alternativen zurückgegeben.

Alle Alternativen befinden sich in derselben Transkriptionsausgabedatei und werden auf Segmentebene dargestellt. Segmente sind natürliche Sprachpausen, z. B. ein Sprecherwechsel oder eine Audiopause.

Alternative Transkriptionen sind nur für Batch-Transkriptionen verfügbar.

Ihre Transkriptionsausgabe ist folgendermaßen strukturiert. Die Ellipsen ( . . . ) geben Sie in den Codebeispielen an, wo der Inhalt der Kürze halber entfernt wurde.

1. Eine vollständige endgültige 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 vorherigen `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 sortiert.

```

"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 Ihrer Transkriptionsausgabe.

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

## Alternative Transkriptionen anfordern

Sie können alternative Transkriptionen mit den AWS Management Console, AWS CLI, oder AWS SDKs anfordern. Beispiele finden Sie im Folgenden:

### AWS Management Console

1. Melden Sie sich an der [AWS Management Console](#) an.
2. Wählen Sie im Navigationsbereich Transkriptionsaufträge und dann Job erstellen (oben rechts) aus. Dadurch wird die Seite „Jobdetails angeben“ geöffnet.

## 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**.

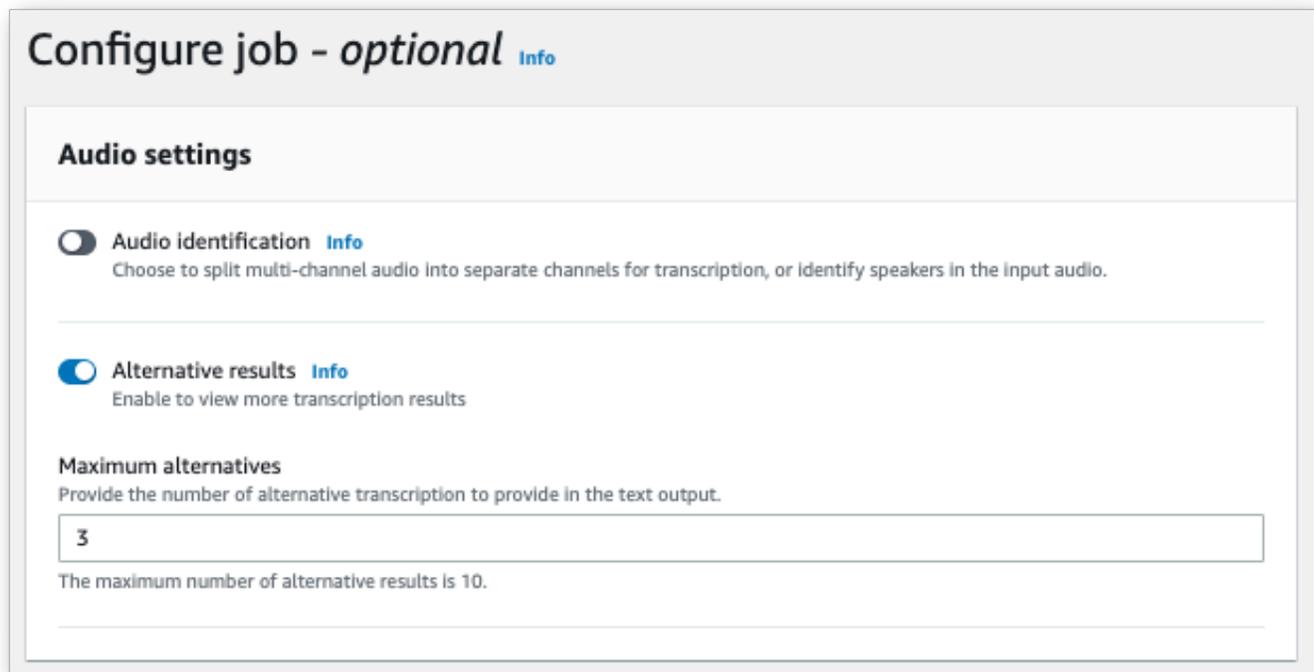
**Language**

Choose the language of the input audio.

► **Additional settings**

3. Füllen Sie auf der Seite „Jobdetails angeben“ alle Felder aus, die Sie einbeziehen möchten, und wählen Sie dann Weiter aus. Dadurch gelangen Sie zur Seite Job konfigurieren — optional.

Wählen Sie Alternative Ergebnisse aus und geben Sie die maximale Anzahl alternativer Transkriptionsergebnisse an, die Sie in Ihrem Transkript haben möchten.



4. Wählen Sie Job erstellen aus, um Ihren Transkriptionsjob auszuführen.

## AWS CLI

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

Beachten Sie, dass Sie, wenn Sie `ShowAlternatives=true` in Ihrer Anfrage angeben, auch angeben müssen `MaxAlternatives`.

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

Hier ist ein weiteres Beispiel für die Verwendung des [start-transcription-job](#) Befehls und ein 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-transcription-job.json` enthält den folgenden Anforderungstext.

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

## AWS SDK for Python (Boto3)

Im folgenden Beispiel wird das verwendete AWS SDK for Python (Boto3), um alternative Transkriptionen anzufordern, indem das `ShowAlternatives` Argument für die Methode [start\\_transcription\\_job](#) verwendet wird. Weitere Informationen erhalten Sie unter [StartTranscriptionJob](#) und [ShowAlternatives](#).

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

Beachten Sie, dass Sie, wenn Sie dies 'ShowAlternatives': True in Ihrer Anfrage angeben, auch angeben müssen `MaxAlternatives`.

```
from __future__ import print_function  
import time  
import boto3  
transcribe = boto3.client('transcribe', 'us-west-2')  
job_name = "my-first-transcription-job"  
job_uri = "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"  
transcribe.start_transcription_job(  

```

```
TranscriptionJobName = job_name,
Media = {
    'MediaFileUri': job_uri
},
OutputBucketName = 'DOC-EXAMPLE-BUCKET',
OutputKey = 'my-output-files/',
LanguageCode = 'en-US',
Settings = {
    'ShowAlternatives':True,
    'MaxAlternatives':4
}
)

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

# Verbesserung der Transkriptionsgenauigkeit mit benutzerdefinierten Vokabeln und benutzerdefinierten Sprachmodellen

Wenn Ihre Medien domänenspezifische oder nicht standardmäßige Begriffe wie Markennamen, Akronyme, Fachausdrücke und Fachjargon enthalten, werden diese Begriffe in Ihrer Transkriptionsausgabe Amazon Transcribe möglicherweise nicht korrekt erfasst.

Um Transkriptionsungenauigkeiten zu korrigieren und Ihre Ausgabe an Ihren spezifischen Anwendungsfall anzupassen, können Sie [Benutzerdefinierte Vokabulare](#) und erstellen [Benutzerdefinierte Sprachmodelle](#).

- [Benutzerdefinierte Vokabulare](#) wurden entwickelt, um sowohl die Erkennung als auch die Formatierung bestimmter Wörter in allen Kontexten zu optimieren und zu verbessern. Dies beinhaltet die Versorgung Amazon Transcribe mit Wörtern und optional mit Aussprache- und Darstellungsformen.

Wenn bestimmte Begriffe in Ihren Transkripten nicht Amazon Transcribe korrekt wiedergegeben werden, können Sie eine benutzerdefinierte Vokabeldatei erstellen, die angibt, Amazon Transcribe wie diese Begriffe angezeigt werden sollen. Dieser wortspezifische Ansatz eignet sich am besten für die Korrektur von Begriffen wie Markennamen und Akronymen.

- [Benutzerdefinierte Sprachmodelle](#) sind so konzipiert, dass sie den Kontext erfassen, der mit Begriffen verbunden ist. Dies beinhaltet die Bereitstellung Amazon Transcribe einer großen Menge an domänenspezifischen 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 Amazon Transcribe Ihre domänenspezifische Sprache unterrichtet. Ein benutzerdefiniertes Sprachmodell kann beispielsweise lernen, wann „Scholle“ (Eisscholle) im Vergleich zu „Fluss“ (lineare Strömung) verwendet wird.

Dieser kontextsensitive Ansatz eignet sich am besten für die Transkription großer Mengen domänenspezifischer Sprache. Benutzerdefinierte Sprachmodelle können im Vergleich zu benutzerdefinierten Vokabeln allein zu erheblichen Genauigkeitsverbesserungen führen. Wenn Sie Batch-Transkriptionen verwenden, können Sie Ihrer Anfrage sowohl ein benutzerdefiniertes Sprachmodell als auch ein benutzerdefiniertes Vokabular hinzufügen.

**i** Tip

Verwenden Sie benutzerdefinierte Vokabeln in Verbindung mit Ihren benutzerdefinierten Sprachmodellen, um die höchste Transkriptionsgenauigkeit zu erreichen.

Eine Videodemo zum Erstellen eines benutzerdefinierten Vokabels mithilfe von finden Sie unter [Verwenden eines benutzerdefinierten Vokabels](#).AWS Management Console

Eine Videodemo zur Erstellung und Verwendung benutzerdefinierter Sprachmodelle finden Sie unter [Verwenden von benutzerdefinierten Sprachmodellen \(CLM\) zur Steigerung der Transkriptionsgenauigkeit](#).

**i** Tauchen Sie mit demAWS Machine Learning Blog tiefer ein

Benutzerdefinierte Vokabeln:

- [Live-Transkriptionen von F1-Rennen mitAmazon Transcribe](#)

Benutzerdefinierte Sprachmodelle:

- [Erstellung benutzerdefinierter Sprachmodelle zur speech-to-text LeistungssteigerungAmazon Transcribe](#)
- [Steigern Sie die Transkriptionsgenauigkeit von Vorlesungen mit benutzerdefinierten Sprachmodellen fürAmazon 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 Management Console, 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.

**ℹ Tip**

Sie können Ihr benutzerdefiniertes Vokabular mit dem AWS Management Console testen. Sobald Ihr benutzerdefiniertes Vokabular einsatzbereit ist, melden Sie sich bei an AWS Management Console, wählen Sie Echtzeit-Transkription, scrollen Sie zu Anpassungen, schalten Sie Benutzerdefiniertes Vokabular ein und wählen Sie Ihr benutzerdefiniertes Vokabular aus der Dropdownliste 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, die AWS Management Console, die 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 finden 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 <b>Andorra-la-Vella</b> oder <b>Los-Angeles</b> .			Legt fest, wie Ihr Eintrag in Ihrer Transkriptionsausgabe aussehen soll. Zum Beispiel ist <b>Andorra-la-Vella</b> in der Spalte Phrase gleich <b>Andorra la Vella</b> 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“ <b>C.L.I.</b> (nicht <b>C.L.I</b> ) und „ABCs“ ist <b>A.B.C.-s</b> (nicht <b>A.B.C-s</b> ).			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
<p>einem Akronym besteht, müssen diese beiden Bestandteile durch einen Bindestrich getrennt werden. Zum Beispiel ist „DynamoDB“</p> <p><b>Dynamo-D.B.</b> .</p>			
<p>In dieser Spalte dürfen keine Ziffern stehen; die Zahlen müssen ausgeschrieben werden. Zum Beispiel ist „VX02Q“</p> <p><b>V.X.-zero-two-Q.</b></p>			

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.
- 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) im Format LF speichern. Wenn Sie ein anderes Format verwenden, z. B. CRLF, kann Ihr benutzerdefiniertes Vokabular nicht verarbeitet werden.
- 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	<b>[TAB]</b> SoundsLike	<b>[TAB]</b> IPA	<b>[TAB]</b> DisplayAs
Los-Angeles	<b>[TAB]</b>	<b>[TAB]</b>	<b>[TAB]</b> Los Angeles
Eva-Maria	<b>[TAB]</b>	<b>[TAB]</b>	<b>[TAB]</b>
A.B.C.-s	<b>[TAB]</b>	<b>[TAB]</b>	<b>[TAB]</b> ABCs
amazon-dot-com	<b>[TAB]</b>	<b>[TAB]</b>	<b>[TAB]</b> amazon.com
C.L.I.	<b>[TAB]</b>	<b>[TAB]</b>	<b>[TAB]</b> CLI
Andorra-la-Vella	<b>[TAB]</b>	<b>[TAB]</b>	<b>[TAB]</b> Andorra la Vella
Dynamo-D.B.	<b>[TAB]</b>	<b>[TAB]</b>	<b>[TAB]</b> DynamoDB
V.X.-zero-two	<b>[TAB]</b>	<b>[TAB]</b>	<b>[TAB]</b> VX02
V.X.-zero-two-Q.	<b>[TAB]</b>	<b>[TAB]</b>	<b>[TAB]</b> VX02Q

## Erstellen von benutzerdefinierten Vokabulartabellen

In den folgenden Beispielen erfahren Sie Amazon Transcribe, wie Sie eine benutzerdefinierte Vokabeltabelle für die Verwendung mit bearbeiten können:

### AWS Management Console

1. Melden Sie sich an der [AWS Management Console](#) 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.

The screenshot shows the 'Custom vocabulary' page in the Amazon Transcribe console. At the top, there's a breadcrumb 'Amazon Transcribe > Custom vocabulary' and a title 'Custom vocabulary' with an 'Info' link. Below the title is a sub-header 'Use custom vocabularies to improve transcription accuracy. Learn more' with a link. The main content is divided into two columns under an 'Overview' section. The left column is titled '1. Create custom vocabulary' and describes creating a vocabulary by uploading a file or adding phrases, mentioning templates (.csv, .txt). The right column is titled '2. Apply to Real-time or batch transcription' and explains that the created vocabulary can be applied to real-time or batch transcription jobs. Below this is a 'Manage vocabularies' section with a search bar, a filter dropdown set to 'All', and a table with columns for Name, Language, Last modified, and Status. The table is currently empty, showing 'Empty resources' and 'No resources to display', with a 'Create vocabulary' button at the bottom.

Sie gelangen auf die Seite Vokabular erstellen. Geben Sie einen Namen für Ihr neues benutzerdefiniertes Vokabular ein.

Hier haben Sie drei Möglichkeiten:

- a. 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**

 ▼

### 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 <a href="#">↗</a>	SoundsLike (optional) <a href="#">↗</a>	IPA (optional) <a href="#">↗</a>	DisplayAs (optional) <a href="#">↗</a>
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 ▼

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

<input type="checkbox"/>	Phrase <a href="#">✎</a> ▼	SoundsLike - optional <a href="#">✎</a> ▼	IPA - optional <a href="#">✎</a> ▼	DisplayAs - optional <a href="#">✎</a> ▼
<input type="checkbox"/>	Amazon-E.-C.-two	-	-	Amazon EC2
<input type="checkbox"/>	Amazon-S.-three	-	-	Amazon S3
<input type="checkbox"/>	Amazon-elasticashe	-	-	Amazon ElastiCache
<input type="checkbox"/>	Amazon-sagemaker	-	-	Amazon SageMaker
<input type="checkbox"/>	A.-W.-S.-iam	-	-	AWS IAM
<input type="checkbox"/>	A.-W.-S.-I.-o.-T.	-	-	AWS IoT
<input type="checkbox"/>	A.-W.-S.-W.-A.-F.	-	-	AWS WAF
<input type="checkbox"/>	c.-plus-plus	-	-	C++
<input type="checkbox"/>	nice-d.-c.-v.	-	-	NICE DCV
<input type="checkbox"/>	w.-w.-w.-dot-amazon-dot-com	-	-	www.amazon.com

Add row

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 ▼

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

<input type="checkbox"/>	Phrase <a href="#">✎</a> ▼	SoundsLike - optional <a href="#">✎</a> ▼	IPA - optional <a href="#">✎</a> ▼	DisplayAs - optional <a href="#">✎</a> ▼
<input type="checkbox"/>	<input type="text" value="Amazon-E.-C. two"/> <span>✕</span> <span>✓</span> <span style="color: red;">⚠ Phrase contains unsupported characters (" "). Phrase contains a formatting error.</span>	-	-	Amazon EC2
<input type="checkbox"/>	Amazon-S.-three	-	-	Amazon S3
<input type="checkbox"/>	c.-plus-plus	-	-	C++
<input type="checkbox"/>	w.-w.-w.-dot-amazon-dot-com	-	-	www.amazon.com

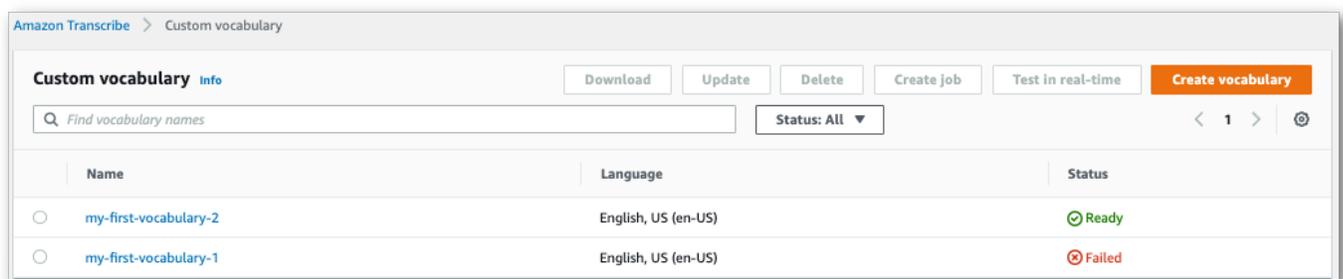
Add row

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.



- 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.

Name	Status	Language	Modified
my-first-vocabulary-1	Failed	English, US (en-US)	October 13 2021, 10:03 (UTC-07:00)

## 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 Management Console benutzerdefinierte Vokabular aus der Dropdownliste aus.

```
aws transcribe create-vocabulary \
--vocabulary-name my-first-vocabulary \
--vocabulary-file-uri s3://DOC-EXAMPLE-BUCKET/my-vocabularies/my-vocabulary-file.txt \
--language-code en-US
```

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 Anfragetext.

```
{
```

```
"VocabularyName": "my-first-vocabulary",
"VocabularyFileUri": "s3://DOC-EXAMPLE-BUCKET/my-vocabularies/my-vocabulary-
table.txt",
"LanguageCode": "en-US"
}
```

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 for Python (Boto3)

In diesem Beispiel wird mithilfe der AWS SDK for 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 Management Console 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 SDKs AWS](#)

```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
vocab_name = "my-first-vocabulary"
response = transcribe.create_vocabulary(
    LanguageCode = 'en-US',
    VocabularyName = vocab_name,
    VocabularyFileUri = 's3://DOC-EXAMPLE-BUCKET/my-vocabularies/my-vocabulary-
table.txt'
)
```

```
while True:
    status = transcribe.get_vocabulary(VocabularyName = vocab_name)
    if status['VocabularyState'] in ['READY', 'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
print(status)
```

### Note

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 TranscribeBeispiele für identitätsbasierte -Richtlinien](#). Amazon Transcribe

## Erstellen eines benutzerdefinierten Vokabulars mithilfe einer Liste

### Important

[Benutzerdefinierte Vokabeln im Listenformat sind veraltet. Wenn Sie also ein neues benutzerdefiniertes Vokabular erstellen, empfehlen wir dringend, das Tabellenformat zu verwenden.](#)

Sie können mithilfe der AWS SDKsAWS Management Console,AWS CLI, oder benutzerdefinierte Vokabeln aus Listen erstellen.

- **AWS Management Console:** Sie müssen eine Textdatei mit Ihrem benutzerdefinierten Wortschatz erstellen und hochladen. Sie können zeilengetrennte oder kommagetrennte Einträge verwenden. Beachten Sie, dass Ihre Liste als Textdatei (\*.txt) im LF Format gespeichert werden muss. Wenn Sie ein anderes Format verwenden, z. B. CRLF wird Ihr benutzerdefiniertes Vokabular von nicht akzeptiertAmazon Transcribe.
- **AWS CLIund AWSSDKs:** Sie müssen Ihr benutzerdefiniertes Vokabular als durch Kommas getrennte Einträge in Ihren API-Aufruf aufnehmen, indem Sie die Flagge verwenden. [Phrases](#)

Wenn ein Eintrag mehrere Wörter enthält, müssen Sie jedes Wort mit einer Silbentrennung trennen. Sie geben beispielsweise „Los Angeles“ als **Los-Angeles** und „Andorra la Vella“ als an. **Andorra-la-Vella**

Hier finden Sie Beispiele für die beiden gültigen Listenformate. Methodenspezifische [Erstellen benutzerdefinierter Vokabellisten](#) Beispiele finden Sie unter.

- Durch Kommas getrennte Einträge:

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

- Zeilengetrennte Einträge:

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

#### Important

Sie können nur Zeichen verwenden, die für Ihre Sprache unterstützt werden. Einzelheiten entnehmen Sie dem [Zeichensatz](#) Ihrer Sprache.

Benutzerdefinierte Vokabellisten werden bei der [CreateMedicalVocabulary](#) Operation nicht unterstützt. Wenn Sie ein benutzerdefiniertes medizinisches Vokabular erstellen, müssen Sie ein Tabellenformat verwenden. Anweisungen [Erstellen eines benutzerdefinierten Vokabulars mithilfe einer Tabelle](#) finden Sie unter.

## Erstellen benutzerdefinierter Vokabellisten

Um eine benutzerdefinierte Vokabelliste für die Verwendung mit zu verarbeiten Amazon Transcribe, sehen Sie sich die folgenden Beispiele an:

### AWS CLI

In diesem Beispiel wird der Befehl [create-vocabulary](#) mit einer benutzerdefinierten Vokabeldatei im Listenformat 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 ist 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 Anfragetext.

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

Sobald der `VocabularyState` Wert von `PENDING` auf geändert `READY` wurde, ist Ihr benutzerdefinierter Wortschatz mit einer Transkription einsatzbereit. Um den aktuellen Status Ihres eine benutzerdefinierte Vokabular zu überprüfen

```
aws transcribe get-vocabulary \  
--vocabulary-name my-first-vocabulary
```

## AWS SDK for Python (Boto3)

In diesem Beispiel wird der verwendet AWS SDK for Python (Boto3), um mithilfe der Methode [create\\_vocabulary ein benutzerdefiniertes Vokabular](#) aus einer Liste zu erstellen. Weitere Informationen finden Sie unter [CreateVocabulary](#).

Weitere Beispiele für die Verwendung der AWS SDKs, einschließlich funktionspezifischer, szenariospezifischer und dienstübergreifender Beispiele, finden Sie in diesem Kapitel. [Codebeispiele für Amazon Transcribe mit SDKs AWS](#)

```
from __future__ import print_function
```

```
import time
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
vocab_name = "my-first-vocabulary"
response = transcribe.create_vocabulary(
    LanguageCode = 'en-US',
    VocabularyName = vocab_name,
    Phrases = [
        'CLI', 'Eva-Maria', 'ABCs'
    ]
)

while True:
    status = transcribe.get_vocabulary(VocabularyName = vocab_name)
    if status['VocabularyState'] in ['READY', 'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
print(status)
```

### Note

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 in Amazon Transcribe finden Sie unter [Amazon TranscribeBeispiele für identitätsbasierte -Richtlinien](#).

## Verwenden eines benutzerdefinierten einen benutzerdefinierten Vokabular

Sobald Ihr benutzerdefiniertes Vokabular erstellt wurde, können Sie es in Ihre Transkriptionsanfragen aufnehmen. Beispiele finden Sie in den folgenden Abschnitten.

Die Sprache des benutzerdefinierten Vokabulars, das Sie in Ihrer Anfrage angeben, muss mit dem Sprachcode übereinstimmen, den Sie für Ihr Medium 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

Um ein benutzerdefiniertes Vokabular mit einer Batch-Transkription zu verwenden, finden Sie im Folgenden Beispiele:

### AWS Management Console

1. Melden Sie sich an der [AWS Management Console](#) an.
2. Wählen Sie im Navigationsbereich Transkriptionsaufträge und dann Job erstellen (oben rechts) aus. Dadurch wird die Seite „Jobdetails angeben“ geöffnet.

### 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**

Nennen Sie Ihren Job und geben Sie Ihre Eingabemedien an. Fügen Sie optional weitere Felder hinzu und wählen Sie dann Weiter.

3. Schalten Sie unten auf der Seite „Job konfigurieren“ im Bereich „Anpassung“ die Option Benutzerdefiniertes Vokabular ein.

## 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](#). [↗](#)

[Cancel](#) [Previous](#) [Create job](#)

#### 4. Wählen Sie Ihr benutzerdefiniertes Vokabular aus dem Dropdown-Menü aus.

Wählen Sie Job erstellen aus, um Ihren Transkriptionsjob auszuführen.

### AWS CLI

In diesem Beispiel werden der [start-transcription-job](#) Befehl und der `Settings` Parameter mit dem `VocabularyName` Unterparameter 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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac \  
--output-bucket-name DOC-EXAMPLE-BUCKET \  
--output-key my-output-files/ \  
--language-code en-US \  
--settings VocabularyName=my-first-vocabulary
```

Hier ist ein weiteres Beispiel für die Verwendung des [start-transcription-job](#) Befehls und ein Anforderungstext, der Ihr benutzerdefiniertes Vokabular für diesen Job 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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"  
  },  
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",  
  "OutputKey": "my-output-files/",  
  "LanguageCode": "en-US",  
  "Settings": {  
    "VocabularyName": "my-first-vocabulary"  
  }  
}
```

## AWS SDK for Python (Boto3)

In diesem Beispiel wird das verwendete AWS SDK for Python (Boto3), um mithilfe des `Settings` Arguments für die Methode `start_transcription_job` ein benutzerdefiniertes Vokabular einzubinden. Weitere Informationen erhalten Sie unter [StartTranscriptionJob](#) und [Settings](#).

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

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

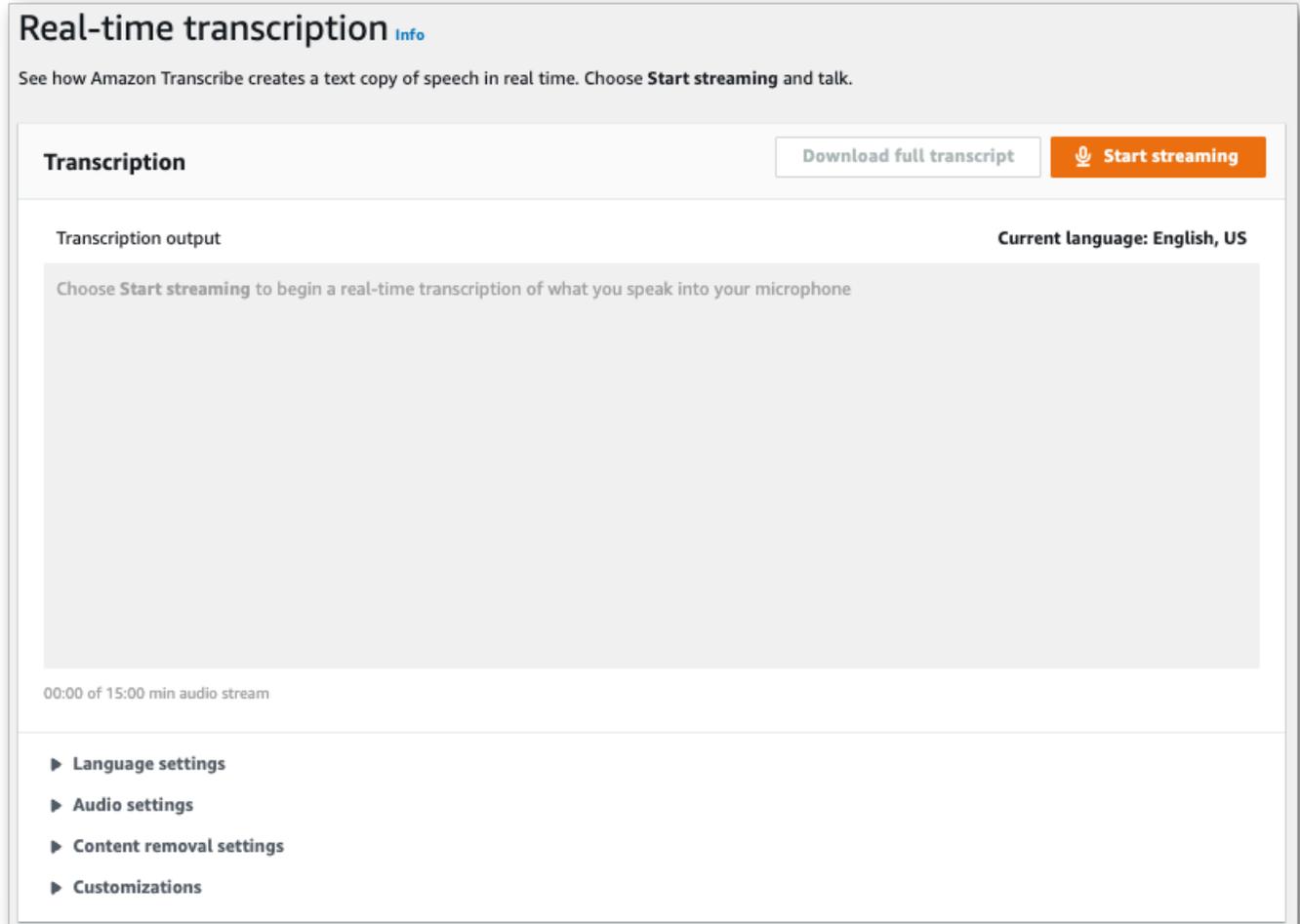
while True:
    status = transcribe.get_transcription_job(TranscriptionJobName = job_name)
    if status['TranscriptionJob']['TranscriptionJobStatus'] in ['COMPLETED', 'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
print(status)
```

## Verwenden eines benutzerdefinierten Vokabels in einer Streaming-Transkription

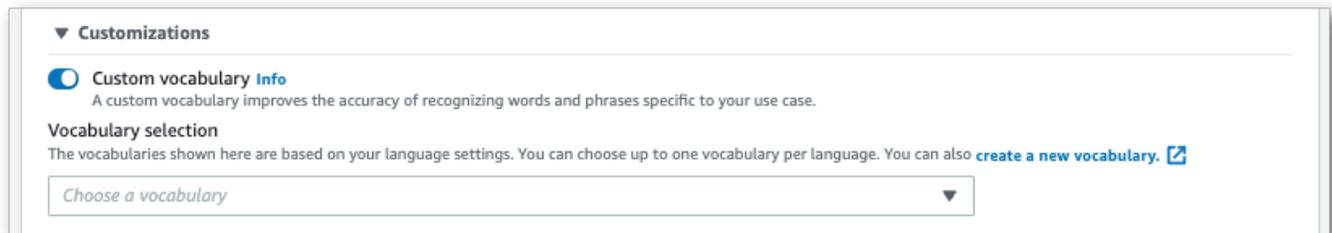
Im Folgenden finden Sie Beispiele, um ein benutzerdefiniertes Vokabular mit einer Streaming-Transkription zu verwenden:

## AWS Management Console

1. Melden Sie sich beim [AWS Management Console](#) an.
2. Wählen Sie im Navigationsbereich Echtzeit-Streaming aus. Scrollen Sie nach unten zu Anpassungen und erweitern Sie dieses Feld, falls es minimiert ist.



3. Schalten Sie auf Benutzerdefiniertes Vokabular und wählen Sie ein benutzerdefiniertes Vokabular aus dem Dropdown-Menü aus.



Fügen Sie alle anderen Einstellungen hinzu, die Sie auf Ihrem Stream anwenden möchten.

4. Sie sind jetzt bereit, Ihren Stream zu transkription. Wählen Sie Streaming starten und beginnen Sie zu sprechen. Um Ihr Diktat zu beenden, wählen Sie Streaming beenden.

## HTTP/2-Stream

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 [Einrichten eines HTTP/2-Streams](#). Weitere Informationen zu Parametern und Headern, die spezifisch für sind Amazon Transcribe, 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 Amazon Transcribe finden Sie unter [Einen WebSocket Stream einrichten](#). Weitere Informationen zu 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 domänenspezifische Sprache zu verbessern. Dies schließt alle Inhalte ein, die nicht dem entsprechen, was Sie in normalen, alltäglichen Gesprächen hören würden. Wenn Sie beispielsweise den Tagungsband einer wissenschaftlichen Konferenz transkribieren, ist es unwahrscheinlich, dass eine Standardtranskription viele der von den Vortragenden verwendeten wissenschaftlichen Begriffe wiedererkennt. In diesem Fall können Sie ein benutzerdefiniertes Sprachmodell trainieren, um die in Ihrer Disziplin verwendeten Fachbegriffe zu erkennen.

Im Gegensatz zu benutzerdefinierten Vokabeln, die die Wiedererkennung eines Wortes verbessern, indem sie Hinweise geben (z. B. zur Aussprache), lernen benutzerdefinierte Sprachmodelle den Kontext, der mit einem bestimmten Wort verknüpft ist. Dazu gehört, wie und wann ein Wort verwendet wird und welche Beziehung ein Wort zu anderen Wörtern hat. Wenn Sie Ihr Modell beispielsweise anhand von klimawissenschaftlichen Forschungsarbeiten trainieren, lernt Ihr Modell möglicherweise, dass „Eisscholle“ ein wahrscheinlicheres Wortpaar ist als „Eisfluss“.

Die unterstützten Sprachen für benutzerdefinierte Sprachmodelle finden Sie unter [Unterstützte Sprachen und sprachspezifische Funktionen](#). Beachten Sie, dass Sie die Sprachenidentifikation nicht aktivieren können, wenn Sie Ihrer Anfrage ein benutzerdefiniertes Sprachmodell hinzufügen (Sie müssen einen Sprachcode angeben).

 API-Operationen speziell für benutzerdefinierte Sprachmodelle

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

## Datenquellen

Sie können jede Art von Textdaten verwenden, um Ihr Modell zu trainieren. Je näher Ihr Textinhalt jedoch an Ihren Audioinhalten ist, desto genauer ist Ihr Modell. Daher ist es wichtig, Textdaten auszuwählen, die dieselben 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. Domaininterne Textdaten haben genau dieselben Begriffe, dieselbe Verwendung und denselben Kontext wie das Audio, das Sie transkribieren möchten.

Wenn Sie nicht über genaue Transkripte verfügen, verwenden Sie Zeitschriftenartikel, technische Berichte, Whitepapers, Konferenzberichte, Benutzerhandbücher, Nachrichtenartikel, Website-Inhalte und jeden anderen Text, der die gewünschten Begriffe enthält, die in einem ähnlichen Kontext wie in Ihrem Audio verwendet werden. Dies wird als domänenbezogene Daten betrachtet.

Um ein robustes benutzerdefiniertes Sprachmodell zu erstellen, ist möglicherweise eine erhebliche Menge an Textdaten erforderlich, 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 werden als Trainingsdaten bezeichnet. Wenn Sie keine (oder nur wenige) domäneninterne Transkripte haben, können Sie optional bis zu 200 MB an Textdaten Amazon Transcribe bereitstellen, um Ihr Modell zu optimieren — dies wird als Tuning-Daten bezeichnet.

## Daten trainieren oder optimieren

Der Zweck von Trainingsdaten besteht darin, Amazon Transcribe zu lehren, neue Begriffe zu erkennen und den Kontext zu erlernen, in dem diese Begriffe verwendet werden. Um ein robustes Modell zu erstellen, ist Amazon Transcribe möglicherweise eine große Menge relevanter Textdaten erforderlich. Es wird dringend empfohlen, so viele Trainingsdaten wie möglich bis zum Limit von 2 GB bereitzustellen.

Der Zweck der Datenoptimierung besteht darin, die aus Ihren Trainingsdaten gewonnenen kontextuellen Beziehungen zu verfeinern und zu optimieren. Zum Erstellen eines benutzerdefinierten Sprachmodells.

Es liegt an Ihnen, zu entscheiden, wie Sie die Trainings- und optional die Tuning-Daten am besten auswählen. Jeder Fall ist einzigartig und hängt von der Art und Menge der Daten ab, die Sie haben. Optimierungsdaten werden empfohlen, wenn Ihnen domäneninterne Trainingsdaten fehlen.

Wenn du dich dafür entscheidest, beide Datentypen einzubeziehen, überschneide deine Trainings- und Tuning-Daten nicht. Trainings- und Tuning-Daten sollten eindeutig sein. Überlappende Daten

können Ihr benutzerdefiniertes Sprachmodell verzerren und verzerren, was sich auf dessen Genauigkeit auswirkt.

Als allgemeine Richtlinie empfehlen wir, wann immer möglich genaue, domäneninterne Texte als Trainingsdaten zu verwenden. Hier sind einige allgemeine Szenarien, die in der Reihenfolge ihrer Präferenz aufgeführt sind:

- Wenn Sie mehr als 10.000 Wörter mit genauem, domäneninternem Transkripttext haben, verwenden Sie ihn als Trainingsdaten. In diesem Fall müssen keine Tuning-Daten hinzugefügt werden. Dies ist das ideale Szenario zum Training eines benutzerdefinierten Sprachmodells.
- Wenn Sie über einen genauen, domäneninternen Transkripttext mit weniger als 10.000 Wörtern verfügen und nicht die gewünschten Ergebnisse erzielen, sollten Sie erwägen, Ihre Trainingsdaten durch domänenbezogene schriftliche Texte wie technische Berichte zu ergänzen. Reservieren Sie in diesem Fall einen kleinen Teil (10-25%) Ihrer Domain-Transkriptdaten, um ihn als Tuning-Daten zu verwenden.
- Wenn Sie keinen domäneninternen Transkripttext haben, laden Sie Ihren gesamten domänenbezogenen Text als Trainingsdaten hoch. In diesem Fall ist Text im Transkriptstil geschriebenem Text vorzuziehen. Dies ist das am wenigsten effektive Szenario für das Training eines benutzerdefinierten Sprachmodells.

Wenn Sie bereit sind, Ihr Modell anzulegen, finden Sie weitere Informationen [Ein benutzerdefiniertes Sprachmodell erstellen](#).

## Ein benutzerdefiniertes Sprachmodell erstellen

Bevor Sie Ihr benutzerdefiniertes Sprachmodell erstellen können, müssen Sie:

- Vorbereiten Ihrer Daten Daten müssen im Klartextformat gespeichert werden und dürfen keine Sonderzeichen enthalten.
- Laden Sie Ihre Daten in einen Amazon S3 Bucket hoch. Es wird empfohlen, separate Ordner für Trainings- und Tuning-Daten zu erstellen.
- Stellen Sie sicher, Amazon Transcribe dass Sie Zugriff auf Ihren Amazon S3 Bucket haben. Sie müssen eine IAM Rolle angeben, die über Zugriffsberechtigungen verfügt, um Ihre Daten verwenden zu können.

## Vorbereiten Ihrer Daten

Sie können alle Ihre Daten in einer Datei zusammenfassen oder als mehrere Dateien speichern. Beachten Sie, dass, wenn Sie Tuning-Daten einbeziehen, diese in einer separaten Datei von Ihren Trainingsdaten gespeichert werden müssen.

Es spielt keine Rolle, wie viele Textdateien Sie für Ihre Trainings- oder Tuning-Daten verwenden. Das Hochladen einer Datei mit 100.000 Wörtern führt zum gleichen Ergebnis wie das Hochladen von 10 Dateien mit 10.000 Wörtern. Bereiten Sie Ihre Textdaten so vor, wie es für Sie am bequemsten ist.

Vergewissern Sie sich, dass alle Ihre Datendateien die folgenden Kriterien erfüllen:

- Sie sind alle in derselben Sprache wie das Modell, das Sie erstellen möchten. Wenn Sie beispielsweise ein benutzerdefiniertes Sprachmodell erstellen möchten, das Audio in US-Englisch (en-US) transkribiert, müssen alle Ihre Textdaten in US-Englisch vorliegen.
- Sie sind im Klartextformat mit UTF-8-Kodierung.
- Sie enthalten keine Sonderzeichen oder Formatierungen wie HTML-Tags.
- Sie belaufen sich auf eine Gesamtgröße von maximal 2 GB für Trainingsdaten und 200 MB für Tuning-Daten.

Wird eines dieser Kriterien nicht erfüllt, schlägt Ihr Modell fehl.

## Hochladen Ihrer Daten

Bevor du deine Daten hochlädst, erstelle einen neuen Ordner für deine Trainingsdaten. Wenn Sie Tuning-Daten verwenden, erstellen Sie einen weiteren separaten Ordner.

Die URIs für Ihre Buckets könnten wie folgt aussehen:

- `s3://DOC-EXAMPLE-BUCKET/my-model-training-data/`
- `s3://DOC-EXAMPLE-BUCKET/my-model-tuning-data/`

Lade deine Trainings- und Tuning-Daten in die entsprechenden Buckets hoch.

Sie können diesen Buckets zu einem späteren Zeitpunkt weitere Daten hinzufügen. Wenn Sie dies jedoch tun, müssen Sie Ihr Modell mit den neuen Daten neu erstellen. Bestehende Modelle können nicht mit neuen Daten aktualisiert werden.

## Zugriff auf Ihre Daten gewähren

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 platziert haben, müssen Sie eine erstellen. Nachdem Sie eine Rolle erstellt haben, können Sie eine Richtlinie anfügen, um die Rolle an die Rolle anfügen. Anfügen der Richtlinie an einen Benutzer anfügen.

Beispiele für Richtlinien finden Sie unter [Amazon Transcribe Beispiele für identitätsbasierte - Richtlinien](#).

Informationen zum Erstellen einer neuen IAM Identität finden Sie unter [IAM Identitäten \(Benutzer, Benutzergruppen und Rollen\)](#).

Weitere Informationen zu Richtlinien finden Sie unter:

- [Richtlinien und Berechtigungen in IAM](#)
- [IAM Richtlinien erstellen](#)
- [Zugriffsverwaltung für AWS-Ressourcen](#)

## Erstellen Sie Ihr benutzerdefiniertes Sprachmodell

Wenn Sie Ihr benutzerdefiniertes Sprachmodell erstellen, müssen Sie ein Basismodell auswählen. Es gibt zwei Basismodelloptionen:

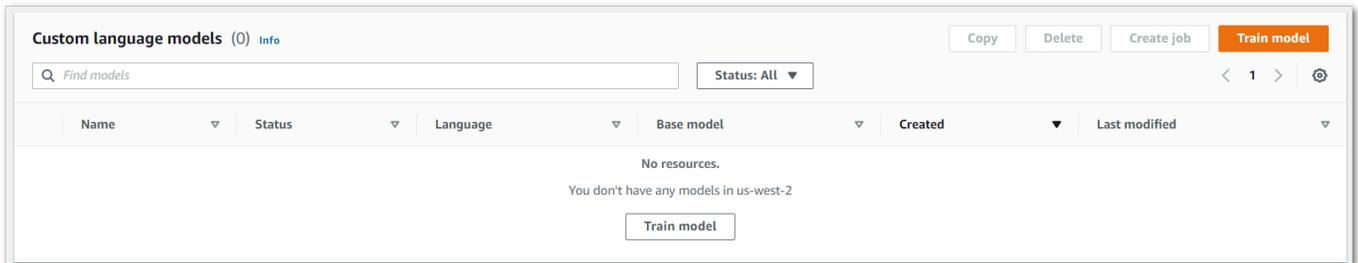
- **NarrowBand:** Verwenden Sie diese Option für Audio mit einer Samplerate von weniger als 16.000 Hz. Dieser Modelltyp wird in der Regel für Telefongespräche verwendet, die mit 8.000 Hz aufgezeichnet wurden.
- **WideBand:** Verwenden Sie diese Option für Audio mit einer Samplerate von mindestens 16.000 Hz.

Sie können benutzerdefinierte Sprachmodelle mithilfe der AWS SDKs, AWS Management Console, AWS CLI, oder erstellen. Sehen Sie sich die folgenden Beispiele an:

### AWS Management Console

1. Melden Sie sich an der [AWS Management Console](#) an.

- Wählen Sie im Navigationsbereich und dann aus. Benutzerdefiniertes Sprachmodell aus. Dadurch wird die Seite Benutzerdefinierte Sprachmodelle geöffnet, auf der Sie vorhandene benutzerdefinierte Sprachmodelle einsehen oder ein neues benutzerdefiniertes Sprachmodell trainieren können.
- Um ein neues Modell zu trainieren, wählen Sie Modell trainieren aus.



Dadurch gelangen Sie zur Eisenbahnmodellseite. Fügen Sie einen Namen hinzu, geben Sie die Sprache an und wählen Sie das gewünschte Basismodell für Ihr Modell aus. Füge dann den Pfad zu deinem Training und optional zu deinen 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.

 ▼

**Base model** [Info](#)

Choose the base model that you want to use to create your custom language model. Choose the model based on the sample rate of your source audio.

**Narrow band**

For audio that has a sample rate less than 16 KHz. Typically, this is 8 KHz audio from telephone conversations.

**Wide band**

For audio that has a sample rate of 16 KHz or greater. Typically, this is 16 KHz audio from media sources.

### Training data [Info](#)

**Training data location on S3**

Type or paste the S3 prefix for the text files that you want to use as training data, or browse to find the files that have matching S3 prefixes.

The file format must be plain text in the language that you have selected for the model. The maximum file size is 2 GB.

### Tuning data - optional [Info](#)

**Tuning data location on S3**

Type or paste the S3 prefix for the text files that you want to use as tuning data, or browse to find the files that have matching S3 prefixes.

The file format must be plain text in the language that you have selected for the model. The maximum file size is 200 MB.

### Access permissions

**IAM role** [Info](#)

Use an existing IAM role

Create an IAM role

By choosing **Train model** you are authorizing creation of this role.

**Role name**

A role that grants access to the S3 input locations.

 ▼

4. Wenn Sie alle Felder ausgefüllt haben, wählen Sie unten auf der Seite Eisenbahnmodell aus.

## AWS CLI

In diesem Beispiel wird der [create-language-model](#) Befehl 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://DOC-EXAMPLE-BUCKET/my-clm-training-  
data/, TuningDataS3Uri=s3://DOC-EXAMPLE-BUCKET/my-clm-tuning-  
data/, DataAccessRoleArn=arn:aws:iam::111122223333:role/ExampleRole \  
--language-code en-US
```

Hier ist ein weiteres Beispiel, in dem der [create-language-model](#) Befehl verwendet wird, und ein 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 Anfragetext.

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

## AWS SDK for Python (Boto3)

In diesem Beispiel wird der `create_language_model` Methode verwendet AWS SDK for Python (Boto3), um ein CLM mit der Methode [create\\_language\\_model](#) zu erstellen. Weitere Informationen erhalten Sie unter [CreateLanguageModel](#) und [LanguageModel](#).

Weitere Beispiele für die Verwendung der AWS SDKs, einschließlich funktionspezifischer, szenariospezifischer und dienstübergreifender Beispiele, finden Sie in diesem Kapitel. [Codebeispiele für Amazon Transcribe mit SDKs AWS](#)

```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
model_name = 'my-first-language-model',
transcribe.create_language_model(
    LanguageCode = 'en-US',
    BaseModelName = 'NarrowBand',
    ModelName = model_name,
    InputDataConfig = {
        'S3Uri': 's3://DOC-EXAMPLE-BUCKET/my-clm-training-data/',
        'TuningDataS3Uri': 's3://DOC-EXAMPLE-BUCKET/my-clm-tuning-data/',
        'DataAccessRoleArn': 'arn:aws:iam::111122223333:role/ExampleRole'
    }
)

while True:
    status = transcribe.get_language_model(ModelName = model_name)
    if status['LanguageModel']['ModelStatus'] in ['COMPLETED', 'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
print(status)
```

## Aktualisierung Ihres benutzerdefinierten Sprachmodells

Amazon Transcribe aktualisiert kontinuierlich die für benutzerdefinierte Sprachmodelle verfügbaren Basismodelle. Um von diesen Updates zu profitieren, empfehlen wir, alle 6 bis 12 Monate neue benutzerdefinierte Sprachmodelle zu schulen.

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` `false` ist, 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 aufnehmen. Beispiele finden Sie in den folgenden Abschnitten.

Die Sprache des Modells, das Sie in Ihrer Anfrage angeben, muss mit dem Sprachcode übereinstimmen, den Sie für Ihr Medium angeben. Wenn die Sprachen nicht übereinstimmen, wird Ihr benutzerdefiniertes Sprachmodell nicht auf Ihre Transkription angewendet und es gibt keine Warnungen oder Fehler.

### Verwendung eines benutzerdefinierten Sprachmodells in einer Batch-Transkription

Um ein benutzerdefiniertes Sprachmodell mit einer Batch-Transkription zu verwenden, finden Sie im Folgenden Beispiele:

#### AWS Management Console

1. Melden Sie sich an der [AWS Management Console](#) an.
2. Wählen Sie im Navigationsbereich Transkriptionsaufträge und dann Job erstellen (oben rechts) aus. Dadurch wird die Seite „Jobdetails angeben“ geöffnet.
3. Wählen Sie im Bereich Auftragseinstellungen unter Modelltyp das Feld Benutzerdefiniertes Sprachmodell aus.

## 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)

- Wählen Sie unter Benutzerdefiniertes Modell ein vorhandenes benutzerdefiniertes Sprachmodell aus dem Dropdown-Menü aus oder erstellen Sie ein neues.

Fügen Sie den Amazon S3 Speicherort Ihrer Eingabedatei im Eingabebereich hinzu.

- Wählen Sie Weiter, um weitere Konfigurationsoptionen zu erhalten.

Wählen Sie Job erstellen aus, um Ihren Transkriptionsjob auszuführen.

## AWS CLI

In diesem Beispiel werden der [start-transcription-job](#) Befehl und der `ModelSettings` Parameter mit dem `VocabularyName` Unterparameter 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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac \  
--output-bucket-name DOC-EXAMPLE-BUCKET \  
--output-key my-output-files/ \  
--language-code en-US \  
--model-settings LanguageModelName=my-first-language-model
```

Hier ist ein weiteres Beispiel für die Verwendung des [start-transcription-job](#) Befehls und ein Anforderungstext, der Ihr benutzerdefiniertes Sprachmodell mit diesem Job 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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"  
  },  
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",  
  "OutputKey": "my-output-files/",  
  "LanguageCode": "en-US",  
  "ModelSettings": {  
    "LanguageModelName": "my-first-language-model"  
  }  
}
```

## AWS SDK for Python (Boto3)

In diesem Beispiel wird das verwendete AWS SDK for Python (Boto3), um mithilfe des `ModelSettings` Arguments für die Methode [start\\_transcription\\_job](#) ein benutzerdefiniertes Sprachmodell einzubinden. Weitere Informationen erhalten Sie unter [StartTranscriptionJob](#) und [ModelSettings](#).

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

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

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

## Verwenden eines benutzerdefinierten Sprachmodells in einer Streaming-Transkription

Um ein benutzerdefiniertes Sprachmodell mit einer Streaming-Transkription zu verwenden, finden Sie im Folgenden Beispiele:

### AWS Management Console

1. Melden Sie sich beim [AWS Management Console](#) an.
2. Wählen Sie im Navigationsbereich Echtzeit-Streaming aus. Scrollen Sie nach unten zu Anpassungen und erweitern Sie dieses Feld, falls es minimiert ist.

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

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

### Transcription

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

Transcription output Current language: English, US

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

00:00 of 15:00 min audio stream

- ▶ Language settings
- ▶ Audio settings
- ▶ Content removal settings
- ▶ Customizations

- Schalten Sie auf Benutzerdefiniertes Sprachmodell um und wählen Sie ein Modell aus dem Dropdown-Menü aus.

### ▼ Customizations

- Custom vocabulary** [Info](#)  
A custom vocabulary improves the accuracy of recognizing words and phrases specific to your use case.
- Partial results stabilization** [Info](#)  
Configure Amazon Transcribe to present results that don't change as it processes the transcription output from your stream.
- Custom language model** [Info](#)  
Select the model you want to use with this streaming session.

**Custom model selection** [Info](#)  
The models shown here are based on your language settings. You can choose up to one model per language. You can also [create a new model](#).

Choose model ▼

Fügen Sie alle anderen Einstellungen hinzu, die auf Ihrem Stream angewendet werden sollen.

4. Sie sind jetzt bereit, Ihrem Stream zu transkribieren. Wählen Sie Streaming starten und beginnen Sie zu sprechen. Um Ihr Diktat zu beenden, wählen Sie Streaming beenden.

## HTTP/2-Stream

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 [Einrichten eines HTTP/2-Streams](#). Weitere Informationen zu Parametern und Headern, die spezifisch für Amazon Transcribe sind, 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-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 Amazon Transcribe finden Sie unter [Einen WebSocket Stream einrichten](#). Weitere Informationen zu 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.

# Verwenden von benutzerdefinierten Vokabelfiltern zum Löschen, Maskieren oder Markieren von Wörtern

Ein benutzerdefinierter Vokabelfilter ist eine Textdatei, die eine benutzerdefinierte Liste einzelner Wörter enthält, die Sie in Ihrer Transkriptionsausgabe ändern möchten.

Ein häufiger Anwendungsfall ist das Entfernen anstößiger oder profaner Begriffe. Benutzerdefinierte Vokabelfilter sind jedoch vollständig benutzerdefiniert, sodass Sie beliebige Wörter auswählen können. Wenn Sie beispielsweise ein neues Produkt haben, das kurz vor der Markteinführung steht, können Sie den Produktnamen in Besprechungsprotokollen maskieren. In diesem Fall halten Sie die Beteiligten geheim up-to-date und halten den Produktnamen bis zur Markteinführung geheim.

Die Vokabelfilterung hat drei Anzeigemethoden: `maskremove`, `undtag`. Sehen Sie sich die folgenden Beispiele an, um zu sehen, wie die einzelnen funktionieren.

- **Maske:** Ersetzt angegebene 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 ihrer Stelle zurück.

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

- **Tag:** Fügt jedem angegebenen Wort ein Tag (`"vocabularyFilterMatch": true`) hinzu, ändert das Wort selbst jedoch nicht. Das Tagging ermöglicht das schnelle Ersetzen und Bearbeiten von Transkripten.

```
"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 exakte Wortübereinstimmungen, wenn sie in Ihrem Transkript erscheinen, entsprechend der von Ihnen angegebenen Filtrationsmethode.

Benutzerdefinierte Vokabelfilter können auf Batch- und Streaming-Transkriptionsanfragen angewendet werden. Informationen zum Erstellen eines benutzerdefinierten Worts finden Sie unter [Einen Vokabelfilter erstellen](#). Informationen zum Anwenden Ihres benutzerdefinierten Vokabelfilters finden Sie unter [Einen benutzerdefinierten Vokabelfilter verwenden](#).

#### Note

Amazon Transcribe maskiert automatisch rassistisch sensible Begriffe. Sie können diesen Standardfilter jedoch deaktivieren, indem Sie sich an den [AWS Technischen Support](#) wenden.

Eine Videoanleitung zur Wortschatzfilterung finden Sie unter [Verwenden von Vokabelfiltern](#).

#### API-Operationen, die für die Vokabelfilterung spezifisch sind

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

## Einen Vokabelfilter erstellen

Einen benutzerdefinierten Vokabel-Filter lässt sich auf zwei Arten lösen:

1. Speichern Sie eine Liste von zeilengetrennten Wörtern als einfache Textdatei mit UTF-8-Codierung.
  - Sie können diesen Ansatz mit den [AWS Management Console](#), [AWS CLI](#), oder [AWS SDKs](#) verwenden.
  - Wenn Sie den [verwenden AWS Management Console](#), können Sie einen lokalen Pfad oder eine [Amazon S3 URI](#) für Ihre benutzerdefinierte Vokabeldatei angeben.

- Wenn Sie die AWS CLI oder AWS SDKs verwenden, müssen Sie Ihre benutzerdefinierte Vokabeldatei in einen Amazon S3 Bucket hochladen und die Amazon S3 URI in Ihre Anfrage aufnehmen.
2. Fügen Sie eine Liste kommasetrennter Wörter direkt in Ihre API-Anfrage ein.
- Sie können diesen Ansatz mit den AWS CLI oder AWS SDKs verwenden, die den [Words](#) Parameter verwenden.

Beispiele für jede Methode finden Sie unter [Benutzerdefinierte Vokabelfilter erstellen](#)

Dinge, die Sie bei der Erstellung Ihres benutzerdefinierten Vokabelfilters beachten sollten:

- Bei Wörtern muss die Groß- und Kleinschreibung ignoriert werden. Zum Beispiel werden „Fluch“ und „FLUCH“ gleich behandelt.
- Nur exakte Wortübereinstimmungen werden gefiltert. Wenn Ihr Filter beispielsweise „schwören“ enthält, Ihre Medien jedoch die Wörter „schwören“ oder „fluchen“ enthalten, werden diese nicht gefiltert. Nur Fälle von „Fluch“ werden gefiltert. Sie müssen daher alle Varianten der Wörter einbeziehen, die Sie filtern möchten.
- Filter gelten nicht für Wörter, die in anderen Wörtern enthalten sind. Wenn ein benutzerdefinierter Vokabelfilter beispielsweise „Marine“, aber nicht „U-Boot“ enthält, wird „U-Boot“ im Transkript nicht geändert.
- Jeder Eintrag kann nur ein Wort (keine Leerzeichen) enthalten.
- Wenn Sie Ihren benutzerdefinierten Vokabelfilter als Textdatei speichern, muss er im Klartextformat mit UTF-8-Codierung vorliegen.
- Sie können bis zu 100 benutzerdefinierte Vokabelfilter pro Filter verwenden, AWS-Konto und jeder kann bis zu 50 KB groß sein.
- Sie können nur Zeichen verwenden, die für Ihre Sprache unterstützt werden. Einzelheiten finden Sie im [Zeichensatz](#) Ihrer Sprache.

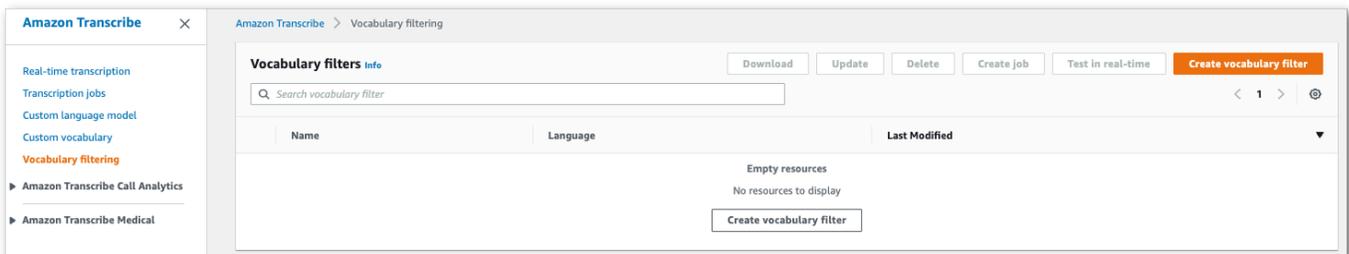
## Benutzerdefinierte Vokabelfilter erstellen

In den folgenden Beispielen erfahren Sie Amazon Transcribe, wie Sie einen benutzerdefinierten Vokabelfilter für die Verwendung mit verarbeiten:

## AWS Management Console

Bevor Sie fortfahren, speichern Sie Ihren benutzerdefinierten Vokabelfilter als Textdatei (\*.txt). Sie können Ihre Datei optional in einen Amazon S3 Bucket hochladen.

1. Melden Sie sich an der [AWS Management Console](#) an.
2. Wählen Sie im Navigationsbereich die Option Vokabularfilter aus. Dadurch wird die Seite Vokabelfilter geöffnet, auf der Sie vorhandene benutzerdefinierte Vokabelfilter anzeigen oder einen neuen erstellen können.
3. Wählen Sie Vokabelfilter erstellen aus.



Dadurch gelangen Sie zur Seite „Vokabelfilter erstellen“. Geben Sie einen Namen für Ihren Vokabular ein.

Wählen Sie unter Vokabeleingabequelle die Option Datei-Upload oder S3-Speicherort. Geben Sie dann den Speicherort Ihrer benutzerdefinierten Vokabeldatei 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**

**Vocabulary input source Info**

File upload

S3 location

**Vocabulary filter file location on S3**

Provide a path to the S3 location where your vocabulary filter file is stored. To find a path, go to [Amazon S3](#)

File format: txt, maximum size 50 KB.

### Tags - optional

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

No tags associated with the resource.

You can add up to 50 more tags.

4. Fügen Sie optional Tags zu Ihrem benutzerdefinierten Vokabelfilter hinzu. Wenn Sie alle Felder ausgefüllt haben, wählen Sie unten auf der Seite die Option Vokabelfilter erstellen aus. Wenn bei der Verarbeitung Ihrer Datei keine Fehler aufgetreten sind, kehren Sie zur Seite mit den Vokabelfiltern zurück.

Ihr benutzerdefinierter Vokabular ist jetzt einsatzbereit.

## AWS CLI

In diesem Beispiel wird der [create-vocabulary-filter](#) Befehl verwendet, um eine Wortliste zu einem verwendbaren benutzerdefinierten Vokabelfilter zu verarbeiten. Weitere Informationen finden Sie unter [CreateVocabularyFilter](#).

Option 1: Mithilfe des `words` Parameters können Sie Ihre Wortliste zu Ihrer Anfrage hinzufügen.

```
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 die URI der Datei mithilfe des `vocabulary-filter-file-uri` Parameters 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://DOC-EXAMPLE-BUCKET/my-vocabulary-filters/my-vocabulary-filter.txt
```

Hier ist ein weiteres Beispiel mit dem [create-vocabulary-filter](#) Befehl und einem Anforderungstext, der Ihren benutzerdefinierten Vokabelfilter 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: Mithilfe des `Words` Parameters können Sie Ihre Wortliste zu Ihrer Anfrage hinzufügen.

```
{  
  "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 die URI der Datei mithilfe des `VocabularyFilterFileUri` Parameters in Ihre Anfrage aufnehmen.

```
{  
  "VocabularyFilterName": "my-first-vocabulary-filter",  
  "LanguageCode": "en-US",
```

```
"VocabularyFilterFileUri": "s3://DOC-EXAMPLE-BUCKET/my-vocabulary-filters/my-
vocabulary-filter.txt"
}
```

### Note

Wenn Sie `VocabularyFilterFileUri` in Ihrer Anfrage angeben, können Sie es nicht `Words` verwenden. Sie müssen das eine oder das andere wählen.

## AWS SDK for Python (Boto3)

In diesem Beispiel wird der verwendete AWS SDK for Python (Boto3), um mithilfe der Methode [create\\_vocabulary\\_filter](#) einen benutzerdefinierten Vokabelfilter zu erstellen. Weitere Informationen finden Sie unter [CreateVocabularyFilter](#).

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

Option 1: Mithilfe des `Words` Parameters können Sie Ihre Wortliste zu Ihrer Anfrage hinzufügen.

```
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 die URI der Datei mithilfe des `VocabularyFilterFileUri` Parameters 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://DOC-EXAMPLE-BUCKET/my-vocabulary-filters/my-
vocabulary-filter.txt'
)
```

### Note

Wenn Sie `esVocabularyFilterFileUri` in Ihrer Anfrage angeben, können Sie `es nichtWords` verwenden. Sie müssen 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 Berechtigungen für den Zugriff auf 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 -Richtlinien](#). Amazon Transcribe

## Einen benutzerdefinierten Vokabelfilter verwenden

Sobald Ihr benutzerdefinierter Vokabelfilter erstellt wurde, können Sie ihn in Ihre Transkriptionsanfragen aufnehmen. Beispiele finden Sie in den folgenden Abschnitten.

Die Sprache des benutzerdefinierten Vokabelfilters, den Sie in Ihrer Anfrage angeben, muss mit dem Sprachcode übereinstimmen, den Sie für Ihr Medium angeben. Wenn Sie die Sprachidentifikation verwenden und mehrere Sprachoptionen angeben, können Sie pro angegebener Sprache einen benutzerdefinierten Vokabelfilter hinzufügen. Wenn die Sprachen Ihrer benutzerdefinierten Vokabelfilter nicht mit der in Ihrem Audio identifizierten Sprache übereinstimmen, werden Ihre Filter nicht auf Ihre Transkription angewendet und es gibt keine Warnungen oder Fehler.

## Verwendung eines benutzerdefinierten Vokabelfilters in einer Batch-Transkription

Um einen benutzerdefinierten Vokabelfilter mit einer Batch-Transkription zu verwenden, finden Sie im Folgenden Beispiele:

### AWS Management Console

1. Melden Sie sich an der [AWS Management Console](#) an.
2. Wählen Sie im Navigationsbereich Transkriptionsaufträge und dann Job erstellen (oben rechts) aus. Dadurch wird die Seite „Jobdetails angeben“ geöffnet.

## 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](#)

Nennen Sie Ihren Job und geben Sie Ihre Eingabemedien an. Fügen Sie optional weitere Felder hinzu und wählen Sie dann Weiter.

3. Schalten Sie auf der Seite „Job konfigurieren“ im Bereich zum Entfernen von Inhalten die Option Vokabelfilterung ein.

## 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](#).

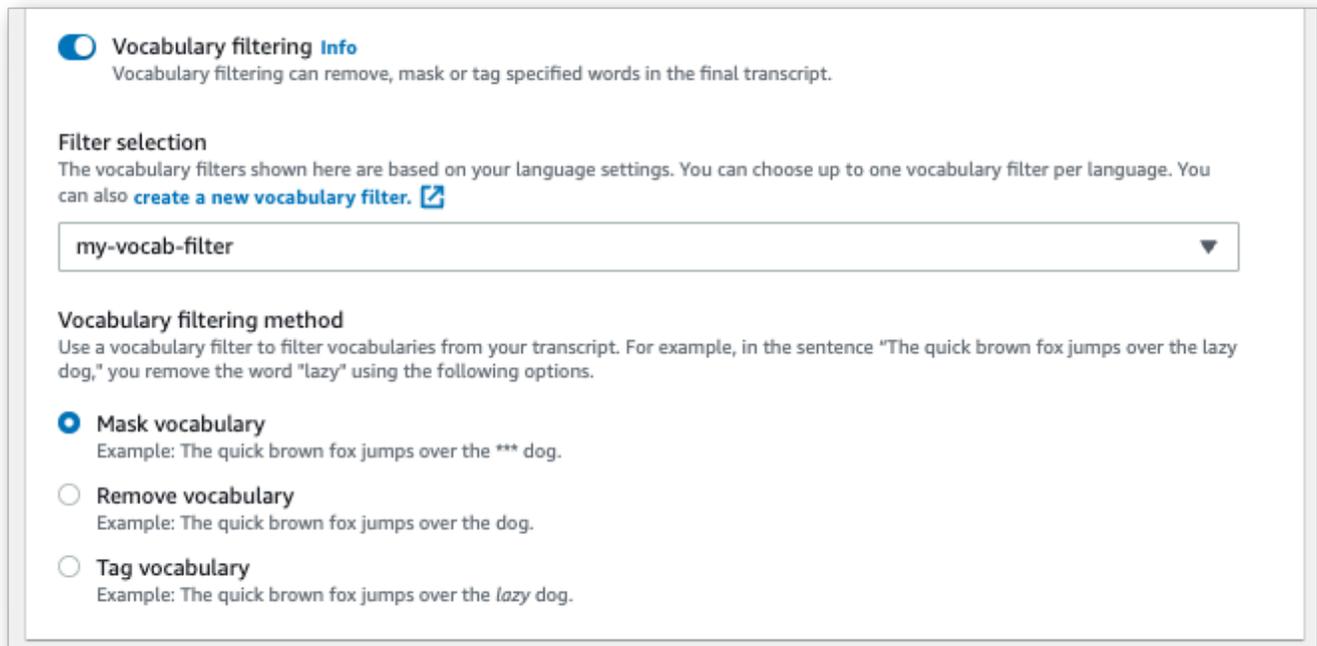
Choose a vocabulary filter ▼

### 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 Vokabelfilter aus dem Dropdown-Menü aus und geben Sie die Filtrationsmethode an.



5. Wählen Sie Job erstellen aus, um Ihren Transkriptionsjob auszuführen.

## AWS CLI

In diesem Beispiel werden der [start-transcription-job](#) Befehl und der `Settings` Parameter mit den `VocabularyFilterMethod` Unterparametern `VocabularyFilterName` und 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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac \
--output-bucket-name DOC-EXAMPLE-BUCKET \
--output-key my-output-files/ \
--language-code en-US \
--settings VocabularyFilterName=my-first-vocabulary-filter,VocabularyFilterMethod=mask
```

Hier ist ein weiteres Beispiel für die Verwendung des [start-transcription-job](#) Befehls und ein Anforderungstext, der Ihren benutzerdefinierten Vokabelfilter für diesen Job 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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
  },
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",
  "OutputKey": "my-output-files/",
  "LanguageCode": "en-US",
  "Settings": {
    "VocabularyFilterName": "my-first-vocabulary-filter",
    "VocabularyFilterMethod": "mask"
  }
}
```

### AWS SDK for Python (Boto3)

In diesem Beispiel wird der verwendet AWS SDK for Python (Boto3), um mithilfe des `Settings` Arguments für die Methode [start\\_transcription\\_job](#) einen benutzerdefinierten Vokabelfilter einzubinden. Weitere Informationen erhalten Sie unter [StartTranscriptionJob](#) und [Settings](#).

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

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

```
        'VocabularyFilterMethod': 'mask'
    }
)

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

## Verwenden eines benutzerdefinierten Vokabelfilters in einer Streaming-Transkription

Um einen benutzerdefinierten Vokabelfilter mit einer Streaming-Transkription zu verwenden, finden Sie im Folgenden Beispiele:

### AWS Management Console

1. Melden Sie sich beim [AWS Management Console](#) an.
2. Wählen Sie im Navigationsbereich Echtzeit-Streaming aus. Scrollen Sie nach unten zu den Einstellungen zum Entfernen von Inhalten und erweitern Sie dieses Feld, wenn es minimiert ist.

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

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

**Transcription**

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

Transcription output Current language: English, US

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

00:00 of 15:00 min audio stream

- ▶ **Language settings**
- ▶ **Audio settings**
- ▶ **Content removal settings**
- ▶ **Customizations**

- Schalten Sie die Vokabelfilterung ein. Wählen Sie einen benutzerdefinierten Vokabelfilter aus dem Dropdown-Menü aus und geben Sie die Filtrationsmethode 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](#).

[+](#)

my-vocab-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 auf Ihren Stream angewendet werden sollen.

4. Sie sind jetzt bereit, Ihren Stream zu transkribieren. Wählen Sie Streaming starten und beginnen Sie zu sprechen. Um Ihr Diktat zu beenden, wählen Sie Streaming beenden.

## HTTP/2-Stream

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 [Einrichten eines HTTP/2-Streams](#). Weitere Informationen zu Parametern und Headern, die spezifisch für Amazon Transcribe sind, 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-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 Amazon Transcribe finden Sie unter [Einen WebSocket Stream einrichten](#). Weitere Informationen zu 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.

# Erkennung toxischer Sprache

Die Erkennung toxischer Sprache soll dabei helfen, Social-Media-Plattformen zu moderieren, die Folgendes beinhalten: peer-to-peer-Dialog, wie Online-Gaming- und Social-Chat-Plattformen. Der Gebrauch toxischer Sprache kann für Einzelpersonen, Gleichaltrige und Gemeinschaften zutiefst schädlich sein. Das Markieren schädlicher Sprache hilft Unternehmen dabei, Gespräche höflich zu führen und eine sichere und integrative Online-Umgebung zu schaffen, in der Benutzer frei erstellen, teilen und teilnehmen 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 grafische Darstellung. Zusätzlich zum Text: Amazon Transcribe Die Toxizitätserkennung verwendet Sprachhinweise wie Töne und Tonhöhen, um die toxische Absicht in der Sprache zu ermitteln. Dies ist eine Verbesserung gegenüber Standardsystemen zur Inhaltsmoderation, die so konzipiert sind, dass sie sich nur auf bestimmte Begriffe konzentrieren, ohne die Absicht zu berücksichtigen.

Amazon Transcribe kennzeichnet und kategorisiert toxische Sprache, wodurch das Datenvolumen, das manuell verarbeitet werden muss, minimiert wird. Dies ermöglicht es Inhaltsmoderatoren, den Diskurs auf ihren Plattformen schnell und effizient zu verwalten.

Zu den Kategorien toxischer Sprache gehören:

- **Profanität:** Sprache, die Wörter, Phrasen oder Akronyme enthält, die unhöflich, vulgär oder beleidigend sind.
- **Hassrede:** Sprache, die eine Person oder Gruppe aufgrund einer Identität (wie Rasse, ethnische Zugehörigkeit, Geschlecht, Religion, sexuelle Orientierung, Fähigkeit und nationale Herkunft) kritisiert, beleidigt, denunziert oder entmenschlicht.
- **Sexuell:** Sprache, die auf sexuelles Interesse, Aktivität oder Erregung hinweist, wobei direkte oder indirekte Hinweise auf Körperteile, körperliche Merkmale oder Geschlecht verwendet werden.
- **Beleidigungen:** Sprache, die erniedrigende, demütigende, spöttische, beleidigende oder herabsetzende Sprache beinhaltet. Diese Art von Sprache wird auch als Mobbing bezeichnet.
- **Gewalt oder Bedrohung:** Sprache, die Drohungen beinhaltet, die darauf abzielen, einer Person oder Gruppe Schmerzen, Verletzungen oder Feindseligkeit zuzufügen.
- **Grafik:** Sprache, die visuell beschreibende und unangenehm lebendige Bilder verwendet. Diese Art von Sprache ist oft absichtlich ausführlich, um das Unbehagen des Empfängers zu verstärken.

- **Belästigung oder Beleidigung:** Sprache, die das psychische Wohlbefinden des Empfängers beeinflussen soll, einschließlich erniedrigender und objektivierender Ausdrücke. Diese Art von Sprache wird auch als Belästigung bezeichnet.

Die Toxizitätserkennung analysiert Sprachsegmente (die Sprache zwischen natürlichen Pausen) und weist diesen Segmenten Konfidenzwerte zu. Konfidenzwerte sind Werte zwischen 0 und 1. Ein höherer Konfidenzwert weist auf eine höhere Wahrscheinlichkeit hin, dass es sich bei dem Inhalt um toxische Sprache in der zugehörigen Kategorie handelt. Sie können diese Konfidenzwerte verwenden, um den für Ihren Anwendungsfall geeigneten Schwellenwert für den Nachweis von Toxizität festzulegen.

#### Note

Der Nachweis von Toxizität ist nur für Batch-Transkriptionen in US-Englisch verfügbar (en-US).

Ansehen [Beispielausgabe](#) im JSON-Format.

## Verwendung der Erkennung toxischer Sprache

### Verwendung der Erkennung toxischer Sprache in einer Batch-Transkription

Im Folgenden finden Sie Beispiele zur Verwendung der toxischen Spracherkennung mit einer Batch-Transkription:

#### AWS Management Console

1. Melden Sie sich an der [AWS Management Console](#) an.
2. Wählen Sie im Navigationsbereich **Transkriptionsjobs**, wählen Sie dann **Job erstellen** (oben rechts). Dies öffnet die **Geben Sie die Jobdetails an** Seite.

## 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.

3. Auf derGeben Sie die Jobdetails anSeite, Sie können auch die Bearbeitung von PII aktivieren, wenn Sie möchten. Beachten Sie, dass die anderen aufgelisteten Optionen bei der Toxizitätserkennung nicht unterstützt werden. Klicken Sie auf Next (Weiter). Das bringt dich zumJob konfigurieren — optionalSeite. In derAudioeinstellungenBedienfeld, wählenNachweis von Toxizität.

### Audio settings

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

---

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

---

**Toxicity detection** [Info](#)  
Flag toxic speech in your transcription output

### Content removal

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

**PII redaction** [Info](#)  
Label the type of PII and also mask the content with the PII entity type in the transcription output. For example, (123) 456-7890 will be masked as [PHONE].

---

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

### Customization

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

Cancel

4. AuswählenJob erstellenum Ihren Transkriptionsjob auszuführen.
5. Sobald Ihr Transkriptionsauftrag abgeschlossen ist, können Sie Ihr Transkript von derHerunterladenDrop-down-Menü auf der Detailseite des Transkriptionsauftrags.

## AWS CLI

In diesem Beispiel wird der [start-transcription-job](#) Befehl und `ToxicityDetection` Parameter. 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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac \  
--output-bucket-name DOC-EXAMPLE-BUCKET \  
--output-key my-output-files/ \  
--language-code en-US \  
--toxicity-detection ToxicityCategories=ALL
```

Hier ist ein weiteres Beispiel mit dem [start-transcription-job](#) Befehl und ein Anforderungstext, der die Toxizitätserkennung beinhaltet.

```
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 Anfragetext.

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

## AWS SDK for Python (Boto3)

In diesem Beispiel wird der AWS SDK for Python (Boto3) um zu aktivieren `ToxicityDetection` für die `Transkriptionsjob starten` Methode. Weitere Informationen erhalten Sie unter [StartTranscriptionJob](#) und [ToxicityDetection](#).

Für weitere Beispiele mit dem AWS SDKs, einschließlich funktionspezifischer, szenariospezifischer und dienstübergreifender Beispiele, finden Sie auf der [Codebeispiele für Amazon Transcribe mit SDKs](#) [AWS](#) Kapitel.

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

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

## Beispielausgabe

Toxische Sprache wird in Ihrer Transkriptionsausgabe markiert und kategorisiert. Jeder Fall von toxischer Sprache wird kategorisiert und ihm wird ein Konfidenzwert zugewiesen (ein Wert zwischen 0 und 1). Ein höherer Konfidenzwert weist auf eine höhere Wahrscheinlichkeit hin, dass es sich bei dem Inhalt um toxische Sprache innerhalb der angegebenen Kategorie handelt.

### Beispielausgabe (JSON)

Im Folgenden finden Sie eine Beispielausgabe im JSON-Format, die kategorisierte toxische Sprache mit zugehörigen Konfidenzwerten zeigt.

```
{
  "jobName": "my-toxicity-job",
  "accountId": "111122223333",
  "results": {
    "transcripts": [...],
    "items": [...],
    "toxicity_detection": [
      {
        "text": "What the * are you doing man? That's why I didn't want to play
with your * . man it was a no, no I'm not calming down * man. I well I spent I spent
too much * money on this game.",
        "toxicity": 0.7638,
        "categories": {
          "profanity": 0.9913,
          "hate_speech": 0.0382,
          "sexual": 0.0016,
          "insult": 0.6572,
          "violence_or_threat": 0.0024,
          "graphic": 0.0013,
          "harassment_or_abuse": 0.0249
        },
        "start_time": 8.92,
        "end_time": 21.45
      },
      Items removed for brevity
    ],
    {
      "text": "What? Who? What the * did you just say to me? What's your
address? What is your * address? I will pull up right now on your * * man. Take your *
back to , tired of this **.",
      "toxicity": 0.9816,
    }
  ]
}
```

```
    "categories": {
      "profanity": 0.9865,
      "hate_speech": 0.9123,
      "sexual": 0.0037,
      "insult": 0.5447,
      "violence_or_threat": 0.5078,
      "graphic": 0.0037,
      "harassment_or_abuse": 0.0613
    },
    "start_time": 43.459,
    "end_time": 54.639
  },
]
},
...
"status": "COMPLETED"
}
```

# Schwärzen oder Identifizieren persönlich identifizierbarer Informationen

Die Redaktion wird verwendet, um vertrauliche Inhalte in Form von personenbezogenen Daten (PII) aus Ihren Transkripten zu maskieren oder zu entfernen. Die Arten von personenbezogenen Daten, die redigiert 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, personenbezogene Daten zu kennzeichnen, ohne sie zu [Beispiel für eine PII-Identifikationsausgabe](#) redigieren. Ein Ausgabebeispiel finden Sie unter [Beispiel für eine PII-Identifikationsausgabe](#).

Wenn die Schwärzung aktiviert ist, haben Sie die Möglichkeit, nur ein redigiertes Transkript oder sowohl ein redigiertes als auch ein unredigiertes Transkript zu generieren. Wenn Sie sich dafür entscheiden, nur ein redigiertes Transkript zu erstellen, beachten Sie, dass Ihre Medien der einzige Ort sind, an dem die gesamte Konversation gespeichert wird. Wenn Sie Ihre Originalmedien löschen, gibt es keine Aufzeichnungen über die unredigierten personenbezogenen Daten. Aus diesem Grund kann es ratsam sein, zusätzlich zu einem redigierten Transkript ein unredigiertes Transkript zu erstellen.

Weitere Informationen zur Bearbeitung personenbezogener Daten mit Batch-Transkriptionen finden Sie unter: [Schwärzen von PII in Ihrem Batch-Job](#).

Weitere Informationen zur Bearbeitung von personenbezogenen Daten oder zur Identifizierung mit Streaming-Transkriptionen finden Sie unter: [Schwärzen oder Identifizieren von PII in einem Echtzeit-Datenstrom](#).

## Important

Die Redaktionsfunktion dient zum Identifizieren und Entfernen vertraulicher Daten. Aufgrund des prädiktiven Charakters des maschinellen Lernens können jedoch möglicherweise nicht alle vertraulichen Daten in Ihrem Transkript identifiziert und entfernt werden. Wir empfehlen Ihnen dringend, alle redigierten Ausgaben zu überprüfen, um sicherzustellen, dass sie Ihren Anforderungen entsprechen.

Die Redaktion von Amazon SNS erfüllt nicht die Anforderungen der US-Health Insurance Portability and Accountability Act of 1996 (HIPAA).

Eine Videoanleitung zur Redaktionsfunktion Amazon Transcribe der App finden Sie unter [Verwenden der Inhaltsredaktion, um personenbezogene Daten zu identifizieren und zu redigieren](#).

## 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. In dem Teil der Transkriptionsausgabe können Sie sich auch die Art der personenbezogenen Daten ansehen, die geschwärzt wurden. word-for-word Ein Beispiel für die Ausgabe finden Sie unter [Beispiel für eine redigierte Ausgabe \(Batch\)](#).

Die Schwärzung mit Batch-Transkriptionen ist in US-Englisch (en-US) und US-Spanisch () verfügbar. es-US 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, die bei Amazon Transcribe Batch-Transkriptionen erkannt werden können

PII-Typ	Beschreibung
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.

PII-Typ	Beschreibung
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.
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 Zahl ist in der Regel 4-stellig und wird als Monat/Jahr oder MM/JJJJ formatiert. Amazon Transcribe Kann beispielsweise Ablaufdaten wie den 21. Januar, den 1. Januar 2021 und den 1. 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 Debitkartennummern, wenn nur die letzten 4 Ziffern vorhanden sind.
EMAIL	Eine E-Mail-Adresse, z. B. efua.owusu@email.com.

PII-Typ	Beschreibung
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.
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.

Sie können einen Batch-Transkriptionsauftrag mit dem AWS SDK AWS Management Console AWS CLI, oder starten.

## AWS Management Console

1. Melden Sie sich an der [AWS Management Console](#) 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)**

<input checked="" type="checkbox"/> BANK_ACCOUNT_NUMBER	<input checked="" type="checkbox"/> BANK_ROUTING	<input checked="" type="checkbox"/> CREDIT_DEBIT_NUMBER
<input checked="" type="checkbox"/> CREDIT_DEBIT_CVV	<input checked="" type="checkbox"/> CREDIT_DEBIT_EXPIRY	<input checked="" type="checkbox"/> PIN

**Personal (5 of 5 selected)**

<input checked="" type="checkbox"/> NAME	<input checked="" type="checkbox"/> ADDRESS	<input checked="" type="checkbox"/> PHONE
<input checked="" type="checkbox"/> EMAIL	<input checked="" type="checkbox"/> SSN	

---

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

5. Wählen Sie Auftrag erstellen, um Ihren Transkriptionsauftrag auszuführen.

## AWS CLI

In diesem Beispiel werden der Befehl und der [start-transcription-job](#) Parameter verwendet. `content-redaction` Weitere Informationen finden Sie unter [StartTranscriptionJob](#) und [ContentRedaction](#).

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

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

Hier ist ein weiteres Beispiel, in dem die [start-transcription-job](#) Methode verwendet wird, und der Anfragetext redigiert die PII für diesen Job.

```
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 Anfragetext.

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

## AWS SDK for Python (Boto3)

In diesem Beispiel wird der verwendet AWS SDK for Python (Boto3) , um den Inhalt mithilfe des ContentRedaction Arguments für die Methode [start\\_transcription\\_job](#) zu redigieren. Weitere Informationen finden 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 SDKs AWS](#)

```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
job_name = "my-first-transcription-job"
job_uri = "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
transcribe.start_transcription_job(
    TranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'DOC-EXAMPLE-BUCKET',
    OutputKey = 'my-output-files/',
    LanguageCode = 'en-US',
    ContentRedaction = {
        'RedactionOutput': 'redacted',
        'RedactionType': 'PII',
        'PiiEntityTypes': [
            'NAME', 'ADDRESS', 'BANK_ACCOUNT_NUMBER'
        ]
    }
)

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

### Note

Die Bearbeitung personenbezogener Daten für Batch-Jobs wird nur in folgenden Ländern unterstützt AWS-Regionen: Asien-Pazifik (Hongkong), Asien-Pazifik (Mumbai), Asien-Pazifik (Seoul), Asien-Pazifik (Singapur), Asien-Pazifik (Sydney), Asien-Pazifik GovCloud (Tokio), (US-West), Kanada (Zentral), EU (Frankfurt), EU (Irland), EU (London), EU (Paris), Naher Osten (Bahrain), Südamerika (Sao Paulo), USA Ost (Nord-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 redigierte Streaming-Ausgabe](#) und [Beispiel für eine PII-Identifikationsausgabe](#).

Die Bearbeitung und Identifizierung personenbezogener Daten mit Streaming-Transkriptionen ist in den folgenden englischen Dialekten verfügbar: Australischer (en-AU), Britischer (), US-amerikanischer () und spanischer US-Dialekt (en-GBen-US). es-US

Die Identifizierung und Schwärzung von personenbezogenen Daten bei Streaming-Aufträgen erfolgt erst nach vollständiger Transkription der Audiosegmente.

Arten von personenbezogenen Daten, die bei Streaming-Transkriptionen erkannt werden können  
Amazon Transcribe

PII-Typ	Beschreibung
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	Beschreibung
	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 Zahl ist in der Regel 4-stellig und wird als Monat/Jahr oder MM/JJJJ formatiert. Amazon Transcribe Kann beispielsweise Ablaufdaten wie den 21. Januar, den 1. Januar 2021 und den 1. 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 Debitkartennummern, 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	Beschreibung
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.

Sie können eine Streaming-Transkription mit AWS Management Console WebSocket, oder HTTP/2 starten.

## AWS Management Console

1. Melden Sie sich beim [AWS Management Console](#) an.
2. Wählen Sie im Navigationsbereich Echtzeit-Streaming aus. Scrollen Sie nach unten zu Einstellungen für das Entfernen von Inhalten und erweitern Sie dieses Feld, falls es minimiert ist.

The screenshot shows the Amazon Transcribe transcription interface. At the top left, the word "Transcription" is displayed. To its right are two buttons: "Download full transcript" and "Start streaming" (which is highlighted in orange). Below this is the "Transcription output" section, which includes the text "Choose Start streaming to begin a real-time transcription of what you speak into your microphone" and a large grey rectangular area for the transcription. To the right of this area, it says "Current language: English, US". Below the transcription area, a timer shows "00:00 of 15:00 min audio stream". At the bottom, there are four expandable settings sections: "Language settings", "Audio settings", "Content removal settings", and "Customizations".

### 3. Umschalten auf PII-Identifizierung und Schwärzung.

The screenshot shows the "Content removal settings" section. It is expanded to show two options, both with radio buttons that are currently unselected. The first option is "Vocabulary filtering" with an "Info" link. Below it, a description reads: "Vocabulary filtering removes, masks, or tags words that you specify in your vocabulary filter. Choose a vocabulary filter to see an example." The second option is "PII Identification & redaction" with an "Info" link. Below it, a description reads: "Identify or redact one or more types of personally identifiable information (PII) in your transcript".

- 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)**

<input checked="" type="checkbox"/> BANK_ACCOUNT_NUMBER	<input checked="" type="checkbox"/> BANK_ROUTING	<input checked="" type="checkbox"/> CREDIT_DEBIT_NUMBER
<input checked="" type="checkbox"/> CREDIT_DEBIT_CVV	<input checked="" type="checkbox"/> CREDIT_DEBIT_EXPIRY	<input checked="" type="checkbox"/> PIN

**Personal (8 of 8 selected)**

<input checked="" type="checkbox"/> NAME	<input checked="" type="checkbox"/> ADDRESS	<input checked="" type="checkbox"/> PHONE
<input checked="" type="checkbox"/> EMAIL	<input checked="" type="checkbox"/> SSN	<input checked="" type="checkbox"/> PASSPORT_NUMBER
<input checked="" type="checkbox"/> DRIVER_ID	<input checked="" type="checkbox"/> AGE	

**Digital footprint (7 of 7 selected)**

<input checked="" type="checkbox"/> URL	<input checked="" type="checkbox"/> USERNAME	<input checked="" type="checkbox"/> PASSWORD
<input checked="" type="checkbox"/> AWS_ACCESS_KEY	<input checked="" type="checkbox"/> AWS_SECRET_KEY	<input checked="" type="checkbox"/> IP_ADDRESS
<input checked="" type="checkbox"/> MAC_ADDRESS		

**Other (1 of 1 selected)**

DATE\_TIME

► Customizations

- 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-Stream

In diesem Beispiel wird eine HTTP/2-Anfrage mit aktivierter PII-Identifizierung oder PII-Redaktion erstellt. Weitere Informationen zur Verwendung von HTTP/2-Streaming mit finden Sie Amazon Transcribe unter [Einrichten eines HTTP/2-Streams](#). 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 PII-Redigierung und Identifikationsausgabe

Die folgenden Beispiele zeigen die redigierte Ausgabe von Batch- und Streaming-Jobs sowie die Identifizierung von personenbezogenen Daten aus einem Streaming-Job.

Transkriptionsaufträge mit Inhaltsredaktion generieren 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 Transkriptausgabe `Good` hat das Wort ein `confidence` von `1.0`. Dieser Konfidenzwert gibt an, dass zu 100 Prozent sicher Amazon Transcribe ist, dass das in diesem Transkript ausgesprochene 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 Transkriptausgabe `0.9999` gibt das `confidence` von `0.9999` an, dass zu 99,99 Prozent sicher Amazon Transcribe ist, dass es sich bei der Entität, die es im Transkript redigiert hat, um personenbezogene Daten handelt.

## Beispiel für eine redigierte 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
my Social
is [PII].
personal
information away. Let's check."
    }
  ],
  "items": [
    {
      "start_time": "2.86",
      "end_time": "3.35",
      "alternatives": [
        {
          "confidence": "1.0",
          "content": "Good"
        }
      ],
      "type": "pronunciation"
    },
    Items removed for brevity
    {
      "start_time": "5.56",
      "end_time": "6.25",
      "alternatives": [
        {
          "content": "[PII]",
          "redactions": [
            {
              "confidence": "0.9999",
              "type": "NAME",
              "category": "PII"
            }
          ]
        }
      ],
      "type": "pronunciation"
    },
    Items removed for brevity
  ],
],
```

```

    },
    "status": "COMPLETED"
  }

```

Hier ist das unredigierte Protokoll 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
job
at redacting that personal information away. Let's check."
      }
    ],
    "items": [
      {
        "start_time": "2.86",
        "end_time": "3.35",
        "alternatives": [
          {
            "confidence": "1.0",
            "content": "Good"
          }
        ],
        "type": "pronunciation"
      },
      Items removed for brevity
      {
        "start_time": "5.56",
        "end_time": "6.25",
        "alternatives": [
          {
            "confidence": "0.9999",
            "content": "Mike",

```

```

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

```

## Beispiel für eine redigierte Streaming-Ausgabe

```

{
  "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 eine PII-Identifikationsausgabe

Die PII-Identifizierung ist eine zusätzliche Funktion, die Sie für Ihren Streaming-Transkriptionsauftrag verwenden können. Die identifizierten PII sind im `Entities` Abschnitt jedes 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 Video-Untertitel

Amazon Transcribe unterstützt die WebVTT-Ausgabe (\*.vtt) und SubRip (\*.srt) zur Verwendung als Videountertitel. Sie können bei der Einrichtung Ihres Batch-Videotranskriptionsauftrags einen oder beide Dateitypen auswählen. Wenn Sie die Untertitelfunktion verwenden, werden Ihre ausgewählten Untertiteldateien und eine reguläre Transkriptdatei (mit zusätzlichen Informationen) erstellt. Untertitel- und Transkriptionsdateien werden an dasselbe Ziel ausgegeben.

Untertitel werden gleichzeitig mit dem Sprechen des Textes angezeigt und bleiben sichtbar, bis eine natürliche Pause eingetreten ist oder der Sprecher das Sprechen beendet. Beachten Sie, dass keine Untertiteldatei erstellt wird, wenn Sie in Ihrer Transkriptionsanfrage Untertitel aktivieren und Ihr Audio keine Sprache enthält.

## Important

Amazon Transcribe verwendet einen Standard-Startindex von 0 für die Ausgabe von Untertiteln, der sich vom gebräuchlicheren Wert von 1 unterscheidet. Wenn Sie einen Startindex von benötigen<sup>1</sup>, können Sie dies in der AWS Management Console oder in Ihrer API-Anfrage mithilfe des [OutputStartIndex](#) Parameters angeben.

Die Verwendung des falschen Startindex kann zu Kompatibilitätsfehlern mit anderen Diensten führen. Überprüfen Sie daher, welchen Startindex Sie benötigen, bevor Sie Ihre Untertitel erstellen. Wenn Sie sich nicht sicher sind, welchen Wert Sie verwenden sollen, empfehlen wir Ihnen, sich zu entscheiden<sup>1</sup>. Weitere Informationen [Subtitles](#) finden Sie unter.

Funktionen, die mit Untertiteln unterstützt werden:

- **Inhaltsschwärzung** — Jeder redigierte Inhalt wird sowohl in Ihren Untertiteln als auch in den regulären Transkript-Ausgabedateien als PII " wiedergegeben. Das Audio wird nicht verändert.
- **Vokabelfilter** — Untertiteldateien werden aus der Transkriptionsdatei generiert, sodass alle Wörter, die Sie in Ihrer Standard-Transkriptionsausgabe filtern, auch in Ihren Untertiteln gefiltert werden. Gefilterter Inhalt wird als Leerzeichen oder \* \* \* in Ihren Transkript- und Untertiteldateien angezeigt. Das Audio wird nicht verändert.
- **Sprechertagebücher** — Wenn sich in einem bestimmten Untertitelsegment mehrere Sprecher befinden, werden Bindestriche verwendet, um die einzelnen Sprecher zu unterscheiden. Dies gilt sowohl für WebVTT als auch für SubRip Formate; zum Beispiel:

- -- Von Person 1 gesprochener Text
- -- Von Person 2 gesprochener Text

Untertiteldateien werden am selben Amazon S3 Ort wie Ihre Transkriptionsausgabe gespeichert.

Eine [Videoanleitung zum Erstellen von Untertiteln finden Sie unter Amazon Transcribe Video Snacks: Erstellen von Videountertiteln, ohne Code zu schreiben.](#)

## Generieren von Untertiteldateien

Sie können Untertiteldateien mit den AWS Management Console, AWS CLI, oder AWSSDKs erstellen. Sehen Sie sich die folgenden Beispiele an:

### AWS Management Console

1. Melden Sie sich an der [AWS Management Console](#) an.
2. Wählen Sie im Navigationsbereich Transkriptionsaufträge und dann Job erstellen (oben rechts). Dadurch wird die Seite „Auftragsdetails angeben“ geöffnet. Die Untertiteloptionen befinden sich im Bedienfeld „Ausgabedaten“.
3. Wählen Sie die gewünschten Formate für Ihre Untertiteldateien aus und wählen Sie dann einen Wert für Ihren Startindex. Beachten Sie, dass die Amazon Transcribe Standardeinstellung 1 ist 0, aber häufiger verwendet wird. Wenn Sie sich nicht sicher sind, welchen Wert Sie verwenden sollen, empfehlen wir die Auswahl 1, da dies die Kompatibilität mit anderen Diensten verbessern kann.

## Output data

Output data location type info [Info](#)

- Service-managed S3 bucket  
The output will be removed after 90 days when the job expires.
- Customer specified S3 bucket  
The output will not be removed from bucket even after the job expires.

Subtitle file format [Info](#)

- SRT (SubRip)
- VTT (WebVTT)

Specify the start index

0 ▼

- Füllen Sie alle anderen Felder aus, die Sie auf der Seite „Auftragsdetails angeben“ hinzufügen möchten, und wählen Sie dann Weiter aus. Dadurch gelangen Sie zur Seite „Job konfigurieren — optional“.
- Wählen Sie Job erstellen aus, um Ihren Transkriptionsjob auszuführen.

## AWS CLI

In diesem Beispiel werden der [start-transcription-job](#) Befehl und der Subtitles Parameter 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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac \
--output-bucket-name DOC-EXAMPLE-BUCKET \
--output-key my-output-files/ \
--language-code en-US \
--subtitles Formats=vtt,srt,OutputStartIndex=1
```

Hier ist ein weiteres Beispiel, in dem der [start-transcription-job](#) Befehl verwendet wird, und ein Anforderungstext, der diesem Job Untertitel 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 Anfragetext.

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

## AWS SDK for Python (Boto3)

In diesem Beispiel wird das verwendete AWS SDK for Python (Boto3), um Untertitel mithilfe des `Subtitles` Arguments für die Methode [start\\_transcription\\_job](#) hinzuzufügen. Weitere Informationen erhalten Sie unter [StartTranscriptionJob](#) und [Subtitles](#).

Weitere Beispiele für die Verwendung der AWS SDKs, einschließlich funktionspezifischer, szenariospezifischer und dienstübergreifender Beispiele, finden Sie in diesem Kapitel. [Codebeispiele für Amazon Transcribe mit SDKs AWS](#)

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

```
OutputKey = 'my-output-files/',
LanguageCode = 'en-US',
Subtitles = {
    'Formats': [
        'vtt', 'srt'
    ],
    'OutputStartIndex': 1
}
)

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

# 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 [Analyse nach Anrufen](#) und [Echtzeit-Call-Analytics](#).

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 [Analyse nach Anrufen mit Echtzeit-Transkriptionen](#).

## API-Vorgänge speziell für Call Analytics

Nach dem Anruf: [CreateCallAnalyticsCategory](#), [DeleteCallAnalyticsCategory](#), [DeleteCallAnalyticsJob](#), [GetCallAnalyticsCategory](#), [GetCallAnalyticsJob](#), [ListCallAnalyticsCategories](#), [ListCallAnalyticsJobs](#), [StartCallAnalyticsJob](#), [UpdateCallAnalyticsCategory](#)  
Echtzeit: [StartCallAnalyticsStreamTranscription](#)  
[StartCallAnalyticsStreamTranscriptionWebSocket](#)

## Häufige Anwendungsfälle

Transkription nach Anrufen:

- 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.

#### Transkriptionen in Echtzeit:

- 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 für 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 [Ausgabe für Analyse nach Anrufen](#) und [Echtzeit-Call-Analytics-Ausgabe](#).

## 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.
- [AuftragswarteschlangenWarteschlange tesch](#) 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. 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. Weitere Informationen finden Sie in der [AWS allgemeinen Referenz](#).

 Tauchen Sie mit dem Blog zum AWS Machine Learning tiefer ein

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-Dokumenten-anwendung](#) an, mit der Sie Ihr Transkript in ein easy-to-read 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.

## Analyse nach Anrufen

Call Analytics bietet Analysen nach Anrufen, die für die Überwachung von Trends im Kundenservice nützlich sind.

Die Transkriptionen der Anrufe bieten folgende Erkenntnisse:

- [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

## Anruferkenntnisse nach dem Gespräch

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.
- **Nicht-Sprachzeit:** Misst Zeiträume, in denen nicht gesprochen wird. 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:

- Nicht-Gesprächszeit: Zeitspannen, in denen weder der Kunde noch der Kundendienstmitarbeiter spricht.
- 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 (back-and-forth 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

Die Bearbeitung personenbezogener Daten nach einem Anruf 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 [Schwärzen oder Identifizieren 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 funktioniert out-of-the-box und unterstützt daher keine Anpassungen wie Modelltraining oder benutzerdefinierte Kategorien.

Hier ist ein [Beispiel für die Ausgabe](#).

## Erstellen von Kategorien für Transkription nach Anrufen

Die Analyse nach Anrufen unterstützt die Erstellung benutzerdefinierter Kategorien, sodass Sie Ihre Transkriptionsanalysen 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 Analysekategorien 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 Anrufkategorisierung](#).

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

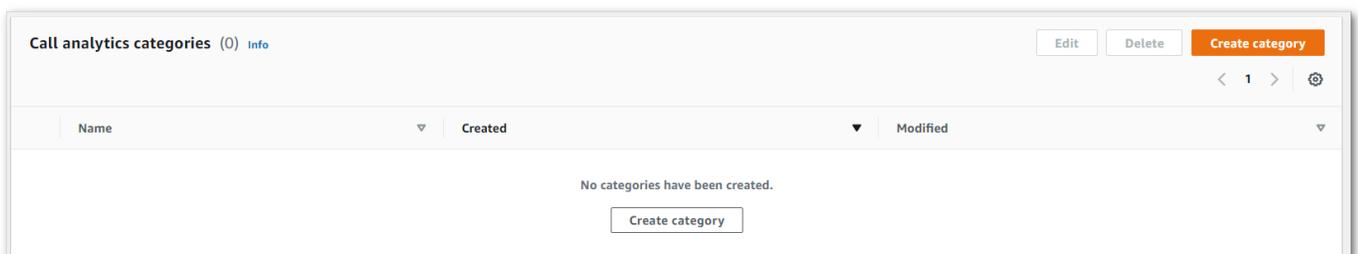
## Nach Anrufen- versus Echtzeit-Kategorien

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 Nach Anrufen-Kategorie erstellt. Entsprechende Kategorien für die Analyse nach Anrufen sind nach Abschluss der Transkription für die Analyse nach Anrufen in Ihrer Ausgabe verfügbar.

Um eine neue Kategorie für die Analyse nach Anrufen zu erstellen, können Sie die SDKs AWS AWS Management Console oder AWS CLI verwenden. Beispiele finden Sie im Folgenden:

### AWS Management Console

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 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 Kategorietyt 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.

The screenshot shows the 'Rules' configuration page in Amazon Transcribe. At the top, it states: 'All the rule conditions must be met for a transcription job to be classified in this category.' Below this, there is a section for 'Rule 1' with a 'Delete rule' button. The rule's condition is: 'When no word has been spoken for more than 5 minute(s) during the entire call.' Underneath, the 'Rule type' is set to 'Non-talk time', with an 'Info' link. A dropdown menu is open, showing four options: 'Non-talk time' (selected), 'Interruption time', 'Transcript content match', and 'Transcript sentiment match'. Each option has a brief description of when the rule is triggered. At the bottom of the dropdown is an 'Add rule' button. A note at the very bottom says: '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** Delete rule

When the duration of the interruption was more than 15 second(s) during the entire call when the speaker was agent.

**Rule type** [Info](#)  
Choose the rule that you want to define.

Interruption time ▼

**Logic** [Info](#)  
Define the conditions that must be met.

When the duration of the interruption was more than   ▼

during the  ▼

when the speaker was  ▼

AND

**▼ Rule 2** Delete rule

When the sentiment is negative during the last 2 minute(s) when the speaker was either.

**Rule type** [Info](#)  
Choose the rule that you want to define.

Transcript sentiment match ▼

**Logic** [Info](#)  
Define the conditions that must be met.

When the sentiment is  ▼

during the  ▼   ▼

when the speaker was  ▼

You can add up to 18 more rules.

8. 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 for Python (Boto3)

In diesem Beispiel wird mithilfe der AWS SDK for 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 SDKs AWS](#)

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](#).

## Nicht-Gesprächszeit-Übereinstimmung

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 Amazon Transcribe Audio entsprechen 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 Management Console oder AWS CLI verwenden. Beispiele finden Sie im Folgenden:

## AWS Management Console

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
<input type="radio"/>	CatchNegativeSentiment	POST_CALL	February 17 2023, 10:43 (UTC-08:00)	February 17 2023, 10:43 (UTC-08:00)

If the above categories aren't relevant to your use case, you can create a new category. [Create a new category.](#)

3. 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**

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.

### Input data [Info](#)

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

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

**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.

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.

MyTranscribeRole

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
<input type="radio"/>	CatchNegativeSentiment	POST_CALL	February 17 2023, 10:43 (UTC-08:00)	February 17 2023, 10:43 (UTC-08:00)

If the above categories aren't relevant to your use case, you can create a new category. [Create a new category.](#) 

## 6. 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 finden 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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac \  
--output-location s3://DOC-EXAMPLE-BUCKET/my-output-files/ \  
--data-access-role-arn arn:aws:iam::111122223333:role/ExampleRole \  
--channel-definitions ChannelId=0,ParticipantRole=AGENT \  
ChannelId=1,ParticipantRole=CUSTOMER
```

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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"  
  },  
  "OutputLocation": "s3://DOC-EXAMPLE-BUCKET/my-output-files/",  
  "ChannelDefinitions": [  
    {  
      "ChannelId": 0,  
      "ParticipantRole": "AGENT"  
    },  
    {  
      "ChannelId": 1,  
      "ParticipantRole": "CUSTOMER"  
    }  
  ]  
}
```

```

    }
  ]
}

```

## AWS SDK for Python (Boto3)

In diesem Beispiel wird der verwendet AWS SDK for Python (Boto3) , um einen Call Analytics-Job mit der Methode [start\\_call\\_analytics\\_job](#) zu starten. Weitere Informationen finden 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 SDKs AWS](#)

```

from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
job_name = "my-first-call-analytics-job"
job_uri = "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
output_location = "s3://DOC-EXAMPLE-BUCKET/my-output-files/"
data_access_role = "arn:aws:iam::111122223333:role/ExampleRole"
transcribe.start_call_analytics_job(
    CallAnalyticsJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    DataAccessRoleArn = data_access_role,
    OutputLocation = output_location,
    ChannelDefinitions = [
        {
            'ChannelId': 0,
            'ParticipantRole': 'AGENT'
        },
        {
            'ChannelId': 1,
            'ParticipantRole': 'CUSTOMER'
        }
    ]
)

while True:
    status = transcribe.get_call_analytics_job(CallAnalyticsJobName = job_name)

```

```
if status['CallAnalyticsJob']['CallAnalyticsJobStatus'] in ['COMPLETED', 'FAILED']:
    break
print("Not ready yet...")
time.sleep(5)
print(status)
```

## Ausgabe für Analyse nach Anrufen

Analysetranskripte nach dem Anruf werden segmentweise angezeigt turn-by-turn . Dazu gehören Anrufrkategorisierung, Anrufmerkmale (Lautstärkewerte, Unterbrechungen, Nicht-Gesprächszeit, Gesprächsgeschwindigkeit), Anruferzusammenfassung (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](#).

### Anrufrkategorisierung

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 Anruferzusammenfassung

So sieht die generative Anruferzusammenfassung 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 Anruferzusammenfassung

### Note

Bereitgestellt von Amazon Bedrock: AWS implementiert [automatisierte Missbrauchserkennung](#). Da die durch generative KI gestützte Zusammenfassung nach

erfolgtem 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 Management Console

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
<input type="radio"/>	CatchNegativeSentiment	POST_CALL	February 17 2023, 10:43 (UTC-08:00)	February 17 2023, 10:43 (UTC-08:00)

Aktivieren der generativen Anruferzusammenfassung

If the above categories aren't relevant to your use case, you can create a new category. [Create a new category.](#) 

## 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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac \  
--output-location s3://DOC-EXAMPLE-BUCKET/my-output-files/ \  
--data-access-role-arn arn:aws:iam::111122223333:role/ExampleRole \  
--channel-definitions ChannelId=0,ParticipantRole=AGENT  
ChannelId=1,ParticipantRole=CUSTOMER  
--settings '{"Summarization":{"GenerateAbstractiveSummary":true}}'
```

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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"  
  },  
  "OutputLocation": "s3://DOC-EXAMPLE-BUCKET/my-output-files/",  
  "ChannelDefinitions": [  
    {  
      "ChannelId": 0,  
      "ParticipantRole": "AGENT"  
    }  
  ]  
}
```

```
    },
    {
      "ChannelId": 1,
      "ParticipantRole": "CUSTOMER"
    }
  ],
  "Settings": {
    "Summarization":{
      "GenerateAbstractiveSummary": true
    }
  }
}
```

## AWS SDK for Python (Boto3)

[In diesem Beispiel wird mithilfe der Methode AWS SDK for 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 SDKs AWS](#)

```
from __future__ import print_function
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
job_name = "my-first-call-analytics-job"
job_uri = "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
output_location = "s3://DOC-EXAMPLE-BUCKET/my-output-files/"
data_access_role = "arn:aws:iam::111122223333:role/ExampleRole"
transcribe.start_call_analytics_job(
    CallAnalyticsJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    DataAccessRoleArn = data_access_role,
    OutputLocation = output_location,
    ChannelDefinitions = [
        {
            'ChannelId': 0,
```

```
        'ParticipantRole': 'AGENT'
    },
    {
        'ChannelId': 1,
        'ParticipantRole': 'CUSTOMER'
    }
],
Settings = {
    "Summarization":
        {
            "GenerateAbstractiveSummary": true
        }
}
)

while True:
    status = transcribe.get_call_analytics_job(CallAnalyticsJobName = job_name)
    if status['CallAnalyticsJob']['CallAnalyticsJobStatus'] in ['COMPLETED', 'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
print(status)
```

## Echtzeit-Call-Analytics

Echtzeit-Call-Analytics bieten Echtzeit-Einblicke, die zur Behebung von Problemen und zur Abschwächung von Eskalationen verwendet werden können.

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

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.

## Einblicke in Echtzeit

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 die Phrase „Ich möchte mit dem Manager sprechen“ festlegen, wird nach genau dieser Wortgruppe 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. Es funktioniert out-of-the-box 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**

Die Identifizierung personenbezogener Daten in Echtzeit wird mit den folgenden englischen Dialekten unterstützt: Australisch (en-AU), Britisch (en-GB), USA (en-US) und mit spanischsprachigem 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 Schwärzen oder Identifizieren 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**

Die Bearbeitung personenbezogener Daten in Echtzeit 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 [Schwärzen oder Identifizieren 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 funktioniert out-of-the-box und unterstützt daher keine Anpassungen wie Modelltraining oder benutzerdefinierte Kategorien.

Hier ist ein [Beispiel für die Ausgabe](#).

## Kategorien für Echtzeit-Transkriptionen erstellen

Echtzeit-Call Analyticsn unterstützen die Erstellung benutzerdefinierter Kategorien, mit denen Sie Ihre Transkriptanalysen 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.

Call-Analytics-Transkriptionen in Echtzeit unterstützen nur Regeln, die [TranscriptFilter](#) (Schlüsselwortübereinstimmung) 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.

### Nach Anrufen- versus Echtzeit-Kategorien

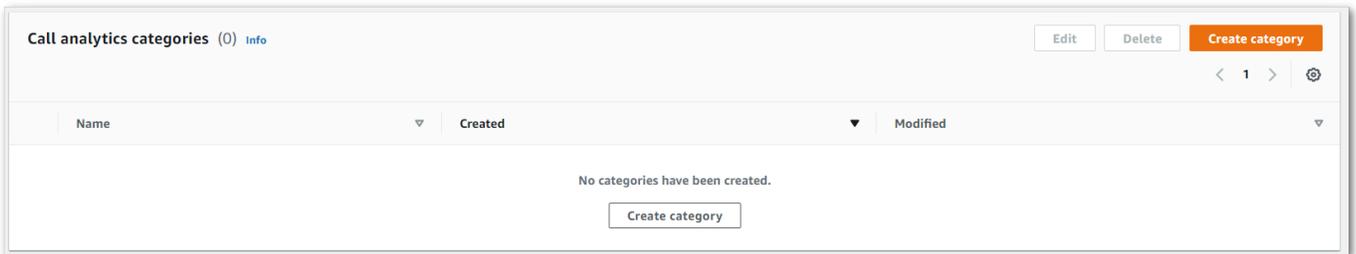
Wenn Sie eine neue Kategorie erstellen, können Sie angeben, ob sie als Nach Anrufen-Kategorie (POST\_CALL) oder als Echtzeitkategorie (REAL\_TIME) erstellt werden soll. Wenn Sie keine Option angeben, wird Ihre Kategorie standardmäßig als Nach Anrufen-Kategorie erstellt. Echtzeit-Kategorieübereinstimmungen können verwendet werden, um Echtzeit-Warnungen 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 Management Console, AWS CLI oder AWS verwenden; Beispiele finden Sie im Folgenden:

### AWS Management Console

1. Wählen Sie im Navigationsbereich Amazon Transcribe unter Amazon Transcribe Call Analytics aus.

- Wählen Sie Anruf-Analytics-Kategorien, um zur Seite Anruf-Analytics-Kategorien zu gelangen. Wählen Sie die Schaltfläche Kategorie erstellen.



- 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 Kategorietyt.

### 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](#)**

Choose category type

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.

Choose a template

- 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.

## 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.

- 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.

**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.

Choose a rule type ▲

**Transcript content match**  
Trigger the rule when the speaker says the words or phrases that you specify.

Add rule

You can add up to 19 more rules.

7. 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 for Python (Boto3)

In diesem Beispiel wird mithilfe der AWS SDK for 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 SDKs AWS](#)

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](#).

## Analyse nach Anrufen mit Echtzeit-Transkriptionen

Die Analyse nach Anrufen ist ein optionales Feature, die mit Echtzeit-Transkriptionen von Call Analytics 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
- **Nicht-Sprachzeit:** Misst Zeiträume, in denen nicht gesprochen wird
- **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 einem Anruf aus einem Audiostream 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 Management Console(nur Demo), HTTP/2 oder verwenden. WebSockets Beispiele finden Sie unter [Start einer Call Analytics in Echtzeit](#).

### Important

Derzeit bietet The AWS Management Console Only eine Demo für Anrufanalysen in Echtzeit mit vorinstallierten Audiobeispielen. Wenn Sie Ihr eigenes Audio verwenden möchten, müssen Sie die API (HTTP/2 oder ein SDK) verwenden. WebSockets

## Beispiel für die Analyse nach Anrufen

Transkripte nach dem Anruf werden segmentweise angezeigt turn-by-turn . 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 HTTP/2 oder WebSockets verwenden. Beispiele finden Sie im Folgenden: AWS Management Console

#### Important

Derzeit bietet The AWS Management Console Only eine Demo für Call Analytics in Echtzeit mit vorinstallierten Audiobeispielen. Wenn Sie Ihr eigenes Audio verwenden möchten, müssen Sie die API (HTTP/2 oder ein SDK) verwenden. WebSockets

#### AWS Management Console

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 Management Console 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

- Wählen Sie für Schritt 1: Eingabe von Audiodaten eine Demo-Testdatei aus dem Dropdown-Menü.



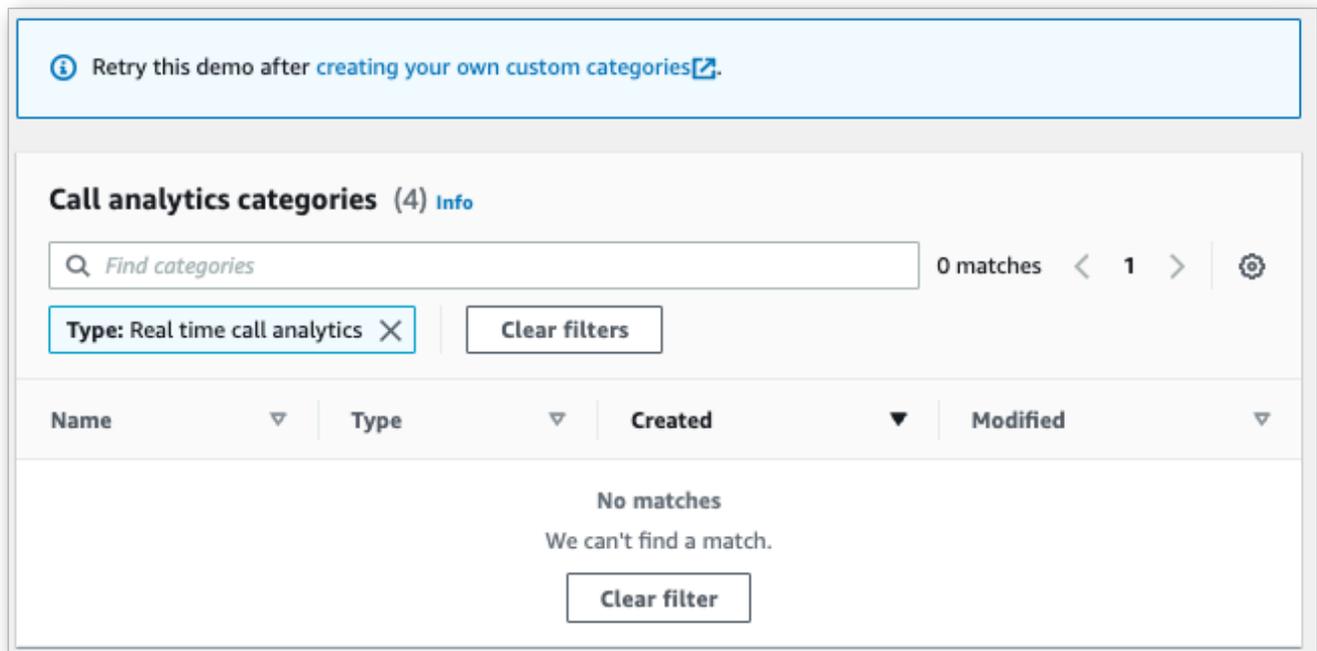
### Step 1: Specify input audio

Input audio file

Insurance complaints (en-US) ▲
Insurance complaints (en-US) ✓
Hospitality complaints (en-US)

- 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

 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 durchzuführen, können Sie die Analyse nach Anrufen aktivieren. Dadurch erhalten Sie dieselben Analysen wie bei einer Analyse nach Anrufen, einschließlich Unterbrechungen, Lautstärke, Nichtgesprächszeit, Gesprächsgeschwindigkeit, Gesprächszeit, Probleme, Maßnahmen und Ergebnisse. Die Analyse nach Anrufen-Ausgabe wird in einer von Ihrem Echtzeit-Call-Analytics-Transkript getrennten Datei 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

Format: s3://bucket, s3://bucket/prefix/, or s3://bucket/prefix/object.

Encryption [Info](#)

**IAM role** [Info](#)

[Create an IAM role](#) that grants access to the output bucket and KMS key (if specified) with the trust policy shown below

▶ Trust Policy

6. Wählen Sie dann Start Streaming (Streamen starten).

## HTTP/2-Stream

In diesem Beispiel wird eine HTTP/2-Anfrage mit aktivierter Call Analytics erstellt. Weitere Informationen zur Verwendung von HTTP/2-Streaming mit finden Sie Amazon Transcribe unter [Einrichten eines HTTP/2-Streams](#) 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://DOC-EXAMPLE-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://DOC-EXAMPLE-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 WebSocket HTTP/2-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.

## Echtzeit-Call-Analytics-Ausgabe

Call Analytics-Transkripte in Echtzeit werden turn-by-turn segmentweise 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
    }
  ],
},

```

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

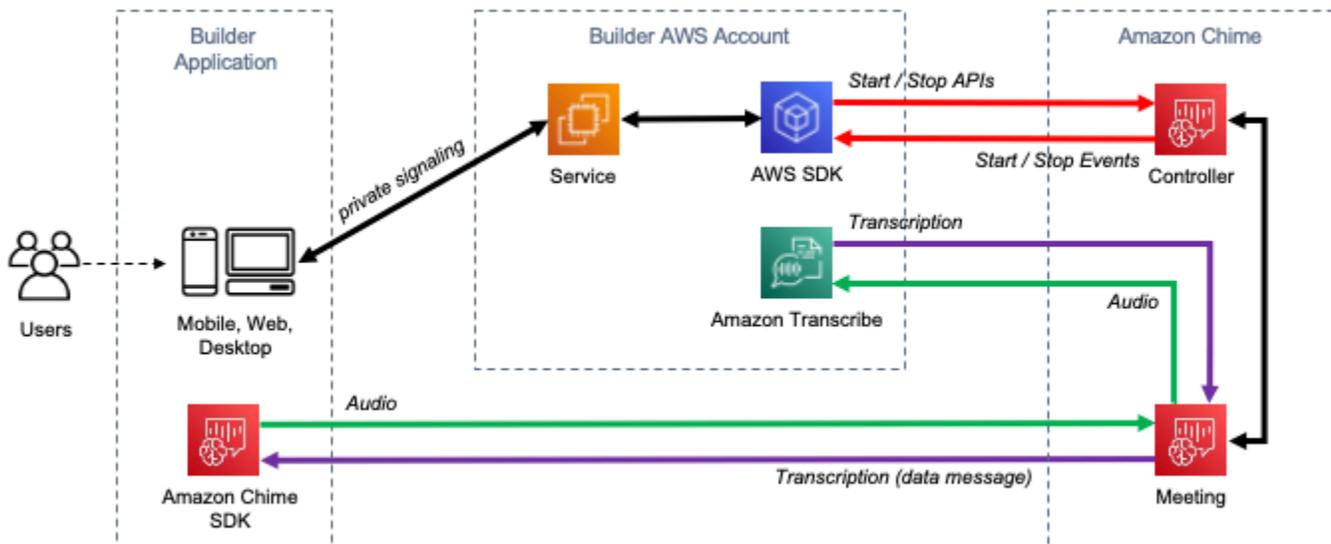
# Transkribieren Ihrer Amazon Chime Anrufe in Echtzeit

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 Audio-Streaming zu 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 dann deren Audio über einen einzigen Stream an Amazon Transcribe zwei separate Kanäle. Die Besprechungsteilnehmer erhalten vom Benutzer zugeschriebene Transkriptionen über Amazon Chime SDK-Datennachrichten. Bereitstellungsbeispiele finden Sie im [Amazon Chime SDK Developer Guide](#).

Der Datenfluss einer Amazon Chime Transkription ist im folgenden Diagramm dargestellt:



Weitere Informationen und detaillierte Anweisungen zum Einrichten von Amazon Chime Echtzeit-Transkriptionen finden Sie [unter Verwenden der Amazon Chime SDK-Live-Transkription](#) im Amazon Chime SDK-Entwicklerhandbuch. Informationen zu API-Vorgängen finden Sie in der [Amazon Chime SDK-API-Referenz](#).

 Tauchen Sie mit dem AWS Machine Learning Blog tiefer ein

Weitere Informationen zur Verbesserung der Genauigkeit mit Transkriptionen in Echtzeit finden Sie unter:

- [Amazon ChimeSDK-Besprechungen unterstützen jetzt die Live-Transkription mit Amazon Transcribe und Amazon Transcribe Medical](#)
- [Amazon ChimeSDK für Telemedizin-Lösung](#)

# Codebeispiele für Amazon Transcribe mit SDKs AWS

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 Servicefunktionen aufrufen, können Sie Aktionen im Kontext der zugehörigen Szenarien und serviceübergreifenden Beispiele sehen.

Szenarien sind Codebeispiele, die Ihnen zeigen, wie Sie eine bestimmte Aufgabe ausführen können, indem Sie mehrere Funktionen innerhalb desselben Services aufrufen.

Serviceübergreifende Beispiele sind Beispielanwendungen, die über mehrere AWS-Services hinweg arbeiten.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

## Codebeispiele

- [Aktionen für Amazon Transcribe mithilfe von SDKs AWS](#)
  - [Verwendung CreateVocabulary mit einem AWS SDK oder CLI](#)
  - [Verwendung DeleteMedicalTranscriptionJob mit einem AWS SDK oder CLI](#)
  - [Verwendung DeleteTranscriptionJob mit einem AWS SDK oder CLI](#)
  - [Verwendung DeleteVocabulary mit einem AWS SDK oder CLI](#)
  - [Verwendung GetTranscriptionJob mit einem AWS SDK oder CLI](#)
  - [Verwendung GetVocabulary mit einem AWS SDK oder CLI](#)
  - [Verwendung ListMedicalTranscriptionJobs mit einem AWS SDK oder CLI](#)
  - [Verwendung ListTranscriptionJobs mit einem AWS SDK oder CLI](#)
  - [Verwendung ListVocabularies mit einem AWS SDK oder CLI](#)
  - [Verwendung StartMedicalTranscriptionJob mit einem AWS SDK oder CLI](#)
  - [Verwendung StartStreamTranscriptionAsync mit einem AWS SDK oder CLI](#)
  - [Verwendung StartTranscriptionJob mit einem AWS SDK oder CLI](#)
  - [Verwendung UpdateVocabulary mit einem AWS SDK oder CLI](#)
- [Szenarien für Amazon Transcribe mit SDKs AWS](#)

- [Erstellen und verfeinern Sie mithilfe eines SDK ein benutzerdefiniertes Amazon Transcribe Transcribe-Vokabular AWS](#)
- [Transkribieren Sie Audio und rufen Sie Auftragsdaten mit Amazon Transcribe mithilfe eines SDK ab AWS](#)
- [Serviceübergreifende Beispiele für Amazon Transcribe mit SDKs AWS](#)
  - [Eine Amazon-Transcribe-App entwickeln](#)
  - [Erstellen einer Amazon-Transcribe-Streaming-App](#)
  - [Mithilfe eines AWS SDK Text in Sprache und zurück in Text umwandeln](#)

## Aktionen für Amazon Transcribe mithilfe von SDKs AWS

Die folgenden Codebeispiele zeigen, wie einzelne Amazon Transcribe Transcribe-Aktionen mit AWS SDKs durchgeführt werden. Diese Auszüge rufen die Amazon Transcribe API auf und sind Codeauszüge aus größeren Programmen, die im Kontext ausgeführt werden müssen. Jedes Beispiel enthält einen Link zu GitHub, über den Sie Anweisungen zum Einrichten und Ausführen des Codes finden.

Die folgenden Beispiele enthalten nur die am häufigsten verwendeten Aktionen. Eine vollständige Liste finden Sie in der [Amazon Transcribe API-Referenz](#).

### Beispiele

- [Verwendung CreateVocabulary mit einem AWS SDK oder CLI](#)
- [Verwendung DeleteMedicalTranscriptionJob mit einem AWS SDK oder CLI](#)
- [Verwendung DeleteTranscriptionJob mit einem AWS SDK oder CLI](#)
- [Verwendung DeleteVocabulary mit einem AWS SDK oder CLI](#)
- [Verwendung GetTranscriptionJob mit einem AWS SDK oder CLI](#)
- [Verwendung GetVocabulary mit einem AWS SDK oder CLI](#)
- [Verwendung ListMedicalTranscriptionJobs mit einem AWS SDK oder CLI](#)
- [Verwendung ListTranscriptionJobs mit einem AWS SDK oder CLI](#)
- [Verwendung ListVocabularies mit einem AWS SDK oder CLI](#)
- [Verwendung StartMedicalTranscriptionJob mit einem AWS SDK oder CLI](#)
- [Verwendung StartStreamTranscriptionAsync mit einem AWS SDK oder CLI](#)
- [Verwendung StartTranscriptionJob mit einem AWS SDK oder CLI](#)

- [Verwendung UpdateVocabulary mit einem AWS SDK oder CLI](#)

## Verwendung **CreateVocabulary** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird `CreateVocabulary`.

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

AWS SDK for .NET

### Note

Es gibt noch mehr dazu [GitHub](#). Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) 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 for .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 dieser 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://DOC-EXAMPLE-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. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) 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.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

## Verwendung **DeleteMedicalTranscriptionJob** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird `DeleteMedicalTranscriptionJob`.

.NET

AWS SDK for .NET

### Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) 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 for .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. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) 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 for JavaScript -Entwicklerhandbuch](#).
- Einzelheiten zur API finden Sie [DeleteMedicalTranscriptionJob](#) in der AWS SDK for JavaScript API-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

## Verwendung **DeleteTranscriptionJob** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird `DeleteTranscriptionJob`.

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

AWS SDK for .NET

### Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) 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 for .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. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) 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 for JavaScript -Entwicklerhandbuch](#).
- Einzelheiten zur API finden Sie [DeleteTranscriptionJob](#) in der AWS SDK for JavaScript API-Referenz.

## Python

### SDK für Python (Boto3)

#### Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) 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.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

## Verwendung **DeleteVocabulary** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird `DeleteVocabulary`.

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

AWS SDK for .NET

### Note

Es gibt noch mehr dazu [GitHub](#). Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) 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 for .NET API-Referenz.

## CLI

### AWS CLI

Löschen eines benutzerdefinierten Vokabulars

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. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) 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.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

## Verwendung **GetTranscriptionJob** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird `GetTranscriptionJob`.

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

#### AWS SDK for .NET

##### Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) 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 for .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://DOC-EXAMPLE-BUCKET/your-audio-file.file-
extension"
    },
    "Transcript": {
      "TranscriptFileUri": "https://Amazon-S3-file-location-of-
transcription-output"
    },
    "StartTime": "2020-09-18T22:27:23.970000+00:00",
    "CreationTime": "2020-09-18T22:27:23.948000+00:00",
    "CompletionTime": "2020-09-18T22:28:21.197000+00:00",
    "Settings": {
      "ChannelIdentification": false,
      "ShowAlternatives": false
    },
    "IdentifyLanguage": true,
    "IdentifiedLanguageScore": 0.8672199249267578
  }
}
```

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. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) 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.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

## Verwendung **GetVocabulary** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird `GetVocabulary`.

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

#### AWS SDK for .NET

##### Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) 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 for .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. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) 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.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

## Verwendung **ListMedicalTranscriptionJobs** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird `ListMedicalTranscriptionJobs`.

.NET

AWS SDK for .NET

### Note

Es gibt noch mehr dazu [GitHub](#). Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) 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 for .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 `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 `--next-token`

```
aws transcribe list-medical-transcription-jobs
```

#### Ausgabe:

```
{
  "NextToken": "3/PblzkiGhzjER3KHuQt2fmbPLF7cDYafjFMEoGn440N/
gsuUSTIkGyanvRE6WMXfd/ZTEc2EZj+P9eii/
z102FDYli6RLI0WoRX4RwMisVrh9G0Kie0Y8ikBCdtqLZB10Wa9McC+eb01
+LaDtZPC4u6ttoHLRlEfzqstHXSgapXg3tEBtm9piIaPB6M0M5BB6t86+qtmocTR/
qrteHZBBudhTfbCwhsxaqujHiiUvFdm3BQbKKKIW06yV9b+4f38oD2lVIan
+vfUs3gBYA15VTDmXXzQPbQ0HPjtwmFI+IWX15nSUjWuN3TUylHgPwzDaYT8qBtu0Z+3UG4V6b
+K2CC0XszXg5rBq9hYgNzy4XoFh/6s5DoSnzq49Q9xHgHdT2yBADFmvFK7myZBs j75+2vQZ0SVpWUPy3WT/32zFAC
+mFYfUjtTZ8n/jq7aQEjQ42A
+X/7K6Jg0cdVPtEg8P1Dr5kgYYG3q30mYXX37U3FZuJmnTI63VtIXsNn0U5eGoY0btpk00Nq9UkzgjSjxqj84ZD5n
+S0EGy9ZUYBJRRcGeYUM3Q4DbSjFuwSAqcFdLIWZdp8qIREMQIBWy7BLwSdyqsQo2vRrd53hm5aWM7SVf6pPq6X/
IXR5+1eU00D8/coaTT4ES2DerbV6RkV4o0VT1d0SdVX/
MmtkNG8nYj8PqU07w7988quh1ZP6D80veJS1q73tUUR9MjnGernW2tAnvnLNhdefBcD
+sZVfYq3iBMFY7wTy1P1G6NqW9GrYDYox3tTPWlD7phpbVSyKrh/
PdYrps5UxnsGoA1b7L/FfAXDfUoGrGUB4N3JsPYXX9D++g+6gV1qBBs/
WfF934aKqfD6UTggm/zV3GA0WiBpfvAZRvEb924i6yGHyMC7y5401ZAwSBupmI
+FFd13CaP04kN1vJlth6aM5vUPXg4BpyUhtbRhwD/KxCvf9K0tLJGyL1A==",
  "MedicalTranscriptionJobSummaries": [
```

```
{
  "MedicalTranscriptionJobName": "vocabulary-dictation-medical-
transcription-job",
  "CreationTime": "2020-09-21T21:17:27.016000+00:00",
  "StartTime": "2020-09-21T21:17:27.045000+00:00",
  "CompletionTime": "2020-09-21T21:17:59.561000+00:00",
  "LanguageCode": "en-US",
  "TranscriptionJobStatus": "COMPLETED",
  "OutputLocationType": "CUSTOMER_BUCKET",
  "Specialty": "PRIMARYCARE",
  "Type": "DICTATION"
},
{
  "MedicalTranscriptionJobName": "alternatives-dictation-medical-
transcription-job",
  "CreationTime": "2020-09-21T21:01:14.569000+00:00",
  "StartTime": "2020-09-21T21:01:14.592000+00:00",
  "CompletionTime": "2020-09-21T21:01:43.606000+00:00",
  "LanguageCode": "en-US",
  "TranscriptionJobStatus": "COMPLETED",
  "OutputLocationType": "CUSTOMER_BUCKET",
  "Specialty": "PRIMARYCARE",
  "Type": "DICTATION"
},
{
  "MedicalTranscriptionJobName": "alternatives-conversation-medical-
transcription-job",
  "CreationTime": "2020-09-21T19:09:18.171000+00:00",
  "StartTime": "2020-09-21T19:09:18.199000+00:00",
  "CompletionTime": "2020-09-21T19:10:22.516000+00:00",
  "LanguageCode": "en-US",
  "TranscriptionJobStatus": "COMPLETED",
  "OutputLocationType": "CUSTOMER_BUCKET",
  "Specialty": "PRIMARYCARE",
  "Type": "CONVERSATION"
},
{
  "MedicalTranscriptionJobName": "speaker-id-conversation-medical-
transcription-job",
  "CreationTime": "2020-09-21T18:43:37.157000+00:00",
  "StartTime": "2020-09-21T18:43:37.265000+00:00",
  "CompletionTime": "2020-09-21T18:44:21.192000+00:00",
  "LanguageCode": "en-US",
  "TranscriptionJobStatus": "COMPLETED",
```

```
        "OutputLocationType": "CUSTOMER_BUCKET",
        "Specialty": "PRIMARYCARE",
        "Type": "CONVERSATION"
    },
    {
        "MedicalTranscriptionJobName": "multichannel-conversation-medical-
transcription-job",
        "CreationTime": "2020-09-20T23:46:44.053000+00:00",
        "StartTime": "2020-09-20T23:46:44.081000+00:00",
        "CompletionTime": "2020-09-20T23:47:35.851000+00:00",
        "LanguageCode": "en-US",
        "TranscriptionJobStatus": "COMPLETED",
        "OutputLocationType": "CUSTOMER_BUCKET",
        "Specialty": "PRIMARYCARE",
        "Type": "CONVERSATION"
    }
]
}
```

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 in der Befehlsreferenz [ListMedicalTranscriptionJobs](#).AWS CLI

## JavaScript

### SDK für JavaScript (v3)

#### Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) 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 for JavaScript -Entwicklerhandbuch](#).

- Einzelheiten zur API finden Sie [ListMedicalTranscriptionJobs](#) in der AWS SDK for JavaScript API-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

## Verwendung **ListTranscriptionJobs** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird `ListTranscriptionJobs`.

.NET

AWS SDK for .NET

### Note

Es gibt noch mehr dazu [GitHub](#). Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) 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 for .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. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) 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. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) 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 for JavaScript -Entwicklerhandbuch](#).
- Einzelheiten zur API finden Sie [ListTranscriptionJobs](#) in der AWS SDK for JavaScript API-Referenz.

## Python

### SDK für Python (Boto3)

#### Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) 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.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

## Verwendung **ListVocabularies** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird `ListVocabularies`.

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

### AWS SDK for .NET

#### Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) 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 for .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. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) 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.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

## Verwendung **StartMedicalTranscriptionJob** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird **StartMedicalTranscriptionJob**.

.NET

AWS SDK for .NET

### Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) 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 for .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": "DOC-EXAMPLE-BUCKET",  
  "Media": {  
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"  
  }  
}
```

Ausgabe:

```
{  
  "MedicalTranscriptionJob": {  
    "MedicalTranscriptionJobName": "simple-dictation-medical-transcription-  
job",  
    "TranscriptionJobStatus": "IN_PROGRESS",  
    "LanguageCode": "language-code",  
    "Media": {  
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"  
    },  
    "StartTime": "2020-09-20T00:35:22.256000+00:00",  
    "CreationTime": "2020-09-20T00:35:22.218000+00:00",  
    "Specialty": "PRIMARYCARE",  
    "Type": "DICTATION"  
  }  
}
```

```
}
```

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": "DOC-EXAMPLE-BUCKET",  
  "Media": {  
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"  
  }  
}
```

Ausgabe:

```
{  
  "MedicalTranscriptionJob": {  
    "MedicalTranscriptionJobName": "simple-conversation-medical-  
transcription-job",  
    "TranscriptionJobStatus": "IN_PROGRESS",  
    "LanguageCode": "language-code",  
    "Media": {  
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"  
    },  
    "StartTime": "2020-09-20T23:19:49.965000+00:00",  
    "CreationTime": "2020-09-20T23:19:49.941000+00:00",  
    "Specialty": "PRIMARYCARE",  
  }  
}
```

```
    "Type": "CONVERSATION"
  }
}
```

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": "DOC-EXAMPLE-BUCKET",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-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://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
},
"StartTime": "2020-09-20T23:46:44.081000+00:00",
"CreationTime": "2020-09-20T23:46:44.053000+00:00",
"Settings": {
  "ChannelIdentification": true
},
"Specialty": "PRIMARYCARE",
"Type": "CONVERSATION"
}
}
```

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": "DOC-EXAMPLE-BUCKET",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-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://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
    },
    "StartTime": "2020-09-21T18:43:37.265000+00:00",
    "CreationTime": "2020-09-21T18:43:37.157000+00:00",
    "Settings": {
      "ShowSpeakerLabels": true,
      "MaxSpeakerLabels": 2
    },
    "Specialty": "PRIMARYCARE",
    "Type": "CONVERSATION"
  }
}

```

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": "DOC-EXAMPLE-BUCKET",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-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://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
    },
    "StartTime": "2020-09-21T19:09:18.199000+00:00",
    "CreationTime": "2020-09-21T19:09:18.171000+00:00",
    "Settings": {
      "ShowAlternatives": true,
      "MaxAlternatives": 2
    },
    "Specialty": "PRIMARYCARE",
    "Type": "CONVERSATION"
  }
}
```

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": "DOC-EXAMPLE-BUCKET",  
  "Media": {  
    "MediaFileUri": "s3://DOC-EXAMPLE-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://DOC-EXAMPLE-BUCKET/your-audio-file.extension"  
    },  
    "StartTime": "2020-09-21T21:01:14.592000+00:00",  
    "CreationTime": "2020-09-21T21:01:14.569000+00:00",  
  }  
}
```

```
    "Settings": {
      "ShowAlternatives": true,
      "MaxAlternatives": 2
    },
    "Specialty": "PRIMARYCARE",
    "Type": "DICTATION"
  }
}
```

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": "DOC-EXAMPLE-BUCKET",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-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://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
    },
    "StartTime": "2020-09-21T21:17:27.045000+00:00",
    "CreationTime": "2020-09-21T21:17:27.016000+00:00",
    "Settings": {
      "VocabularyName": "cli-medical-vocab-1"
    },
    "Specialty": "PRIMARYCARE",
    "Type": "DICTATION"
  }
}
```

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. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) 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 for JavaScript -Entwicklerhandbuch](#).

- Einzelheiten zur API finden Sie [StartMedicalTranscriptionJob](#) in der AWS SDK for JavaScript API-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

## Verwendung **StartStreamTranscriptionAsync** mit einem AWS SDK oder CLI

Das folgende Codebeispiel zeigt, wie es verwendet wird **StartStreamTranscriptionAsync**.

C++

SDK für C++

### Note

Es gibt noch mehr dazu [GitHub](#). Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
int main() {
    Aws::SDKOptions options;

    Aws::InitAPI(options);
    {
        //TODO(User): Set to the region of your AWS account.
        const Aws::String region = Aws::Region::US_WEST_2;

        //Load a profile that has been granted AmazonTranscribeFullAccess AWS
        managed permission policy.
        Aws::Client::ClientConfiguration config;
#ifdef _WIN32
        // ATTENTION: On Windows with the AWS C++ SDK, this example only runs if
        the SDK is built
        // with the curl library.
        // For more information, see the accompanying ReadMe.
        // For more information, see "Building the SDK for Windows with curl".
```

```

// https://docs.aws.amazon.com/sdk-for-cpp/v1/developer-guide/setup-
windows.html
//TODO(User): Update to the location of your .crt file.
config.caFile = "C:/curl/bin/curl-ca-bundle.crt";
#endif

config.region = region;

TranscribeStreamingServiceClient client(config);
StartStreamTranscriptionHandler handler;
handler.SetOnErrorCallback(
    [](const Aws::Client::AWSError<TranscribeStreamingServiceErrors>
&error) {
        std::cerr << "ERROR: " + error.GetMessage() << std::endl;
    });
//SetTranscriptEventCallback called for every 'chunk' of file
transcribed.
// Partial results are returned in real time.
handler.SetTranscriptEventCallback([](const TranscriptEvent &ev) {
    for (auto &r: ev.GetTranscript().GetResults()) {
        if (r.GetIsPartial()) {
            std::cout << "[partial] ";
        }
        else {
            std::cout << "[Final] ";
        }
        for (auto &alt: r.GetAlternatives()) {
            std::cout << alt.GetTranscript() << std::endl;
        }
    }
});

StartStreamTranscriptionRequest request;
request.SetMediaSampleRateHertz(SAMPLE_RATE);
request.SetLanguageCode(LanguageCode::en_US);
request.SetMediaEncoding(
    MediaEncoding::pcm); // wav and aiff files are PCM formats.
request.SetEventStreamHandler(handler);

auto OnStreamReady = [](AudioStream &stream) {
    Aws::FStream file(FILE_NAME, std::ios_base::in |
std::ios_base::binary);
    if (!file.is_open()) {
        std::cerr << "Failed to open " << FILE_NAME << '\n';
    }
}

```

```
std::array<char, BUFFER_SIZE> buf;
int i = 0;
while (file) {
    file.read(&buf[0], buf.size());

    if (!file)
        std::cout << "File: only " << file.gcount() << " could be
read"
                << std::endl;

    Aws::Vector<unsigned char> bits{buf.begin(), buf.end()};
    AudioEvent event(std::move(bits));
    if (!stream) {
        std::cerr << "Failed to create a stream" << std::endl;
        break;
    }
    //The std::basic_istream::gcount() is used to count the
characters in the given string. It returns
//the number of characters extracted by the last read()
operation.

    if (file.gcount() > 0) {
        if (!stream.WriteAudioEvent(event)) {
            std::cerr << "Failed to write an audio event" <<
std::endl;

            break;
        }
    }
    else {
        break;
    }
    std::this_thread::sleep_for(std::chrono::milliseconds(
file.
    25)); // Slow down because we are streaming from a

}
if (!stream.WriteAudioEvent(
    AudioEvent())) {
    // Per the spec, we have to send an empty event (an event
without a payload) at the end.
    std::cerr << "Failed to send an empty frame" << std::endl;
}
else {
    std::cout << "Successfully sent the empty frame" <<
std::endl;
}
}
```

```

        stream.flush();
        stream.Close();
    };

    Aws::Utils::Threading::Semaphore signaling(0 /*initialCount*/, 1 /*maxCount*/);
    auto OnResponseCallback = [&signaling](
        const TranscribeStreamingServiceClient * /*unused*/,
        const Model::StartStreamTranscriptionRequest & /*unused*/,
        const Model::StartStreamTranscriptionOutcome &outcome,
        const std::shared_ptr<const Aws::Client::AsyncCallerContext> & /*unused*/) {

        if (!outcome.IsSuccess()) {
            std::cerr << "Transcribe streaming error "
                << outcome.GetError().GetMessage() << std::endl;
        }

        signaling.Release();
    };

    std::cout << "Starting..." << std::endl;
    client.StartStreamTranscriptionAsync(request, OnStreamReady,
    OnResponseCallback,
                                     nullptr /*context*/);
    signaling.WaitOne(); // Prevent the application from exiting until we're
    done.
    std::cout << "Done" << std::endl;
}

Aws::ShutdownAPI(options);

return 0;
}

```

- Einzelheiten zur API finden Sie [StartStreamTranscriptionAsync](#) in der AWS SDK for C++ API-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

## Verwendung **StartTranscriptionJob** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird `StartTranscriptionJob`.

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

#### AWS SDK for .NET

##### Note

Es gibt noch mehr dazu [GitHub](#). Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) 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 for .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://DOC-EXAMPLE-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://DOC-EXAMPLE-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://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-  
file-name.file-extension"  
    },  
    "StartTime": "2020-09-17T16:07:56.817000+00:00",  
    "CreationTime": "2020-09-17T16:07:56.784000+00:00",  
    "Settings": {
```

```

        "ChannelIdentification": true
    }
}
}

```

Weitere Informationen finden Sie unter [Transkribieren von Mehrkanal-Audio](#) im Amazon-Transcribe-Entwicklerhandbuch.

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://DOC-EXAMPLE-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://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-
file-name.file-extension"
    }
  }
}

```

```

    },
    "StartTime": "2020-09-17T16:22:59.696000+00:00",
    "CreationTime": "2020-09-17T16:22:59.676000+00:00",
    "Settings": {
      "ShowSpeakerLabels": true,
      "MaxSpeakerLabels": 2
    }
  }
}

```

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://DOC-EXAMPLE-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://DOC-EXAMPLE-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://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-
file-name.file-extension"
    },
    "StartTime": "2020-09-18T16:36:18.568000+00:00",
    "CreationTime": "2020-09-18T16:36:18.547000+00:00",
    "Settings": {
      "VocabularyFilterName": "your-vocabulary-filter",
      "VocabularyFilterMethod": "remove"
    }
  }
}
```

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://DOC-EXAMPLE-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://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-
file-name.file-extension"
    },
    "StartTime": "2020-09-18T16:36:18.568000+00:00",
    "CreationTime": "2020-09-18T16:36:18.547000+00:00",
    "Settings": {
      "VocabularyName": "your-vocabulary"
    }
  }
}

```

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://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-
file-name.file-extension"
  }
}
```

Ausgabe:

```
{
  "TranscriptionJob": {
    "TranscriptionJobName": "cli-identify-language-transcription-job",
    "TranscriptionJobStatus": "IN_PROGRESS",
    "Media": {
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-
file-name.file-extension"
    },
    "StartTime": "2020-09-18T22:27:23.970000+00:00",
    "CreationTime": "2020-09-18T22:27:23.948000+00:00",
    "IdentifyLanguage": true
  }
}
```

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 ungeschwärtzten 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://DOC-EXAMPLE-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://DOC-EXAMPLE-BUCKET/your-audio-file.file-  
extension"  
    },  
    "StartTime": "2020-09-28T17:56:01.835000+00:00",  
    "CreationTime": "2020-09-28T17:56:01.801000+00:00",  
    "ModelSettings": {  
      "LanguageModelName": "cli-clm-2"  
    }  
  }  
}
```

Weitere Informationen finden Sie unter [Verbessern der domänenspezifischen Transkriptionsgenauigkeit mit benutzerdefinierten Sprachmodellen](#) im Amazon-Transcribe-Entwicklerhandbuch.

- Einzelheiten zur API finden Sie unter [StartTranscriptionJob AWS CLI Befehlsreferenz](#).

## Java

### SDK für Java 2.x

#### Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
public class TranscribeStreamingDemoApp {
    private static final Region REGION = Region.US_EAST_1;
    private static TranscribeStreamingAsyncClient client;

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

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

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

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

    private static InputStream getStreamFromMic() throws LineUnavailableException
    {

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

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

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

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

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

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

private static StartStreamTranscriptionResponseHandler getResponseHandler() {
    return StartStreamTranscriptionResponseHandler.builder()
        .onResponse(r -> {
            System.out.println("Received Initial response");
        })
        .onError(e -> {
            System.out.println(e.getMessage());
            StringWriter sw = new StringWriter();
            e.printStackTrace(new PrintWriter(sw));
            System.out.println("Error Occurred: " + sw.toString());
        })
        .onComplete(() -> {
            System.out.println("=== All records stream successfully
===");
        })
        .subscriber(event -> {
            List<Result> results = ((TranscriptEvent)
event).transcript().results();
```

```
        if (results.size() > 0) {
            if (!
results.get(0).alternatives().get(0).transcript().isEmpty()) {

                System.out.println(results.get(0).alternatives().get(0).transcript());
            }
        }
    })
    .build();
}

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

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

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

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

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

public static class SubscriptionImpl implements Subscription {
```

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

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

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

    demand.getAndAdd(n);

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

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

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

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

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

    return audioBuffer;
}

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

- Einzelheiten zur API finden Sie [StartTranscriptionJob](#) in der AWS SDK for Java 2.x API-Referenz.

## JavaScript

### SDK für JavaScript (v3)

#### Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) 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 for JavaScript -Entwicklerhandbuch](#).
- Einzelheiten zur API finden Sie [StartTranscriptionJob](#) in der AWS SDK for JavaScript API-Referenz.

## Python

### SDK für Python (Boto3)

#### Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) 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.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

## Verwendung **UpdateVocabulary** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird `UpdateVocabulary`.

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

### AWS SDK for .NET

#### Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) 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 for .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://DOC-EXAMPLE-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. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) 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.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

## Szenarien für Amazon Transcribe mit SDKs AWS

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 erledigen können. Jedes Szenario enthält einen Link zu GitHub, wo Sie Anweisungen zur Einrichtung und Ausführung des Codes finden.

## Beispiele

- [Erstellen und verfeinern Sie mithilfe eines SDK ein benutzerdefiniertes Amazon Transcribe Transcribe-Vokabular AWS](#)
- [Transkribieren Sie Audio und rufen Sie Auftragsdaten mit Amazon Transcribe mithilfe eines SDK ab AWS](#)

## Erstellen und verfeinern Sie mithilfe eines SDK ein benutzerdefiniertes Amazon Transcribe Transcribe-Vokabular AWS

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. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) 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_vocabularies(vocabulary_filter, transcribe_client):
    """
    Lists the custom vocabularies created for this AWS account.

    :param vocabulary_filter: The returned vocabularies must contain this string
    in
                               their names.
    :param transcribe_client: The Boto3 Transcribe client.
    :return: The list of retrieved vocabularies.
    """
    try:
        response =
transcribe_client.list_vocabularies(NameContains=vocabulary_filter)
        vocabs = response["Vocabularies"]
        next_token = response.get("NextToken")
        while next_token is not None:
            response = transcribe_client.list_vocabularies(
                NameContains=vocabulary_filter, NextToken=next_token
            )
            vocabs += response["Vocabularies"]
            next_token = response.get("NextToken")
        logger.info(
            "Got %s vocabularies with filter %s.", len(vocabs), vocabulary_filter
        )
    except ClientError:
        logger.exception(
            "Couldn't list vocabularies with filter %s.", vocabulary_filter
        )
        raise
    else:
        return vocabs

def delete_vocabulary(vocabulary_name, transcribe_client):
    """
    Deletes a custom vocabulary.

    :param vocabulary_name: The name of the vocabulary to delete.
    :param transcribe_client: The Boto3 Transcribe client.
    """
    try:
        transcribe_client.delete_vocabulary(VocabularyName=vocabulary_name)
        logger.info("Deleted vocabulary %s.", vocabulary_name)
```

```
except ClientError:
    logger.exception("Couldn't delete vocabulary %s.", vocabulary_name)
    raise
```

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 SDK AWS](#). 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 SDK ab AWS

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. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) 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.toString());
        })
        .onComplete(() -> {
            System.out.println("=== All records stream successfully
===");
        });
}
```

```
        })
        .subscriber(event -> {
            List<Result> results = ((TranscriptEvent)
event).transcript().results();
            if (results.size() > 0) {
                if (!
results.get(0).alternatives().get(0).transcript().isEmpty()) {

System.out.println(results.get(0).alternatives().get(0).transcript());
                }
            }
        })
        .build();
    }

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

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

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

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

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

```
    }  
}  
  
public static class SubscriptionImpl implements Subscription {  
    private static final int CHUNK_SIZE_IN_BYTES = 1024 * 1;  
    private final Subscriber<? super AudioStream> subscriber;  
    private final InputStream inputStream;  
    private ExecutorService executor = Executors.newFixedThreadPool(1);  
    private AtomicLong demand = new AtomicLong(0);  
  
    SubscriptionImpl(Subscriber<? super AudioStream> s, InputStream  
inputStream) {  
        this.subscriber = s;  
        this.inputStream = inputStream;  
    }  
  
    @Override  
    public void request(long n) {  
        if (n <= 0) {  
            subscriber.onError(new IllegalArgumentException("Demand must be  
positive"));  
        }  
  
        demand.getAndAdd(n);  
  
        executor.submit(() -> {  
            try {  
                do {  
                    ByteBuffer audioBuffer = getNextEvent();  
                    if (audioBuffer.remaining() > 0) {  
                        AudioEvent audioEvent =  
audioEventFromBuffer(audioBuffer);  
                        subscriber.onNext(audioEvent);  
                    } else {  
                        subscriber.onComplete();  
                        break;  
                    }  
                } while (demand.decrementAndGet() > 0);  
            } catch (Exception e) {  
                subscriber.onError(e);  
            }  
        });  
    }  
}
```

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

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

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

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

    return audioBuffer;
}

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

- API-Details 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. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) 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 SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

## Serviceübergreifende Beispiele für Amazon Transcribe mit SDKs AWS

Die folgenden Beispielanwendungen verwenden AWS SDKs, um Amazon Transcribe mit anderen zu kombinieren. AWS-Services Jedes Beispiel enthält einen Link zu GitHub, über den Sie Anweisungen zur Einrichtung und Ausführung der Anwendung finden.

### Beispiele

- [Eine Amazon-Transcribe-App entwickeln](#)
- [Erstellen einer Amazon-Transcribe-Streaming-App](#)
- [Mithilfe eines AWS SDK Text in Sprache und zurück in Text umwandeln](#)

## Eine Amazon-Transcribe-App entwickeln

Das folgende Codebeispiel zeigt, wie Amazon Transcribe verwendet wird, um Sprachaufnahmen im Browser zu transkribieren und anzuzeigen.

## JavaScript

### SDK für JavaScript (v3)

Erstellen Sie eine App, die Amazon Transcribe verwendet, um Sprachaufnahmen im Browser zu transkribieren und anzuzeigen. Die App verwendet zwei Amazon Simple Storage Service (Amazon S3)-Buckets, einen zum Hosten des Anwendungscodes und einen zum Speichern von Transkriptionen. Die App verwendet einen Amazon-Cognito-Benutzerpool zur Authentifizierung Ihrer Benutzer. Authentifizierte Benutzer verfügen über AWS Identity and Access Management (IAM-) Berechtigungen für den Zugriff auf die erforderlichen AWS Dienste.

Den vollständigen Quellcode und Anweisungen zur Einrichtung und Ausführung finden Sie im vollständigen Beispiel unter [GitHub](#)

Dieses Beispiel ist auch verfügbar im [AWS SDK for JavaScript Entwicklerhandbuch für v3](#).

In diesem Beispiel verwendete Dienste

- Amazon Cognito Identity
- Amazon S3
- Amazon Transcribe

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Dienst mit einem SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

## 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 SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

## Mithilfe eines AWS SDK Text in Sprache und zurück in Text umwandeln

Wie das aussehen kann, sehen Sie am nachfolgenden Beispielcode:

- Verwenden Sie Amazon Polly, um eine Nur-Text-Eingabedatei (UTF-8) in eine 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 Klartext-Eingabedatei (UTF-8) in eine Audiodatei zu synthetisieren, die Audiodatei in einen Amazon-S3-Bucket hochzuladen, diese Audiodatei mit Amazon Transcribe 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 SDK AWS](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

# Sicherheit in Amazon Transcribe

Die Sicherheit in der Cloud hat bei AWS höchste Priorität. Als AWS-Kunde profitieren Sie von einer Rechenzentrums- und Netzwerkarchitektur, die eingerichtet wurde, um die Anforderungen der anspruchsvollsten Organisationen in puncto Sicherheit zu erfüllen.

Sicherheit ist eine übergreifende Verantwortlichkeit zwischen AWS und Ihnen. Das [Modell der geteilten Verantwortung](#) beschreibt dies als Sicherheit der Cloud selbst und Sicherheit in der Cloud:

- **Sicherheit der Cloud:** AWS ist zuständig für den Schutz der Infrastruktur, die AWS -Services in der AWS Cloud ausführt. AWS stellt Ihnen außerdem Services bereit, die Sie sicher verwenden können. Auditoren von Drittanbietern testen und überprüfen die Effektivität unserer Sicherheitsmaßnahmen im Rahmen der [AWS-Compliance-Programme](#) regelmäßig. Informationen zu den Compliance-Programmen, die für Amazon Transcribe gelten, finden Sie unter [Im Rahmen des Compliance-Programms zugelassene AWS-Services](#).
- **Sicherheit in der Cloud:** Ihr Verantwortungsumfang wird durch den AWS -Service bestimmt, den Sie verwenden. 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. Die folgenden Themen veranschaulichen, wie Sie Amazon Transcribe zur Erfüllung Ihrer Sicherheits- und Compliance-Ziele konfigurieren können. Sie erfahren außerdem, wie Sie andere AWS -Services verwenden können, um Ihre Amazon Transcribe -Ressourcen zu überwachen und zu schützen.

## Themen

- [Identity and Access Management für Amazon Transcribe](#)
- [Datenschutz in Amazon Transcribe](#)
- [Überwachung Amazon Transcribe](#)
- [Konformitätsvalidierung für Amazon Transcribe](#)
- [Ausfallsicherheit in Amazon Transcribe](#)
- [Sicherheit der Infrastruktur in Amazon Transcribe](#)
- [Schwachstellenanalyse und -management in Amazon Transcribe](#)
- [Bewährte Methoden für die Sicherheit 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](#)
- [Vermeidung des Problems des verwirrten Stellvertreters \(dienstübergreifend\)](#)
- [Amazon TranscribeBeispiele für identitätsbasierte -Richtlinien](#)
- [Fehlerbehebung bei Amazon Transcribe Identität und Zugriff](#)

## Zielgruppe

Die Art und Weise, wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von der Arbeit ab, in der Sie tätig sind. Amazon Transcribe

**Dienstbenutzer** — Wenn Sie den Amazon Transcribe Dienst für Ihre Arbeit verwenden, stellt Ihnen Ihr Administrator die erforderlichen Anmeldeinformationen und Berechtigungen zur Verfügung. Wenn Sie für Ihre Arbeit mehr Amazon Transcribe Funktionen verwenden, benötigen Sie möglicherweise zusätzliche Berechtigungen. Wenn Sie die Funktionsweise der Zugriffskontrolle nachvollziehen, wissen Sie bereits, welche Berechtigungen Sie von Ihrem Administrator anfordern müssen. Unter [Fehlerbehebung bei Amazon Transcribe Identität und Zugriff](#) finden Sie nützliche Informationen für den Fall, dass Sie keinen Zugriff auf eine Feature in Amazon Transcribe haben.

**Serviceadministrator** — Wenn Sie in Ihrem Unternehmen für die Amazon Transcribe Ressourcen verantwortlich sind, haben Sie wahrscheinlich vollen Zugriff auf Amazon Transcribe. Es ist Ihre Aufgabe, zu bestimmen, auf welche Amazon Transcribe Funktionen und Ressourcen Ihre Servicebenutzer zugreifen sollen. Sie müssen dann Anträge an Ihren IAM-Administrator stellen, um die Berechtigungen Ihrer Servicenutzer zu ändern. Lesen Sie die Informationen auf dieser Seite, um

die Grundkonzepte von IAM nachzuvollziehen. Weitere Informationen darüber, wie Ihr Unternehmen IAM nutzen kann Amazon Transcribe, finden Sie unter [Wie Amazon Transcribe funktioniert mit IAM](#).

IAM-Administrator: Wenn Sie als IAM-Administrator fungieren, sollten Sie Einzelheiten dazu kennen, wie Sie Richtlinien zur Verwaltung des Zugriffs auf Amazon Transcribe verfassen können. Beispiele für Amazon Transcribe identitätsbasierte Richtlinien, die Sie in IAM verwenden können, finden Sie unter [Amazon Transcribe Beispiele für identitätsbasierte -Richtlinien](#)

## Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich AWS mit Ihren Identitätsdaten anmelden. Sie müssen als IAM-Benutzer authentifiziert (angemeldet AWS) sein oder eine IAM-Rolle annehmen. Root-Benutzer des AWS-Kontos

Sie können sich AWS als föderierte Identität anmelden, indem Sie Anmeldeinformationen verwenden, die über eine Identitätsquelle bereitgestellt wurden. AWS IAM Identity Center (IAM Identity Center) -Benutzer, die Single Sign-On-Authentifizierung Ihres Unternehmens und Ihre Google- oder Facebook-Anmeldeinformationen sind Beispiele für föderierte Identitäten. Wenn Sie sich als Verbundidentität anmelden, hat der Administrator vorher mithilfe von IAM-Rollen einen Identitätsverbund eingerichtet. Wenn Sie über den Verbund darauf zugreifen AWS, übernehmen Sie indirekt eine Rolle.

Je nachdem, welcher Benutzertyp Sie sind, können Sie sich beim AWS Management Console oder beim AWS Zugangsportale anmelden. Weitere Informationen zur Anmeldung finden Sie AWS unter [So melden Sie sich bei Ihrem an AWS-Konto](#) im AWS-Anmeldung Benutzerhandbuch.

Wenn Sie AWS programmgesteuert darauf zugreifen, AWS stellt es ein Software Development Kit (SDK) und eine Befehlszeilenschnittstelle (CLI) bereit, mit denen Sie Ihre Anfragen mithilfe Ihrer Anmeldeinformationen kryptografisch signieren können. Wenn Sie keine AWS Tools verwenden, müssen Sie Anfragen selbst signieren. Weitere Informationen zur Verwendung der empfohlenen Methode, um Anfragen selbst zu [signieren, finden Sie im IAM-Benutzerhandbuch unter AWS API-Anfragen](#) signieren.

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen angeben. AWS empfiehlt beispielsweise, die Multi-Faktor-Authentifizierung (MFA) zu verwenden, um die Sicherheit Ihres Kontos zu erhöhen. Weitere Informationen finden Sie unter [Multi-Faktor-Authentifizierung](#) im AWS IAM Identity Center - Benutzerhandbuch und [Verwenden der Multi-Faktor-Authentifizierung \(MFA\) in AWS](#) im IAM-Benutzerhandbuch.

## AWS-Konto Root-Benutzer

Wenn Sie ein AWS-Konto erstellen, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle AWS-Services Ressourcen im 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. Schützen Sie Ihre Root-Benutzer-Anmeldeinformationen und verwenden Sie diese, um die Aufgaben auszuführen, die nur der Root-Benutzer ausführen kann. Eine vollständige Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter [Aufgaben, die Root-Benutzer-Anmeldeinformationen erfordern](#) im IAM-Benutzerhandbuch.

## Verbundidentität

Als bewährte Methode sollten menschliche Benutzer, einschließlich Benutzer, die Administratorzugriff benötigen, für den Zugriff AWS-Services mithilfe temporärer Anmeldeinformationen den Verbund mit einem Identitätsanbieter verwenden.

Eine föderierte Identität ist ein Benutzer aus Ihrem Unternehmensbenutzerverzeichnis, einem Web-Identitätsanbieter AWS Directory Service, dem Identity Center-Verzeichnis oder einem beliebigen Benutzer, der mithilfe AWS-Services von Anmeldeinformationen zugreift, die über eine Identitätsquelle bereitgestellt wurden. Wenn föderierte Identitäten darauf zugreifen AWS-Konten, 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 AWS-Konten Anwendungen zu verwenden. Informationen zu IAM Identity Center finden Sie unter [Was ist IAM Identity Center?](#) im AWS IAM Identity Center - Benutzerhandbuch.

## IAM-Benutzer und -Gruppen

Ein [IAM-Benutzer](#) ist eine Identität innerhalb Ihres Unternehmens AWS-Konto, die über spezifische Berechtigungen für eine einzelne Person oder Anwendung verfügt. Wenn möglich, empfehlen wir, temporäre Anmeldeinformationen zu verwenden, anstatt IAM-Benutzer zu erstellen, die langfristige Anmeldeinformationen wie Passwörter und Zugriffsschlüssel haben. Bei speziellen Anwendungsfällen, die langfristige Anmeldeinformationen mit IAM-Benutzern erfordern, empfehlen wir jedoch, die Zugriffsschlüssel zu rotieren. Weitere Informationen finden Sie unter [Regelmäßiges](#)

[Rotieren von Zugriffsschlüsseln für Anwendungsfälle, die langfristige Anmeldeinformationen erfordern](#) im IAM-Benutzerhandbuch.

Eine [IAM-Gruppe](#) ist eine Identität, die eine Sammlung von IAM-Benutzern angibt. Sie können sich nicht als Gruppe anmelden. Mithilfe von Gruppen können Sie Berechtigungen für mehrere Benutzer gleichzeitig angeben. Gruppen vereinfachen die Verwaltung von Berechtigungen, wenn es zahlreiche Benutzer gibt. Sie könnten beispielsweise einer Gruppe mit dem Namen IAMAdmins Berechtigungen zum Verwalten von IAM-Ressourcen erteilen.

Benutzer unterscheiden sich von Rollen. Ein Benutzer ist einer einzigen Person oder Anwendung eindeutig zugeordnet. Eine Rolle kann von allen Personen angenommen werden, die sie benötigen. Benutzer besitzen dauerhafte Anmeldeinformationen. Rollen stellen temporäre Anmeldeinformationen bereit. Weitere Informationen finden Sie unter [Erstellen eines IAM-Benutzers \(anstatt einer Rolle\)](#) im IAM-Benutzerhandbuch.

## IAM-Rollen

Eine [IAM-Rolle](#) ist eine Identität innerhalb Ihres Unternehmens AWS-Konto, die über bestimmte Berechtigungen verfügt. Sie ist einem IAM-Benutzer vergleichbar, ist aber nicht mit einer bestimmten Person verknüpft. Sie können vorübergehend eine IAM-Rolle in der übernehmen, AWS Management Console indem Sie die Rollen [wechseln](#). Sie können eine Rolle übernehmen, indem Sie eine AWS CLI oder AWS API-Operation aufrufen oder eine benutzerdefinierte URL verwenden. Weitere Informationen zu Methoden für die Verwendung von Rollen finden Sie unter [Verwenden von IAM-Rollen](#) im IAM-Benutzerhandbuch.

IAM-Rollen mit temporären Anmeldeinformationen sind in folgenden Situationen hilfreich:

- **Verbundbenutzerzugriff** – Um einer Verbundidentität Berechtigungen zuzuweisen, erstellen Sie eine Rolle und definieren Berechtigungen für die Rolle. Wird eine Verbundidentität authentifiziert, so wird die Identität der Rolle zugeordnet und erhält die von der Rolle definierten Berechtigungen. Informationen zu Rollen für den Verbund finden Sie unter [Erstellen von Rollen für externe Identitätsanbieter](#) im IAM-Benutzerhandbuch. Wenn Sie IAM Identity Center verwenden, konfigurieren Sie einen Berechtigungssatz. Wenn Sie steuern möchten, worauf Ihre Identitäten nach der Authentifizierung zugreifen können, korreliert IAM Identity Center den Berechtigungssatz mit einer Rolle in IAM. Informationen zu Berechtigungssätzen finden Sie unter [Berechtigungssätze](#) im AWS IAM Identity Center -Benutzerhandbuch.
- **Temporäre IAM-Benutzerberechtigungen** – Ein IAM-Benutzer oder eine -Rolle kann eine IAM-Rolle übernehmen, um vorübergehend andere Berechtigungen für eine bestimmte Aufgabe zu erhalten.

- **Kontoübergreifender Zugriff** – Sie können eine IAM-Rolle verwenden, um einem vertrauenswürdigen Prinzipal in einem anderen Konto den Zugriff auf Ressourcen in Ihrem Konto zu ermöglichen. Rollen stellen die primäre Möglichkeit dar, um kontoübergreifendem Zugriff zu gewähren. Bei einigen können Sie AWS-Services jedoch eine Richtlinie direkt an eine Ressource anhängen (anstatt eine Rolle als Proxy zu verwenden). Informationen zum Unterschied zwischen Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter [Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch](#).
- **Serviceübergreifender Zugriff** — Einige verwenden Funktionen in anderen. AWS-Services AWS-Services Wenn Sie beispielsweise einen Aufruf in einem Service tätigen, führt dieser Service häufig Anwendungen in Amazon-EC2 aus oder speichert Objekte in Amazon-S3. Ein Dienst kann dies mit den Berechtigungen des aufrufenden Prinzipals mit einer Servicerolle oder mit einer serviceverknüpften Rolle tun.
- **Forward Access Sessions (FAS)** — Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, in Kombination mit der Anfrage, Anfragen an AWS-Service nachgelagerte Dienste zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter [Zugriffssitzungen weiterleiten](#).
- **Servicerolle** – Eine Servicerolle ist eine [IAM-Rolle](#), die ein Service übernimmt, 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.
- **Dienstbezogene Rolle** — Eine dienstbezogene Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Servicebezogene 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.
- **Anwendungen, die auf Amazon EC2 ausgeführt werden** — Sie können eine IAM-Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2-Instance ausgeführt werden und API-Anfragen stellen AWS CLI . AWS Das ist eher zu empfehlen, als Zugriffsschlüssel innerhalb der EC2-Instance zu speichern. Um einer EC2-Instance eine AWS Rolle zuzuweisen und sie allen ihren Anwendungen zur Verfügung zu stellen, erstellen Sie

ein Instance-Profil, das an die Instance angehängt ist. Ein Instance-Profil enthält die Rolle und ermöglicht, dass Programme, die in der EC2-Instance ausgeführt werden, temporäre Anmeldeinformationen erhalten. Weitere Informationen finden Sie unter [Verwenden einer IAM-Rolle zum Erteilen von Berechtigungen für Anwendungen, die auf Amazon-EC2-Instances ausgeführt werden](#) im IAM-Benutzerhandbuch.

Informationen dazu, wann Sie IAM-Rollen oder IAM-Benutzer verwenden sollten, finden Sie unter [Erstellen einer IAM-Rolle \(anstatt eines Benutzers\)](#) 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 ist ein Objekt, AWS das, wenn es einer Identität oder Ressource zugeordnet ist, deren Berechtigungen definiert. AWS wertet diese Richtlinien aus, wenn ein Prinzipal (Benutzer, Root-Benutzer oder Rollensitzung) eine Anfrage stellt. Berechtigungen in den Richtlinien bestimmen, ob die Anforderung zugelassen oder abgelehnt wird. Die meisten Richtlinien werden AWS als JSON-Dokumente gespeichert. Weitere Informationen zu Struktur und Inhalten von JSON-Richtliniendokumenten finden Sie unter [Übersicht über JSON-Richtlinien](#) im IAM-Benutzerhandbuch.

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

IAM-Richtlinien definieren Berechtigungen für eine Aktion unabhängig von der Methode, die Sie zur Ausführung der Aktion verwenden. Angenommen, es gibt eine Richtlinie, die Berechtigungen für die `iam:GetRole`-Aktion erteilt. Ein Benutzer mit dieser Richtlinie kann Rolleninformationen von der AWS Management Console AWS CLI, der oder der AWS API abrufen.

## Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. 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 [Erstellen von IAM-Richtlinien](#) im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien können weiter als Inline-Richtlinien oder verwaltete Richtlinien kategorisiert werden. Inline-Richtlinien sind direkt in einen einzelnen Benutzer, eine einzelne Gruppe oder eine einzelne Rolle eingebettet. Verwaltete Richtlinien sind eigenständige Richtlinien, die Sie mehreren Benutzern, Gruppen und Rollen in Ihrem System zuordnen können AWS-Konto. Zu den verwalteten Richtlinien gehören AWS verwaltete Richtlinien und vom Kunden verwaltete Richtlinien. Informationen dazu, wie Sie zwischen einer verwalteten Richtlinie und einer eingebundenen Richtlinie wählen, finden Sie unter [Auswahl zwischen verwalteten und eingebundenen Richtlinien](#) im IAM-Benutzerhandbuch.

## Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. 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

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können AWS verwaltete Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

## Zugriffssteuerungslisten (ACLs)

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.

Amazon S3 und Amazon VPC sind Beispiele für Services, die ACLs unterstützen. AWS WAF Weitere Informationen“ zu ACLs finden Sie unter [Zugriffskontrollliste \(ACL\) – Übersicht](#) (Access Control List) im Amazon-Simple-Storage-Service-Entwicklerhandbuch.

## Weitere Richtlinientypen

AWS unterstützt zusätzliche, weniger verbreitete Richtlinientypen. Diese Richtlinientypen können die maximalen Berechtigungen festlegen, die Ihnen von den häufiger verwendeten Richtlinientypen erteilt werden können.

- **Berechtigungsgrenzen** – Eine Berechtigungsgrenze ist ein erweitertes Feature, mit der Sie die maximalen Berechtigungen festlegen können, die eine identitätsbasierte Richtlinie einer IAM-Entität (IAM-Benutzer oder -Rolle) erteilen kann. Sie können eine Berechtigungsgrenze für eine Entität festlegen. Die daraus resultierenden Berechtigungen sind der Schnittpunkt der identitätsbasierten Richtlinien einer Entität und ihrer Berechtigungsgrenzen. Ressourcenbasierte Richtlinien, die den Benutzer oder die Rolle im Feld `Principal` angeben, werden nicht durch Berechtigungsgrenzen eingeschränkt. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen über Berechtigungsgrenzen finden Sie unter [Berechtigungsgrenzen für IAM-Entitäten](#) im IAM-Benutzerhandbuch.
- **Service Control Policies (SCPs)** — SCPs sind JSON-Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OU) in festlegen. AWS Organizations AWS Organizations ist ein Dienst zur Gruppierung und zentralen Verwaltung mehrerer Objekte AWS-Konten , die Ihrem Unternehmen gehören. Wenn Sie innerhalb einer Organisation alle Features aktivieren, können Sie Service-Kontrollrichtlinien (SCPs) auf alle oder einzelne Ihrer Konten anwenden. Das SCP schränkt die Berechtigungen für Entitäten in Mitgliedskonten ein, einschließlich der einzelnen Entitäten. Root-Benutzer des AWS-Kontos Weitere Informationen zu Organizations und SCPs finden Sie unter [Funktionsweise von SCPs](#) im AWS Organizations -Benutzerhandbuch.
- **Sitzungsrichtlinien** – Sitzungsrichtlinien sind erweiterte Richtlinien, die Sie als Parameter übergeben, wenn Sie eine temporäre Sitzung für eine Rolle oder einen verbundenen Benutzer programmgesteuert erstellen. Die resultierenden Sitzungsberechtigungen sind eine Schnittmenge der auf der Identität des Benutzers oder der Rolle basierenden Richtlinien und der Sitzungsrichtlinien. Berechtigungen können auch aus einer ressourcenbasierten Richtlinie stammen. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen finden Sie unter [Sitzungsrichtlinien](#) im IAM-Benutzerhandbuch.

## Mehrere Richtlinientypen

Wenn mehrere auf eine Anforderung mehrere Richtlinientypen angewendet werden können, sind die entsprechenden Berechtigungen komplizierter. Informationen darüber, wie AWS bestimmt wird,

ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie im IAM-Benutzerhandbuch unter [Bewertungslogik für Richtlinien](#).

## Wie Amazon Transcribe funktioniert mit IAM

Bevor Sie IAM zur Verwaltung des Zugriffs auf verwenden, sollten Sie sich darüber informieren Amazon Transcribe, mit welchen IAM-Funktionen Sie arbeiten können. Amazon Transcribe

IAM Funktionen, die Sie mit verwenden können Amazon Transcribe

IAM Merkmal	Amazon Transcribe Unterstützung
<a href="#">Identitätsbasierte Richtlinien</a>	Ja
<a href="#">Ressourcenbasierte Richtlinien</a>	Nein
<a href="#">Richtlinienaktionen</a>	Ja
<a href="#">Richtlinienressourcen</a>	Ja
<a href="#">Richtlinienbedingungsschlüssel (servicespezifisch)</a>	Ja
<a href="#">ACLs</a>	Nein
<a href="#">ABAC (Tags in Richtlinien)</a>	Teilweise
<a href="#">Temporäre Anmeldeinformationen</a>	Ja
<a href="#">Hauptberechtigungen</a>	Ja
<a href="#">Servicerollen</a>	Ja
<a href="#">Service-verknüpfte Rollen</a>	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](#).

## Identitätsbasierte Richtlinien für Amazon Transcribe

Unterstützt Richtlinien auf Identitätsbasis. Ja

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. 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 [Erstellen von IAM-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. Sie können den Prinzipal nicht in einer identitätsbasierten Richtlinie angeben, da er für den Benutzer oder die Rolle gilt, dem er zugeordnet ist. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der [IAM-Referenz für JSON-Richtlinienelemente](#) im IAM-Benutzerhandbuch.

### Beispiele für identitätsbasierte Richtlinien für Amazon Transcribe

Beispiele für Amazon Transcribe identitätsbasierte Richtlinien finden Sie unter [Amazon TranscribeBeispiele für identitätsbasierte -Richtlinien](#)

## Ressourcenbasierte Richtlinien in Amazon Transcribe

Unterstützt ressourcenbasierte Richtlinien Nein

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. 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. Durch das Hinzufügen eines kontoübergreifenden Auftraggebers zu einer ressourcenbasierten Richtlinie ist nur die halbe Vertrauensbeziehung eingerichtet. Wenn sich der Prinzipal und die Ressource unterscheiden AWS-Konten, muss ein IAM-Administrator des vertrauenswürdigen Kontos auch der Prinzipalentsität (Benutzer oder Rolle) die Berechtigung zum Zugriff auf die Ressource erteilen. Sie erteilen Berechtigungen, indem Sie der juristischen Stelle eine identitätsbasierte Richtlinie anfügen. Wenn jedoch eine ressourcenbasierte Richtlinie Zugriff auf einen Prinzipal in demselben Konto gewährt, ist keine zusätzliche identitätsbasierte Richtlinie erforderlich. Weitere Informationen finden Sie unter [Kontenübergreifender Ressourcenzugriff in IAM](#) im IAM-Benutzerhandbuch.

## Richtlinienaktionen für Amazon Transcribe

Unterstützt Richtlinienaktionen	Ja
---------------------------------	----

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Richtlinienaktionen haben normalerweise denselben Namen wie der zugehörige AWS API-Vorgang. Es gibt einige Ausnahmen, z. B. Aktionen, die nur mit Genehmigung durchgeführt werden können und für die es keinen passenden API-Vorgang gibt. Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation 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 -Richtlinien](#)

## Richtlinienressourcen für Amazon Transcribe

Unterstützt Richtlinienressourcen	Ja
-----------------------------------	----

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das bedeutet die Festlegung, welcher Prinzipal Aktionen für welche Ressourcen unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Anweisungen müssen entweder ein `Resource` oder ein `NotResource`-Element enthalten. Als bewährte Methode geben Sie eine Ressource mit dem zugehörigen [Amazon-Ressourcennamen \(ARN\)](#) an. Sie können dies für Aktionen tun, die einen bestimmten Ressourcentyp unterstützen, der als Berechtigungen auf Ressourcenebene bezeichnet wird.

Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, z. B. Auflistungsoperationen, 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 -Richtlinien](#)

## 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 kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Condition` (oder `Condition block`) ermöglicht Ihnen die Angabe der Bedingungen, unter denen eine Anweisung wirksam ist. Das Element `Condition` ist optional. 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.

Wenn Sie mehrere `Condition`-Elemente in einer Anweisung oder mehrere Schlüssel in einem einzelnen `Condition`-Element angeben, wertet AWS diese mittels einer logischen AND-Operation aus. Wenn Sie mehrere Werte für einen einzelnen Bedingungsschlüssel angeben, wertet die Bedingung mithilfe einer logischen OR Operation aus. Alle Bedingungen müssen erfüllt werden, bevor die Berechtigungen der Anweisung gewährt werden.

Sie können auch Platzhaltervariablen verwenden, wenn Sie Bedingungen angeben. Beispielsweise können Sie einem IAM-Benutzer die Berechtigung für den Zugriff auf eine Ressource nur dann gewähren, wenn sie mit dessen IAM-Benutzernamen gekennzeichnet ist. Weitere Informationen finden Sie unter [IAM-Richtlinienelemente: Variablen und Tags](#) im IAM-Benutzerhandbuch.

AWS unterstützt globale Bedingungsschlüssel und dienstspezifische Bedingungsschlüssel. 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 -Richtlinien](#)

## 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

Die attributbasierte Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, bei der Berechtigungen basierend auf Attributen definiert werden. In AWS werden diese Attribute als Tags bezeichnet. Sie können Tags an IAM-Entitäten (Benutzer oder Rollen) und an viele AWS Ressourcen anhängen. Das Markieren von Entitäten und Ressourcen ist der erste Schritt von ABAC. Anschließend entwerfen Sie ABAC-Richtlinien, um Operationen zuzulassen, wenn das Tag des Prinzipals mit dem Tag der Ressource übereinstimmt, auf die sie zugreifen möchten.

ABAC ist in Umgebungen hilfreich, die schnell wachsen, und unterstützt Sie in Situationen, in denen die Richtlinienverwaltung mühsam wird.

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 [Was ist ABAC?](#) 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 [Markieren 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
--	----

Einige funktionieren AWS-Services nicht, wenn Sie sich mit temporären Anmeldeinformationen anmelden. Weitere Informationen, einschließlich Informationen, die mit temporären Anmeldeinformationen AWS-Services [funktionieren AWS-Services](#), finden Sie im [IAM-Benutzerhandbuch unter Diese Option funktioniert mit IAM](#).

Sie verwenden temporäre Anmeldeinformationen, wenn Sie sich mit einer anderen AWS Management Console Methode als einem Benutzernamen und einem Passwort anmelden. Wenn Sie beispielsweise AWS über den Single Sign-On-Link (SSO) Ihres Unternehmens darauf zugreifen, werden bei diesem Vorgang automatisch temporäre Anmeldeinformationen erstellt. Sie erstellen auch automatisch temporäre Anmeldeinformationen, wenn Sie sich als Benutzer bei der Konsole anmelden und dann die Rollen wechseln. Weitere Informationen zum Wechseln von Rollen finden Sie unter [Wechseln zu einer Rolle \(Konsole\)](#) im IAM-Benutzerhandbuch.

Mithilfe der AWS API AWS CLI oder können Sie temporäre Anmeldeinformationen manuell erstellen. Sie können diese temporären Anmeldeinformationen dann für den Zugriff verwenden AWS. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter [Temporäre Sicherheitsanmeldeinformationen in IAM](#).

## Serviceübergreifende Prinzipal-Berechtigungen für Amazon Transcribe

Unterstützt Forward Access Sessions (FAS)	Ja
---	----

Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, kombiniert mit der Anforderung, Anfragen an nachgelagerte Dienste AWS-

Service zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen 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.

## Serviceverknüpfte Rollen für Amazon Transcribe

Unterstützt serviceverknüpfte Rollen	Nein
--------------------------------------	------

Eine serviceverknüpfte Rolle ist eine Art von Servicerolle, die mit einer AWS-Service verknüpft ist. Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Dienstbezogene Rollen werden in Ihrem Dienst angezeigt 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 Service mit



Ein Beispiel für eine Richtlinie zur Übernahme einer Rolle, die zeigt, wie Sie ein verwirrendes Stellvertreterproblem vermeiden können, finden Sie unter [Confused-Deputy-Prävention](#).

## Amazon Transcribe Beispiele für identitätsbasierte -Richtlinien

Benutzer und Rollen haben standardmäßig nicht die Berechtigung, Amazon Transcribe-Ressourcen zu erstellen oder zu ändern. Sie können auch keine Aufgaben über die AWS Management Console, die AWS Command Line Interface (AWS CLI) oder die AWS-API ausführen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter [Erstellen von IAM-Richtlinien](#) im IAM-Benutzerhandbuch.

Einzelheiten zu den von Amazon Transcribe definierten Aktionen und Ressourcentypen, einschließlich des Formats der ARNs für die einzelnen Ressourcentypen, finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für Amazon Transcribe](#) in der Referenz für die Service-Autorisierung.

### Themen

- [Bewährte Methoden für Richtlinien](#)
- [Verwendung der AWS Management Console](#)
- [Erforderliche Berechtigungen für IAM Rollen](#)
- [Für Amazon S3 Verschlüsselungsschlüssel sind Berechtigungen erforderlich](#)
- [Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer](#)
- [AWS KMS Kontextrichtlinie](#)
- [Confused-Deputy-Prävention](#)
- [Anzeigen von Transkriptionsaufträgen anhand von -Tags](#)

## Bewährte Methoden für Richtlinien

Identitätsbasierte 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. Befolgen Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Anleitungen und Empfehlungen:

- Erste Schritte mit AWS-verwaltete Richtlinien und Umstellung auf Berechtigungen mit den geringsten Berechtigungen – Um Ihren Benutzern und Workloads Berechtigungen zu gewähren, verwenden Sie die AWS-verwaltete Richtlinien die Berechtigungen für viele allgemeine Anwendungsfälle gewähren. Sie sind in Ihrem AWS-Konto verfügbar. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie AWS-kundenverwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie unter [AWS-verwaltete Richtlinien](#) oder [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 Service-Aktionen zu gewähren, wenn diese durch ein bestimmtes AWS-Service, wie beispielsweise AWS CloudFormation, verwendet werden. 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 zum IAM Access Analyzer](#) im IAM-Benutzerhandbuch.
- Bedarf einer Multi-Faktor-Authentifizierung (MFA) – Wenn Sie ein Szenario haben, das IAM-Benutzer oder Root-Benutzer in Ihrem AWS-Konto erfordert, 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 [Konfigurieren eines MFA-geschützten API-Zugriffs](#) im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter [Bewährte Methoden für die Sicherheit in IAM](#) im IAM-Benutzerhandbuch.

## Verwendung der AWS Management Console

Um auf die Amazon-Transcribe-Konsole zugreifen zu können, müssen Sie über einen Mindestsatz von Berechtigungen verfügen. Diese Berechtigungen müssen Ihnen das Auflisten und Anzeigen von Details zu den Amazon Transcribe -Ressourcen in Ihrem gestatten 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.

Für Benutzer, die nur Aufrufe an die AWS CLI oder AWS-API durchführen, müssen Sie keine Mindestberechtigungen in der Konsole erteilen. Stattdessen sollten Sie nur Zugriff auf die Aktionen zulassen, die den API-Operation entsprechen, die die Benutzer ausführen möchten.

Um sicherzustellen, dass eine Entität (Benutzer und Rollen) die verwenden kann [AWS Management Console](#), fügen Sie ihnen eine AWS der folgenden verwalteten Richtlinien hinzu.

- `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` dem Bucket-Namen.
- `AmazonTranscribeReadOnlyAccess`: Gewährt Lesezugriff auf Amazon Transcribe Ressourcen, sodass Sie Transkriptionsaufträge und benutzerdefinierte Vokabeln abrufen und auflisten können.

### Note

Sie können die von verwalteten Berechtigungsrichtlinien prüfen, indem Sie sich bei der anmelden in der AWS Management Console und nach dem Richtlinienamen suchen. Eine Suche nach „transcribe“ gibt beide oben aufgeführten Richtlinien (`AmazonTranscribeReadOnly` und `AmazonTranscribeFullAccess`) zurück.

Sie können auch eigene benutzerdefinierte IAM-Richtlinien erstellen, um Berechtigungen für Amazon Transcribe-API-Aktionen zu erteilen. Sie können diese benutzerdefinierten Richtlinien den -Entitäten zuweisen, die diese Berechtigungen benötigen.

## Erforderliche Berechtigungen für IAM Rollen

Wenn Sie eine IAM Rolle zum Aufrufen erstellen Amazon Transcribe, muss diese über die Berechtigung verfügen, auf den Amazon S3 Bucket zuzugreifen. Falls zutreffend, KMS key muss der auch verwendet werden, um den Inhalt des Buckets zu verschlüsseln. In den folgenden Abschnitten finden Sie Beispiele für Richtlinien.

### Richtlinien für Vertrauen

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 die „Vertrauensrichtlinie für Anrufanalysen in Echtzeit“ verwenden müssen, wenn Sie eine Anrufanalyseanfrage in Echtzeit stellen und die Analytik nach dem Anruf aktiviert haben.

### Vertrauenspolitik für Amazon Transcribe

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

```

    }
  ]
}

```

## Vertrauensrichtlinie für Anrufanalysen in Echtzeit

```

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

```

## Amazon S3Input-Bucket-Richtlinie

Die folgende Richtlinie gewährt einer IAM Rolle die Berechtigung, auf Dateien aus dem angegebenen Eingabe-Bucket zuzugreifen.

```

{
  "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 S3Output-Bucket-Richtlinie

Die folgende Richtlinie gewährt einer IAM Rolle die Berechtigung, Dateien in den angegebenen Ausgabe-Bucket zu schreiben.

```

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

```

## Für Amazon S3 Verschlüsselungsschlüssel sind Berechtigungen erforderlich

Wenn Sie einen Bucket verwenden KMS key, um einen Amazon S3 Bucket zu verschlüsseln, nehmen Sie Folgendes in die KMS key Richtlinie auf. Dies ermöglicht einen Amazon Transcribe-Zugriff auf den Inhalt des Buckets. Weitere Informationen zum Erlauben des Zugriffs auf KMS keys finden Sie KMS key im AWS KMS Entwicklerhandbuch unter [Zulassen des Zugriffs auf](#). AWS-Konten

```

{
  "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 enthält Berechtigungen für die Ausführung dieser Aktion auf der Konsole oder für die programmgesteuerte Ausführung über die AWS CLI oder die AWS-API.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ],
      "Resource": ["arn:aws:iam::*:user/${aws:username}"]
    },
    {
      "Sid": "NavigateInConsole",
      "Effect": "Allow",
      "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
      ],
    },
  ],
}

```

```

        "Resource": "*"
    }
]
}

```

## AWS KMS Kontextrichtlinie

Die folgende Richtlinie gewährt der IAM Rolle „ExampleRole“ die Berechtigung, die Operationen AWS KMS Entschlüsseln und Verschlüsseln für dieses spezielle Objekt zu verwenden KMS key. Diese Richtlinie funktioniert nur für Anfragen mit mindestens einem Verschlüsselungskontextpaar, in diesem Fall "color:indigoBlue". Weitere Informationen zum AWS KMS Verschlüsselungskontext finden Sie unter [AWS KMS-Verschlüsselungskontext](#).

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
      },
      "Action": [
        "kms:Decrypt",
        "kms:DescribeKey",
        "kms:Encrypt",
        "kms:GenerateDataKey*",
        "kms:ReEncrypt*"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:EncryptionContext:color": "indigoBlue"
        }
      }
    }
  ]
}

```

## Confused-Deputy-Prävention

Hier ist ein Beispiel für eine Richtlinie zum Annehmen einer Rolle, die zeigt, wie Sie `aws:SourceAccount` mit `aws:SourceArn` und mit Amazon Transcribe den Sie ein

verwirrendes Stellvertreterproblem vermeiden können. Weitere Informationen zur Prävention verwirrter Stellvertreter finden Sie unter [Vermeidung des Problems des verwirrten Stellvertreters \(dienstübergreifend\)](#).

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

## Anzeigen von Transkriptionsaufträgen anhand von -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 können, die das Anzeigen eines Transkriptionsauftrags erlaubt. Die Berechtigung wird jedoch nur erteilt, wenn das Transkription-Job-Tags den Wert des Benutzernamens dieses Benutzers `owner` aufweist. Diese Richtlinie gewährt auch die erforderlichen Berechtigungen, um diese Aktion mit der durchzuführen `AWS Management Console`.

Sie können diese Richtlinie den IAM -Entitäten in Ihrem Konto zuweisen. Wenn eine benannte Rolle `test-role` versucht, einen Transkriptionsjob aufzurufen, muss der Transkriptionsauftrag mit einem Tag versehen sein `owner=test-role` oder `owner=test-role` (bei Bedingungsschlüsselnamen wird nicht zwischen Groß- und Kleinschreibung unterschieden),

andernfalls wird ihnen der Zugriff verweigert. Weitere Informationen finden Sie unter [IAM-JSON-Richtlinienelemente: Bedingung](#) im IAMBenutzerhandbuch.

Weitere Informationen zum Tagging finden Sie unter [Markieren von Ressourcen](#). Amazon Transcribe

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ListTranscriptionJobsInConsole",
      "Effect": "Allow",
      "Action": "transcribe:ListTranscriptionJobs",
      "Resource": "*"
    },
    {
      "Sid": "ViewTranscriptionJobsIfOwner",
      "Effect": "Allow",
      "Action": "transcribe:GetTranscriptionJobs",
      "Resource": "arn:aws:transcribe:*:*:transcription-job/*",
      "Condition": {
        "StringEquals": {"aws:ResourceTag/Owner": "${aws:username}"}
      }
    }
  ]
}
```

## Fehlerbehebung bei 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 autorisiert, eine Aktion in Amazon Transcribe auszuführen.](#)
- [Ich bin nicht berechtigt, iam auszuführen: PassRole](#)
- [Ich möchte Personen außerhalb von mir den Zugriff AWS-Konto auf meine Amazon Transcribe Ressourcen ermöglichen](#)

Ich bin nicht autorisiert, eine Aktion in Amazon Transcribe auszuführen.

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 von mir den Zugriff AWS-Konto auf meine Amazon Transcribe Ressourcen ermöglichen

Sie können eine Rolle erstellen, die Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation für den Zugriff auf Ihre Ressourcen verwenden 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 im IAM-Benutzerhandbuch unter [Kontenübergreifender Ressourcenzugriff in IAM](#).

## Datenschutz in Amazon Transcribe

Das [Modell der geteilten Verantwortung](#) von AWS gilt für den Datenschutz in Amazon Transcribe. Wie in diesem Modell beschrieben, ist AWS verantwortlich für den Schutz der globalen Infrastruktur, in der die gesamte AWS Cloud ausgeführt wird. 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 Blog-Beitrag [AWS-Modell der geteilten Verantwortung und in der DSGVO](#) im AWS-Sicherheitsblog.

Aus Datenschutzgründen empfehlen wir, AWS-Konto-Anmeldeinformationen zu schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einzurichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden zu schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor Authentifizierung (MFA).
- Verwenden Sie SSL/TLS für die Kommunikation mit AWS-Ressourcen. Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit AWS CloudTrail ein.
- Verwenden Sie AWS-Verschlüsselungslösungen zusammen mit allen Standardsicherheitskontrollen in 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 auf AWS über eine Befehlszeilenschnittstelle oder über eine API FIPS 140-2-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter [Federal Information Processing Standard \(FIPS\) 140-2](#).

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 unter Verwendung der Konsole, der API, AWS CLI oder AWS SDKs mit Amazon Transcribe oder anderen AWS-Services arbeiten. 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.

## Datenschutz für den Datenverkehr zwischen Netzwerken

Ein Amazon Virtual Private Cloud (Amazon VPC)-Endpunkt für Amazon Transcribe ist eine logische Entität innerhalb einer VPC, die ausschließlich Verbindungen zu Amazon Transcribe zulässt. Amazon VPC leitet Anfragen an Amazon Transcribe und sendet die entsprechenden Antworten zurück zur

VPC. Weitere Informationen finden Sie unter [AWS PrivateLink-Konzepte](#). Weitere Informationen zum Verwenden von Amazon VPC-Endpunkten mit Amazon Transcribe finden Sie unter [Amazon Transcribe und Schnittstellen-VPC-Endpunkte \(AWS PrivateLink\)](#).

## Datenverschlüsselung

Datenverschlüsselung bezieht sich auf den Schutz von Daten während der Übertragung und im Ruhezustand. Sie können Ihre Daten während der Übertragung zusätzlich Amazon S3 zu standardmäßiger Transport Layer Security (TLS) schützen, indem Sie verwaltete oder KMS keys gespeicherte Schlüssel verwenden.

### Verschlüsselung im Ruhezustand

Amazon Transcribe verwendet den Amazon S3 Standardschlüssel (SSE-S3) für die serverseitige Verschlüsselung von Transkripten in Ihrem Amazon S3 Bucket.

Wenn Sie den [StartTranscriptionJob](#) Vorgang verwenden, können Sie Ihren eigenen Vorgang angeben, KMS key um die Ausgabe eines Transkriptionsauftrags zu verschlüsseln.

Amazon Transcribe verwendet ein mit dem Standardschlüssel verschlüsseltes Amazon EBS-Volumen.

### Verschlüsselung während der Übertragung

Amazon Transcribe verwendet TLS 1.2 mit AWS-Zertifikaten, um Daten während der Übertragung zu verschlüsseln. Dies beinhaltet Streaming-Transkriptionen.

### Schlüsselverwaltung

Amazon Transcribe arbeitet mit KMS keys, um eine verbesserte Verschlüsselung für Ihre Daten bereitzustellen. Mit Amazon S3 können Sie Ihre Eingabemedien verschlüsseln, wenn Sie einen Transkriptionsauftrag erstellen. Die Integration mit AWS KMS ermöglicht die Verschlüsselung der Ausgabe einer [StartTranscriptionJob](#) Anfrage.

Wenn Sie keinen angeben KMS key, wird die Ausgabe des Transkriptionsjobs mit dem Amazon S3 Standardschlüssel (SSE-S3) verschlüsselt.

Weitere Informationen dazu AWS KMS finden Sie im [AWS Key Management Service Entwicklerhandbuch](#).

## Schlüsselverwaltung mit dem AWS Management Console

Um die Ausgabe Ihres Transkriptionsauftrags zu verschlüsseln, können Sie wählen, ob Sie einen KMS key für den AWS-Konto, der die Anfrage stellt, oder einen KMS key von einem anderen verwenden AWS-Konto.

Wenn Sie keinen angeben KMS key, wird die Ausgabe des Transkriptionsjobs mit dem Amazon S3 Standardschlüssel (SSE-S3) verschlüsselt.

Um die Ausgabeverschlüsselung zu aktivieren:

1. Wählen Sie unter Output data (Ausgabedaten) die Option Encryption (Verschlüsselung).



The screenshot shows the 'Encryption' section in the AWS Management Console. It features a radio button labeled 'Encryption Info' which is selected. Below it are two radio button options: 'Use key from current account' (selected) and 'Use key from different account'. Underneath these options is a text input field labeled 'KMS key ID' with a dropdown arrow and the placeholder text 'Choose key'.

2. Wählen Sie aus, ob der KMS key von dem AWS-Konto stammt, den Sie gerade verwenden, oder von einem anderen AWS-Konto. Wenn Sie einen Schlüssel aus dem aktuellen verwenden möchten AWS-Konto, wählen Sie den Schlüssel unter ID (KMS key Schlüssel-ID). 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 die `kms:Encrypt` Berechtigungen für verfügen KMS key. Weitere Informationen finden Sie unter [Erstellen einer Schlüsselrichtlinie](#).

## Schlüsselverwaltung mit der API

Um die Ausgabeverschlüsselung mit der API zu verwenden, müssen Sie angeben, dass Sie den `OutputEncryptionKMSKeyId` Parameter der [StartTranscriptionJobOperation](#) [StartCallAnalyticsJob](#) [StartMedicalTranscriptionJob](#), oder KMS key verwenden.

Wenn Sie einen Schlüssel verwenden, der sich im Strom befindet AWS-Konto, können Sie Ihren KMS key auf eine von vier Arten angeben:

1. Verwenden Sie die KMS key ID selbst. Zum Beispiel `1234abcd-12ab-34cd-56ef-1234567890ab`.

2. Verwenden Sie einen Alias für die KMS key ID. Zum Beispiel `alias/ExampleAlias`.
3. Verwenden Sie den Amazon-Ressourcennamen (ARN) für die KMS key ID. Zum Beispiel `arn:aws:kms:region:account-ID:key/1234abcd-12ab-34cd-56ef-1234567890ab`.
4. Verwenden Sie den ARN für den KMS key Alias. Zum Beispiel `arn:aws:kms:region:account-ID:alias/ExampleAlias`.

Wenn Sie einen Schlüssel verwenden, der sich an einem anderen AWS-Konto als dem aktuellen befindet, können Sie Ihren KMS key auf zwei Arten angeben:

1. Verwenden Sie den ARN für die KMS key ID. Zum Beispiel `arn:aws:kms:region:account-ID:key/1234abcd-12ab-34cd-56ef-1234567890ab`.
2. Verwenden Sie den ARN für den KMS key Alias. Zum Beispiel `arn:aws:kms:region:account-ID:alias/ExampleAlias`.

Beachten Sie, dass die Entität, die die Anfrage stellt, die Berechtigung haben muss, den ausgewählten zu verwenden KMS key.

## AWS KMS-Verschlüsselungskontext

**AWS KMS** Der Verschlüsselungskontext ist eine Abbildung von nicht geheimen Schlüssel/Wert-Paaren im Klartext. Diese Karte stellt zusätzliche authentifizierte Daten dar, sogenannte Verschlüsselungskontextpaare, die eine zusätzliche Sicherheitsebene für Ihre Daten bieten. Amazon Transcribe erfordert einen symmetrischen Verschlüsselungsschlüssel, um die Transkriptionsausgabe in einen vom Kunden angegebenen Amazon S3 Bucket zu verschlüsseln. Weitere Informationen finden Sie unter [Asymmetrische Schlüssel unter AWS KMS](#).

Geben Sie bei der Erstellung Ihrer Verschlüsselungskontextpaare keine vertraulichen Informationen an. Der Verschlüsselungskontext ist nicht geheim — er ist in Ihren CloudTrail Protokollen im Klartext sichtbar (sodass Sie ihn verwenden können, um Ihre kryptografischen Operationen zu identifizieren und zu kategorisieren).

Ihr Verschlüsselungskontextpaar kann Sonderzeichen wie Unterstriche (`_`), Bindestriche (`-`), Schrägstriche (`/`, `\`) und Doppelpunkte (`:`) enthalten.

### Tip

Es kann nützlich sein, die Werte in Ihrem Verschlüsselungskontextpaar den zu verschlüsselnden Daten zuzuordnen. Obwohl dies nicht erforderlich ist, empfehlen wir Ihnen,

nicht sensible Metadaten zu verwenden, die sich auf Ihren verschlüsselten Inhalt beziehen, z. B. Dateinamen, Header-Werte oder unverschlüsselte Datenbankfelder.

Um die Ausgabeverschlüsselung mit der API zu verwenden, legen Sie den `KMSEncryptionContext` Parameter in der [StartTranscriptionJob](#) Operation fest. Um den Verschlüsselungskontext für den Ausgabeverschlüsselungsvorgang bereitzustellen, muss der `OutputEncryptionKMSKeyId` Parameter auf eine symmetrische KMS key ID verweisen.

Sie können [Bedingungsschlüssel](#) mit IAM Richtlinien verwenden AWS KMS, 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 Richtlinie für den Verschlüsselungskontext finden Sie unter [AWS KMS Kontextrichtlinie](#).

Die Verwendung des Verschlüsselungskontextes ist zwar optional, wird aber empfohlen. Weitere Informationen finden Sie unter [Verschlüsselungskontext](#).

## Abmeldung von der Verwendung Ihrer Daten zur Serviceverbesserung

Amazon Transcribe speichert und verwendet standardmäßig Spracheingaben, die verarbeitet wurden, um den Dienst weiterzuentwickeln und Ihr Erlebnis kontinuierlich zu verbessern. Sie können die Verwendung Ihrer Inhalte zur Entwicklung und Verbesserung ablehnen, Amazon Transcribe indem Sie eine AWS Organizations Opt-Out-Richtlinie verwenden. Weitere Informationen zur Opt-Out-Richtlinien zur [Opt-Out-Richtlinien für KI-Services für KI-Services für KI-Services](#) für KI-Services für KI-Services

## Überwachung 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 CloudTrailerfasst 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 EventBridgeist 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

- [Überwachung Amazon Transcribe mit Amazon CloudWatch](#)
- [Überwachung Amazon Transcribe mit AWS CloudTrail](#)
- [Verwenden Amazon EventBridge mit Amazon Transcribe](#)

## Überwachung 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 Verlaufsinfos 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 von 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](#).

## Überwachung 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 Management Console 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 Management Console 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 Management Console, 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 hier:

- [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.

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Mit diesen Informationen können Sie Folgendes bestimmen:

- Gibt an, 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: Einträge in Amazon Transcribe Protokolldateien

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.

```
{
  "Records": [
    {
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "111122223333",
        "arn": "arn:aws:iam:us-west-2:111122223333:user/my-user-name",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "my-user-name"
      },
      "eventTime": "2022-03-07T15:03:45Z",
      "eventSource": "transcribe.amazonaws.com",
      "eventName": "StartTranscriptionJob",
      "awsRegion": "us-west-2",
      "sourceIPAddress": "127.0.0.1",
      "userAgent": "[]",
      "requestParameters": {
        "mediaFormat": "flac",
        "languageCode": "en-US",
        "transcriptionJobName": "my-first-transcription-job",
        "media": {
          "mediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-media-file.flac"
        }
      }
    }
  ]
}
```

```
    },
    "responseElements": {
      "transcriptionJob": {
        "transcriptionJobStatus": "IN_PROGRESS",
        "mediaFormat": "flac",
        "creationTime": "2022-03-07T15:03:44.229000-08:00",
        "transcriptionJobName": "my-first-transcription-job",
        "languageCode": "en-US",
        "media": {
          "mediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-media-file.flac"
        }
      }
    },
    "requestID": "47B8E8D397DCE7A6",
    "eventID": "cdc4b7ed-e171-4cef-975a-ad829d4123e8",
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
  },
  {
    "eventVersion": "1.05",
    "userIdentity": {
      "type": "IAMUser",
      "principalId": "111122223333",
      "arn": "arn:aws:iam:us-west-2:111122223333:user/my-user-name",
      "accountId": "111122223333",
      "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
      "userName": "my-user-name"
    },
    "eventTime": "2022-03-07T15:07:11Z",
    "eventSource": "transcribe.amazonaws.com",
    "eventName": "GetTranscriptionJob",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "[]",
    "requestParameters": {
      "transcriptionJobName": "my-first-transcription-job"
    },
    "responseElements": {
      "transcriptionJob": {
        "settings": {
        },
        "transcriptionJobStatus": "COMPLETED",
        "mediaFormat": "flac",
```

```
    "creationTime": "2022-03-07T15:03:44.229000-08:00",
    "transcriptionJobName": "my-first-transcription-job",
    "languageCode": "en-US",
    "media": {
      "mediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-media-file.flac"
    },
    "transcript": {
      "transcriptFileUri": "s3://DOC-EXAMPLE-BUCKET/my-first-
transcription-job.json"
    }
  },
  "requestID": "BD8798EACDD16751",
  "eventID": "607b9532-1423-41c7-b048-ec2641693c47",
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
]
```

## 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-, [AWS Chatbot](#)-Chat- 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.

## Regeln definieren EventBridge

Um EventBridge Regeln zu definieren, verwenden Sie den [AWS Management Console](#). Wenn Sie eine Regel definieren, verwenden Sie Amazon Transcribe als Servicennamen. 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 Ereignisse

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. z. B. `Transcribe Job State Change`.
- `source`: Die Quelle des Ereignisses. Denn Amazon Transcribe das ist immer `aws.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 lautetFAILED. 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"),
    "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.
- `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://DOC-EXAMPLE-BUCKET/my-output-files/my-redacted-file.JSON",
      "TranscriptFileUri": "s3://DOC-EXAMPLE-BUCKET/my-output-files/my-file.JSON"
    },
    "Media": {
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-output-files/my-redacted-file.WAV",
      "RedactedMediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-output-files/my-redacted-file.WAV"
    }
  }
}
```

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.

## Ereignisse im Wortschatz

Wenn der Status eines [benutzerdefinierten Vokabulars](#) von PENDING zu READY oder wechselt FAILED, 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.

Beachten Sie, dass dieses Ereignis nur für den [CreateVocabulary](#)-API-Vorgang gilt.

```
{
  "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 heruntergeladen AWS Artifact. Weitere Informationen finden Sie unter [Berichte heruntergeladen 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. AWS stellt die folgenden Ressourcen zur Verfügung, die Sie bei der Einhaltung der Vorschriften unterstützen:

- [Schnellstartanleitungen zu Sicherheit und Compliance](#) — In diesen Bereitstellungsleitfäden werden architektonische Überlegungen erörtert und Schritte für die Implementierung von Basisumgebungen beschrieben AWS , bei denen Sicherheit und Compliance im Mittelpunkt stehen.
- [Architecting for HIPAA Security and Compliance on Amazon Web Services](#) — In diesem Whitepaper wird beschrieben, wie Unternehmen HIPAA-fähige Anwendungen erstellen AWS können.

## Note

AWS-Services Nicht alle sind HIPAA-fähig. Weitere Informationen finden Sie in der [Referenz für HIPAA-berechtigte Services](#).

- [AWS Compliance-Ressourcen](#) — Diese Sammlung von Arbeitsmappen und Leitfäden gilt möglicherweise für Ihre Branche und Ihren Standort.
- [AWS Leitfäden zur Einhaltung von Vorschriften für Kunden](#) — Verstehen Sie das Modell der gemeinsamen Verantwortung aus dem Blickwinkel der Einhaltung von Vorschriften. In den Leitfäden werden die bewährten Verfahren zur Sicherung zusammengefasst AWS-Services und die Leitlinien den Sicherheitskontrollen in verschiedenen Frameworks (einschließlich des National Institute of Standards and Technology (NIST), des Payment Card Industry Security Standards Council (PCI) und der International Organization for Standardization (ISO)) zugeordnet.
- [Evaluierung von Ressourcen anhand von Regeln](#) im AWS Config Entwicklerhandbuch — Der AWS Config Service bewertet, wie gut Ihre Ressourcenkonfigurationen den internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.

- [AWS Security Hub](#)— Auf diese AWS-Service Weise erhalten Sie einen umfassenden Überblick über Ihren internen Sicherheitsstatus. AWS Security Hub verwendet Sicherheitskontrollen, um Ihre AWS -Ressourcen zu bewerten und Ihre Einhaltung von Sicherheitsstandards und bewährten Methoden zu überprüfen. Eine Liste der unterstützten Services und Kontrollen finden Sie in der [Security-Hub-Steuerungsreferenz](#).
- [Amazon GuardDuty](#) — Dies AWS-Service erkennt potenzielle Bedrohungen für Ihre Workloads AWS-Konten, Container und Daten, indem es Ihre Umgebung auf verdächtige und böswillige Aktivitäten überwacht. GuardDuty kann Ihnen helfen, verschiedene Compliance-Anforderungen wie PCI DSS zu erfüllen, indem es die in bestimmten Compliance-Frameworks vorgeschriebenen Anforderungen zur Erkennung von Eindringlingen erfüllt.
- [AWS Audit Manager](#)— Auf diese AWS-Service Weise können Sie Ihre AWS Nutzung kontinuierlich überprüfen, um das Risikomanagement und die Einhaltung von Vorschriften und Industriestandards zu vereinfachen.

## Ausfallsicherheit in Amazon Transcribe

Die globale AWS-Infrastruktur ist um AWS-Regionen und Availability Zones herum aufgebaut. AWS-Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, die mit einem Netzwerk mit geringer Latenz, hohem Durchsatz und hoher Redundanz 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 über AWS-Regionen und Availability Zones finden Sie unter [Globale AWS-Infrastruktur](#).

## Sicherheit der Infrastruktur in Amazon Transcribe

Als verwalteter Dienst Amazon Transcribe ist er durch AWS globale Netzwerksicherheit geschützt. Informationen zu AWS-Sicherheitsdiensten und wie AWS die Infrastruktur schützt, finden Sie unter [AWS Cloud-Sicherheit](#). Informationen zum Entwerfen Ihrer AWS-Umgebung anhand der bewährten Methoden für die Infrastruktursicherheit finden Sie unter [Infrastrukturschutz](#) im Security Pillar AWS Well-Architected Framework.

Sie verwenden durch AWS veröffentlichte API-Aufrufe, um über das Netzwerk auf Amazon Transcribe zuzugreifen. Kunden müssen Folgendes unterstützen:

- Transport Layer Security (TLS). Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Verschlüsselungs-Suiten mit Perfect Forward Secrecy (PFS) wie DHE (Ephemeral Diffie-Hellman) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Die meisten modernen Systemen wie Java 7 und höher unterstützen diese Modi.

Außerdem müssen Anforderungen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signiert sein, der einem IAM-Prinzipal zugeordnet ist. Alternativ können Sie mit [AWS Security Token Service](#) (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

## Schwachstellenanalyse und -management in Amazon Transcribe

Konfiguration und IT-Steuererelemente unterliegen der übergreifenden Verantwortlichkeit von AWS und Ihnen, unserem Kunden. Weitere Informationen finden Sie unter [AWS Modell der übergreifenden Verantwortlichkeit](#).

### Amazon Transcribe und Schnittstellen-VPC-Endpunkte (AWS PrivateLink)

Sie können eine private Verbindung zwischen Ihrer VPC und Amazon Transcribe herstellen, indem Sie einen Schnittstellen-VPC-Endpunkt erstellen. Schnittstellen-Endpunkte werden unterstützt von [AWS PrivateLink](#), eine Technologie, mit der Sie privat zugreifen können Amazon Transcribe APIs ohne Internet-Gateway, NAT-Gerät, VPN-Verbindung oder AWS Direct Connect Verbindung. Die Instances in Ihrer VPC benötigen für die Kommunikation mit Amazon Transcribe-APIs keine öffentlichen IP-Adressen. Datenverkehr zwischen Ihrer VPC und Amazon Transcribe verlässt das Amazon-Netzwerk nicht.

Jeder Schnittstellenendpunkt wird durch eine oder mehrere [Elastic Network-Schnittstellen](#) in Ihren Subnetzen dargestellt.

Weitere Informationen finden Sie unter [Schnittstellen-VPC-Endpunkte \(AWS PrivateLink\)](#) im Amazon-VPC-Benutzerhandbuch.

### Überlegungen zu Amazon Transcribe-VPC-Endpunkten

Bevor Sie einen Schnittstellen-VPC-Endpunkt für einrichten Amazon Transcribe, stellen Sie sicher, dass Sie überprüfen [Eigenschaften und Einschränkungen von Schnittstellenendpunkten](#) in der Amazon VPC Benutzerleitfaden.

Amazon Transcribe unterstützt Aufrufe aller seiner API-Aktionen aus der VPC.

## Erstellen eines Schnittstellen-VPC-Endpunkts für Amazon Transcribe

Sie können einen VPC-Endpunkt für den erstellen Amazon Transcribe Dienst unter Verwendung des Amazon VPC AWS Management Console oder AWS CLI. Weitere Informationen finden Sie unter [Erstellen eines Schnittstellenendpunkts](#) im Amazon VPC-Benutzerhandbuch.

Für Batch-Transkriptionen in Amazon Transcribe, erstellen Sie einen VPC-Endpunkt mit dem folgenden Dienstnamen:

- `com.amazonaws.us-west-2.transkribieren`

Für Streaming-Transkriptionen in Amazon Transcribe, erstellen Sie einen VPC-Endpunkt mit dem folgenden Dienstnamen:

- `com.amazonaws.us-west-2.transkribieren und streamen`

Wenn Sie privates DNS für den Endpunkt aktivieren, können Sie API-Anfragen stellen an Amazon Transcribe unter Verwendung seines Standard-DNS-Namens für AWS-Region, zum Beispiel `transcribestreaming.us-east-2.amazonaws.com`.

Weitere Informationen finden Sie unter [Zugriff auf einen Dienst über einen Schnittstellenendpunkt](#) in der Amazon VPC Benutzerleitfaden.

## Erstellen einer VPC-Endpunktrichtlinie für Amazon Transcribe

Sie können Ihrem VPC-Endpunkt eine Endpunktrichtlinie anhängen, 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 Aktionen zur Batch-Transkription

Im Folgenden finden Sie 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": "*"
    }
  ]
}
```

## Gemeinsame Subnetze

Sie können VPC-Endpoints 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 unter [Teilen Sie Ihre VPC mit anderen Konten](#) in der Amazon Virtual Private Cloud Führer.

## Bewährte Methoden für die Sicherheit für Amazon Transcribe

Die folgenden bewährten Methoden sind allgemeine Richtlinien und keine vollständige Sicherheitslösung. Da diese bewährten Methoden für Ihre Umgebung möglicherweise nicht angemessen oder ausreichend sind, sollten Sie sie als hilfreiche Überlegungen und nicht als bindend ansehen.

- Verwenden Sie Datenverschlüsselung, z. B. den AWS KMS Verschlüsselungskontext

AWS KMS Der Verschlüsselungskontext ist eine Abbildung von nicht geheimen Schlüssel/Wert-Paaren im Klartext. Diese Karte stellt zusätzliche authentifizierte Daten dar, sogenannte Verschlüsselungskontextpaare, die eine zusätzliche Sicherheitsebene für Ihre Daten bieten.

Weitere Informationen finden Sie unter [AWS KMS-Verschlüsselungskontext](#).

- Verwenden Sie nach Möglichkeit temporäre Anmeldeinformationen

Verwenden Sie möglichst temporäre Anmeldeinformationen anstelle langfristiger Anmeldeinformationen wie Zugriffsschlüssel. Für Szenarien, in denen Sie IAM Benutzer mit programmatischem Zugriff und langfristigen Anmeldeinformationen benötigen, empfehlen wir, die Zugriffsschlüssel Schlüssel Schlüssel Schlüssel Schlüssel 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, dass Sie IAM Zugriff auf zuletzt verwendete Informationen verwenden, um Zugriffsschlüssel sicher zu rotieren und zu entfernen.

Weitere Informationen finden Sie unter [Rotierende Zugriffsschlüssel](#) und [bewährte Sicherheitsmethoden unter IAM](#).

- Verwenden Sie IAM Rollen für Anwendungen und AWS Dienste, die Amazon Transcribe Zugriff benötigen

Verwenden Sie mithilfe einer IAM Rolle temporäre Anmeldeinformationen für Anwendungen und Services, die Zugriff auf Amazon Transcribe benötigen. Wenn Sie eine Rolle verwenden, müssen Sie keine langfristigen Anmeldeinformationen wie Passwort oder Zugriffsschlüssel an eine Amazon EC2 -Instance oder einen AWS -Service verteilen. IAM Rollen können temporäre

Berechtigungen bereitstellen, die von den Anwendungen beim Anfordern von AWS Ressourcen genutzt werden können.

Weitere Informationen finden Sie unter [IAM Rollen](#) und [Gängige Szenarien für Rollen: Benutzer, Anwendungen und Services](#).

- Verskontrolle

Sie können Tags verwenden, um den Zugriff auf benötigten AWS-Konten. Amazon Transcribe In.-Tags können hinzugefügt werden zu: Transkriptionsaufträgen, benutzerdefinierten Vokabeln, benutzerdefinierten Vokabelfiltern und benutzerdefinierten Sprachmodellen.

Weitere Informationen finden Sie unter [Tag-basierte Zugriffskontrolle](#).

- Verwenden Sie AWS Überwachungstools

Die Überwachung ist ein wichtiger Teil der Aufrechterhaltung von Zuverlässigkeit, Sicherheit, Verfügbarkeit und Performance von Amazon Transcribe und Ihren AWS-Lösungen. Sie können überwatchen Amazon Transcribe mit CloudTrail.

Weitere Informationen finden Sie unter [Überwachung Amazon Transcribe mit AWS CloudTrail](#).

- AWS Config aktivieren

AWS Config kann die Konfigurationen Ihrer AWS Ressourcen beurteilen, prüfen und beurteilen. Mit AWS Config können Sie Änderungen in Konfigurationen und Beziehungen zwischen AWS Ressourcen überprüfen. Sie können auch die Verläufe der Ressourcenkonfiguration analysieren und die generelle Konformität mit den in Ihren internen Richtlinien festgelegten Konfigurationen überprüfen. 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 TranscribeMedical ist ein automatischer Spracherkennungsdienst (ASR) für medizinisches Fachpersonal, das medizinisches Sprechen transkribieren möchte, z. B. vom Arzt diktierte Notizen, Überwachung der Arzneimittelsicherheit, telemedizinische Termine oder Gespräche zwischen Arzt und Patient. Amazon Transcribe Medical ist entweder als Echtzeit-Streaming (per Mikrofon) oder durch Transkription einer hochgeladenen Datei (Batch) verfügbar.

## Important

Amazon TranscribeMedical 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. Für bestimmte Anwendungsfälle sollten die Ergebnisse von entsprechend geschulten menschlichen Gutachtern überprüft und verifiziert werden. Amazon Transcribe Medizinische Transkriptionen sollten in Patientenversorgungsszenarien nur verwendet werden, nachdem sie von geschultem medizinischem Fachpersonal auf Richtigkeit und fundiertes medizinisches Urteilsvermögen überprüft wurden.

Amazon TranscribeMedical arbeitet nach einem Modell der geteilten Verantwortung, wobei Medical für den Schutz der Infrastruktur verantwortlichAWS ist, auf derAmazon Transcribe Medical betrieben wird, und Sie für die Verwaltung Ihrer Daten verantwortlich sind. Weitere Informationen finden Sie unter [Modell der gemeinsamen Verantwortung](#).

Amazon TranscribeMedical ist in US-Englisch (en-US) verfügbar.

Die besten Ergebnisse erzielen Sie, wenn Sie ein verlustfreies Audioformat wie FLAC oder WAV mit PCM-16-Bit-Codierung verwenden. Amazon Transcribe Medicalunterstützt Abtastraten von 16.000 Hz oder höher.

Für die Analyse Ihrer Transkripte können Sie andere verwendenAWS-Services, [Amazon Comprehend Medical](#)z.

## Unterstützte Fachgebiete

Spezialität	Subspezialität	Audioeingang
Kardiologie	none	nur Streaming

Spezialität	Subspezialität	Audioeingang
Neurologie	none	nur Streaming
Onkologie	none	nur Streaming
Primärer Osten	Allgemeinmedizin	Stream-er Stream-er Osten
Primärer Osten	Innere Medizin	Stream-er Stream-er Osten
Primärer Osten	Geburtshilfe und Gynäkologie (OB-GYN)	Stream-er Stream-er Osten
Primärer Osten	Pädiatrie	Stream-er Stream-er Osten
Radiologie	none	nur Streaming
Urologie	none	nur Streaming

## Verfügbarkeit und Kontingente in der Region

Call Analytics wird im Folgenden unterstütztAWS-Regionen:

Region	Art der Transkription
af-south-1 (Kapstadt)	Batch
ap-east-1 (Hongkong)	Batch
ap-northeast-1 (Tokyo)	Stream-er Stream-er Osten
ap-northeast-2 (Seoul)	Stream-er Stream-er Osten
ap-south-1 (Mumbai)	Batch
ap-southeast-1 (Singapur)	Batch
ap-southeast-2 (Sydney)	Stream-er Stream-er Osten
ca-central-1 (Kanada, Zentral)	Stream-er Stream-er Osten

Region	Art der Transkription
eu-central-1 (Frankfurt)	Stream-er Stream-er Osten
eu-north-1 (Stockholm)	Batch
eu-west-1 (Irland)	Stream-er Stream-er Osten
eu-west-2 (Londres)	Stream-er Stream-er Osten
eu-west-3 (Paris)	Batch
me-south-1 (Bahrain)	Batch
sa-east-1 (São Paulo)	Stream-er Stream-er Osten
us-east-1 (Nord-Virginia)	Stream-er Stream-er Osten
us-east-2 (Ohio)	Stream-er Stream-er Osten
us-gov-east-1 (GovCloud, US-Ost)	Stream-er Stream-er Osten
us-gov-west-1 (GovCloud, US-West)	Stream-er Stream-er Osten
us-west-1 (San Francisco)	Batch
us-west-2 (Oregon)	Stream-er Stream-er Osten

Beachten Sie, dass die Regionsunterstützung für [Amazon Transcribe](#) Amazon Transcribe Medical, und [Call Analytics unterschiedlich ist](#).

Die Endpunkte für jede unterstützte Region finden Sie unter [Dienstendpunkte](#) in der AWSAllgemeinen Referenz.

Eine Liste der Kontingente, die sich auf Ihre Transkriptionen beziehen, finden Sie in den [Servicekontingenten](#) in der AWSAllgemeinen Referenz. Einige Kontingente können auf Anfrage geändert werden. Wenn die Spalte „Einstellbar“ den Wert „Ja“ enthält, können Sie eine Erhöhung beantragen. Wählen Sie dazu den bereitgestellten Link aus.

## Medizinische Fachund 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 Patientensuche 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 ist die Mindestanzahl von Anforderungsparametern aufgeführt, 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 TranscribeMedical ermöglicht es Ihnen, alternative Transkriptionen zu erstellen. Weitere Informationen finden Sie unter [Generierung alternativer Transkriptionen](#).

Sie können auch die Lautsprecherpartitionierung aktivieren oder Kanäle in Ihrem Audio identifizieren. Weitere Informationen erhalten Sie unter [Aktivieren der Lautsprecherpartitionierung](#) und [Transkription von Mehrkanal-Audio](#).

## Transkription medizinischer Begriffe und Messungen

Amazon TranscribeMedical kann medizinische Begriffe und Messungen transkribieren. Amazon Transcribe Medizinische Outputs Abkürzungen für gesprochene Begriffe. Zum Beispiel wird „Blutdruck“ als „BP“ transkribiert. Eine Liste der Konventionen, die Amazon Transcribe Medical für medizinische Begriffe und Messungen verwendet, finden Sie in der Tabelle auf dieser Seite. 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.

Im Quellaudio 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	i	Der Patient ist 1,8 m groß.
Fuß	ft	Der Patient ist 6 Fuß groß.
Kilo	kg	Der Patient wiegt 80 kg.
Kilogramm	kg	Der Patient wiegt 80 kg.

Im Quellaudio gesprochener Begriff	In der Ausgabe verwendete Abkürzung	Beispielausgabe
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	Patienten-Blutdruck 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.
postoperativ	POSTOP	Patient kam zur POSTOP-Auswertung.

Im Quellaudio gesprochener Begriff	In der Ausgabe verwendete Abkürzung	Beispielausgabe
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	Patient hatte E/A-Sinusrhythmus
L fünf	L5	Lumbalpunktion wurde zwischen L4 und L5 durchgeführt

## Zahlen transkribieren

Amazon Transcribe in der Medizin werden Ziffern als Zahlen statt als Wörter transkribiert. Beispiel: Die gesprochene Zahl „Eintausendzweihundertzweiundvierzig“ wird als 1242 transkribiert.

Zahlen werden nach den folgenden Regeln transkribiert.

Regel	Beschreibung
Konvertieren von Kardinalzahlen größer als 10 zu Ziffern.	<ul style="list-style-type: none"> <li>„Fünfundfünfzig“ &gt; 55</li> <li>„Einhundert“ &gt; 100</li> <li>„Eintausendundeinunddreißig“ &gt; 1031</li> </ul>

Regel	Beschreibung
	<ul style="list-style-type: none"> <li>• „Einhundertdreißig Millionen vierhundertsechszehntausend siebenhundertneun“ &gt; 123.456.789</li> </ul>
<p>Konvertieren von Kardinalzahlen gefolgt von „Million“ oder „Milliarde“ in Zahlen oder gefolgt von einem Wort, wenn auf „Million“ oder „Milliarde“ keine Zahl folgt.</p>	<ul style="list-style-type: none"> <li>• „hundert Millionen“ &gt; 100 Millionen</li> <li>• „eine Milliarde“ &gt; 1 Milliarde</li> <li>• „zwei Punkt drei Millionen“ &gt; 2,3 Millionen</li> </ul>
<p>Konvertieren von Ordinalzahlen größer als 10 zu Ziffern.</p>	<ul style="list-style-type: none"> <li>• „Der Dreiundvierzigste“ &gt; 43.</li> <li>• „Sechszwanzigste Straße“ &gt; 26. Straße</li> </ul>
<p>Konvertieren von Bruchzahlen in ihr numerisches Format.</p>	<ul style="list-style-type: none"> <li>• „ein Viertel“ &gt; 1/4</li> <li>• „drei Sechzehntel“ &gt; 3/16</li> <li>• „ein Halb“ &gt; 1/2</li> <li>• „ein Hundertstel“ &gt; 1/100</li> </ul>
<p>Konvertieren von Zahlen unter 10 zu Ziffern, wenn mehr als drei hintereinander folgen.</p>	<ul style="list-style-type: none"> <li>• „drei vier fünf“ &gt; 345</li> <li>• „Meine Telefonnummer ist vier zwei fünf fünf fünf fünf eins zwei eins zwei“ &gt; 4255551212</li> </ul>
<p>Dezimalstellen werden durch „Komma“ angegeben</p>	<ul style="list-style-type: none"> <li>• „Dreihundertunddrei Komma Fünf“ &gt; 303,5</li> <li>• „Drei Komma Zwei Drei“ &gt; 3,23</li> <li>• „Null Komma Vier“ &gt; 0,4</li> <li>• „Komma Drei“ &gt; 0,3</li> </ul>
<p>Konvertieren des Wortes „Prozent“ nach einer Zahl in das Prozentzeichen (%).</p>	<ul style="list-style-type: none"> <li>• „Dreiundzwanzig Prozent“ &gt; 23 %</li> <li>• „Dreiundzwanzig Komma Vier Fünf Prozent“ &gt; 23,45 %</li> </ul>
<p>Konvertieren der Wörter „Dollar“, „US-Dollar“, „Australischer Dollar“, „AUD“ oder „USD“ nach einer Zahl in ein Dollarzeichen vor der Zahl.</p>	<ul style="list-style-type: none"> <li>• „Ein Dollar und fünfzehn Cent“ &gt; \$ 1,15</li> <li>• „dreiundzwanzig USD“ &gt; 23 USD</li> <li>• „dreiundzwanzig australische Dollar“ &gt; \$23</li> </ul>

Regel	Beschreibung
Konvertieren der Wörter „Pfund“ oder „Milligramm“ in „lbs“ oder „mg“.	<ul style="list-style-type: none"> <li>• „dreiundzwanzig Pfund“ &gt; 23 £</li> <li>• „fünfundvierzig Milligramm“ &gt; 45 mg</li> </ul>
Konvertieren der Wörter „Rupien“, „Indische Rupien“ oder „INR“ nach einer Zahl in das Rupien-Zeichen (#) vor der Zahl.	<ul style="list-style-type: none"> <li>• „dreiundzwanzig Rupien“ &gt; #23</li> <li>• „fünfzig Rupien dreißig Paise“ &gt; #50,30</li> </ul>
Konvertieren von Zeiten in Zahlen.	<ul style="list-style-type: none"> <li>• „Sieben Uhr morgens Eastern Standard Time“ &gt; 07:00 Eastern Standard Time</li> <li>• „12:30“ &gt; 12:30 Uhr</li> </ul>
Kombinieren mit zwei Ziffern ausgedrückten Jahreszahlen zu vier Ziffern.  Nur gültig für das 20., 21. und 22. Jahrhundert.	<ul style="list-style-type: none"> <li>• „Neunzehnhundertundzweiundsechzig“ &gt; 1962</li> <li>• „das Jahr ist Zwanzig Zwölf“ &gt; das Jahr ist 2012</li> <li>• „Zwanzig Neunzehn“ &gt; 2019</li> <li>• „Einundzwanzig Dreißig“ &gt; 2130</li> </ul>
Umwandeln von Datumsangaben in Zahlen.	<ul style="list-style-type: none"> <li>• „Fünfter Mai Zweitausendundzwölf“ &gt; 5. Mai 2012</li> <li>• „Fünfter Mai 2012“ &gt; 5/5/2012</li> <li>• „Fünfter Mai 2012“ &gt; 5. Mai 2012</li> </ul>
Trennen von Zahlenbereichen durch „bis“	<ul style="list-style-type: none"> <li>• „Dreiundzwanzig bis Siebenundzwanzig“ &gt; 23 bis 27</li> </ul>

## Transkription 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-Transkriptionsauftrag oder einen Echtzeit-Stream verwenden. Mit Batch-Transkriptionsaufträgen können Sie Audiodateien transkribieren. Um sicherzustellen, dass Amazon Transcribe Medical die Transkriptionsergebnisse mit der höchstmöglichen Genauigkeit erstellt, müssen Sie in Ihrem Transkriptionsauftrag oder Stream das medizinische Fachgebiet des Klinikers angeben.

Sie können einen Klinikbesuch in den folgenden medizinischen Fachgebieten transkribieren:

- Kardiologie — nur als Streaming-Transkription verfügbar
- Neurologie — nur als Streaming-Transkription verfügbar
- Onkologie — nur als Streaming-Transkription verfügbar
- Grundversorgung — umfasst die folgenden Arten von Arztpraxen:
  - Allgemeinmedizin
  - Innere Medizin
  - Geburtshilfe und Gynäkologie (OB-GYN)
  - Pädiatrie
- Urologie — nur als Streaming-Transkription verfügbar

Sie können die Transkriptionsgenauigkeit verbessern, indem Sie benutzerdefinierte medizinische Vokabeln verwenden. Informationen zur Funktionsweise benutzerdefinierter medizinischer Vokabeln finden Sie unter [Verbesserung der Transkriptionsgenauigkeit mit maßgeschneiderten medizinischen Vokabeln](#).

Standardmäßig gibt Amazon Transcribe Medical die Transkription mit dem höchsten Konfidenzniveau zurück. Wenn Sie es so konfigurieren möchten, dass alternative Transkriptionen zurückgegeben werden, finden Sie unter [Generierung alternativer Transkriptionen](#).

Hinweise dazu, wie Zahlen und medizinische Messungen in der Transkriptionsausgabe erscheinen, finden Sie unter [Zahlen transkribieren](#) und [Transkription medizinischer Begriffe und Messungen](#).

## Themen

- [Transkription einer Audiodatei eines medizinischen Gesprächs](#)
- [Transkription eines medizinischen Gesprächs in einem Echtzeit-Stream](#)
- [Aktivieren der Lautsprecherpartitionierung](#)
- [Transkription von Mehrkanal-Audio](#)

## Transkription einer Audiodatei eines medizinischen Gesprächs

Verwenden Sie einen Batch-Transkriptionsjob, um Audiodateien von medizinischen Gesprächen zu transkribieren. Sie können dies verwenden, um einen Dialog zwischen Arzt

und Patient zu transkribieren. Sie können einen Batch-Transkriptionsjob entweder in der [StartMedicalTranscriptionJob](#)API oder in der startenAWS Management Console.

Wenn Sie einen medizinischen Transkriptionsjob mit der [StartMedicalTranscriptionJob](#)API starten, geben Sie PRIMARYCARE ihn als Wert des Specialty Parameters an.

## AWS Management Console

Um einen Dialog zwischen Arzt und Patient zu transkribieren (AWS Management Console)

AWS Management Console Um den zum Transkribieren eines Gesprächs zwischen Arzt und Patient zu verwenden, erstellen Sie einen Transkriptionsauftrag und wählen Sie Konversation als Audioeingabetyp.

1. Melden Sie sich an der [AWS Management Console](#) an.
2. Wählen Sie im Navigationsbereich unter Amazon Transcribe Medical die Option Transcription Jobs.
3. Wählen Sie Create job (Auftrag erstellen) aus.
4. Geben Sie auf der Seite „Jobdetails angeben“ unter Auftragseinstellungen Folgendes an.
  - a. Name — der Name des Transkriptionsauftrags.
  - b. Audioeingangstyp — Konversation
5. Geben Sie für die verbleibenden Felder den Amazon S3 Speicherort Ihrer Audiodatei und den Ort an, an dem Sie die Ausgabe Ihres Transkriptionsauftrags speichern möchten.
6. Wählen Sie Next.
7. Wählen Sie Create aus.

## API

Um ein medizinisches Gespräch mithilfe eines Batch-Transkriptionsauftrags (API) zu transkribieren

- Geben Sie für die [StartMedicalTranscriptionJob](#)API Folgendes an.
  - a. Geben Sie für `MedicalTranscriptionJobName` einen eindeutigen Namen in Ihrem AWS-Konto.
  - b. Geben Sie für `LanguageCode` den Sprachcode an, der in Ihrer Audiodatei gesprochenen Sprache entspricht, und geben Sie die Sprache Ihres Vokabelfilters an.

- c. Geben Sie für den `MediaFileUri` Parameter des `Media` Objekts den Namen der Audiodatei an, die Sie transkribieren möchten.
- d. Für `Specialty` geben Sie das medizinische Fachgebiet des Klinikers an, der in der Audiodatei spricht als `PRIMARYCARE`.
- e. Legen Sie für `Type` die Option `CONVERSATION` fest.
- f. Geben Sie für den Amazon S3 Bucket an `OutputBucketName`, in dem die Transkriptionsergebnisse gespeichert werden sollen.

Im Folgenden finden Sie ein Beispiel für eine Anfrage, bei der der verwendet wird, AWS SDK for Python (Boto3) um ein medizinisches Gespräch zwischen einem Arzt des `PRIMARYCARE` Fachgebiets und einem Patienten zu transkribieren.

```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
job_name = "my-first-med-transcription-job"
job_uri = "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-audio-file.flac"
transcribe.start_medical_transcription_job(
    MedicalTranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'DOC-EXAMPLE-BUCKET',
    OutputKey = 'output-files/',
    LanguageCode = 'en-US',
    Specialty = 'PRIMARYCARE',
    Type = 'CONVERSATION'
)

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

Der folgende Beispielcode zeigt die Transkriptionsergebnisse eines Gesprächs zwischen Arzt und Patient.

```
{
  "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://DOC-EXAMPLE-BUCKET/my-input-files/my-audio-file.flac"  
  },  
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",  
  "OutputKey": "my-output-files/",  
  "LanguageCode": "en-US",  
  "Specialty": "PRIMARYCARE",  
  "Type": "CONVERSATION"  
}
```

## Transkription eines medizinischen Gesprächs in einem Echtzeit-Stream

Sie können einen Audiostream eines medizinischen Gesprächs entweder mit HTTP/2 oder mit [WebSocket](#) Protokollen transkribieren. 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
- Primärer Pflege
- Urologie

Jedes medizinische Fachgebiet umfasst viele Arten von Verfahren und Terminen. Kliniker diktieren daher viele verschiedene Arten von Notizen. Verwenden Sie die folgenden Beispiele als Leitfaden, um Ihnen zu helfen, den Wert des `specialty` URI-Parameters der WebSocket-Anfrage oder des `Specialty` Parameters der [StartMedicalStreamTranscription](#) API anzugeben:

- Für elektrophysiologische oder echokardiographische Konsultationen wählen Sie `CARDIOLOGY`.
- Wählen Sie für medizinische onkologische, chirurgische onkologische oder radioonkologische Konsultationen `ONCOLOGY`.
- Wählen Sie für einen Arzt, der einen Patienten, der einen Schlaganfall erlitten hat, entweder einen vorübergehenden ischämischen Anfall oder einen zerebrovaskulären Anfall, konsultiert `NEUROLOGY`.
- Für eine Beratung zum Thema Harninkontinenz wählen Sie `UROLOGY`.
- Wählen Sie für jährliche Untersuchungen oder dringende Pflegebesuche `PRIMARYCARE`.
- Für stationäre Krankenhausbesuche wählen Sie `PRIMARYCARE`.
- Für Konsultationen zu Fruchtbarkeit, Tubenligatur, Einführung einer Spirale oder Abtreibung wählen Sie `PRIMARYCARE`.

## AWS Management Console

Um ein medizinisches Streaming-Gespräch zu transkribieren (AWS Management Console)

Um einen Dialog zwischen Arzt und Patient im Echtzeit-Stream AWS Management Console zu transkribieren, wählen Sie die Option zum Transkribieren eines medizinischen Gesprächs, starten Sie den Stream und beginnen Sie, in das Mikrofon zu sprechen.

1. Melden Sie sich an der [AWS Management Console](#) an.
2. Wählen Sie im Navigationsbereich unter Amazon Transcribe Mediziner die Option Echtzeittranskription.
3. Wählen Sie Konversation.
4. Wählen Sie als medizinisches Fachgebiet das Fachgebiet des Klinikers.

5. Wählen Sie dann Start Streaming (Streamen starten).
6. Sprich in das Mikrofon.

## Transkription eines medizinischen Gesprächs in einem HTTP/2-Stream

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 Eingangs-Audio verwendete Kodierung. Gültige Werte sind pcm, ogg-opus und flac.
- `Specialty`— Die Spezialität des Arztes.
- `Type` – CONVERSATION

Verwenden Sie ein benutzerdefiniertes Vokabular, um die Transkriptionsgenauigkeit bestimmter Begriffe in einem Echtzeit-Stream zu verbessern. Um ein benutzerdefiniertes Vokabular zu aktivieren, setzen Sie den Wert des `VocabularyName` Parameters auf den Namen des benutzerdefinierten Vokabulars, das Sie verwenden möchten. Weitere Informationen finden Sie unter [Verbesserung der Transkriptionsgenauigkeit mit maßgeschneiderten medizinischen Vokabeln](#).

Um die Sprache von verschiedenen Lautsprechern zu beschriften, setzen Sie den `ShowSpeakerLabel` Parameter auf `true`. Weitere Informationen finden Sie unter [Aktivieren der Lautsprecherpartitionierung](#).

Weitere Informationen zum Einrichten eines HTTP/2-Streams zur Transkription eines medizinischen Gesprächs finden Sie unter [Einrichten eines HTTP/2-Streams](#).

## Transkription eines medizinischen Gesprächs in einem WebSocket Stream

Sie können eine WebSocket Anfrage verwenden, um ein medizinisches Gespräch zu transkribieren. Wenn Sie eine WebSocket Anfrage stellen, erstellen Sie eine vorsignierte URI. Diese URI enthält die Informationen, die für die Einrichtung des Audiostreams zwischen Ihrer Anwendung und Amazon Transcribe Medical erforderlich sind. Weitere Informationen zum Erstellen von WebSocket Anfragen finden Sie unter [Einen WebSocket Stream einrichten](#).

Verwenden Sie die folgende Vorlage, um Ihre vorsignierte 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
```

Verwenden Sie ein benutzerdefiniertes Vokabular, um die Transkriptionsgenauigkeit bestimmter Begriffe in einem Echtzeit-Stream zu verbessern. Um ein benutzerdefiniertes Vokabular `vocabulary-name` zu aktivieren, legen Sie den Wert von auf den Namen des benutzerdefinierten Vokabulars fest, das Sie verwenden möchten. Weitere Informationen finden Sie unter [Verbesserung der Transkriptionsgenauigkeit mit maßgeschneiderten medizinischen Vokabeln](#).

Um die Sprache von verschiedenen Lautsprechern zu beschriften, setzen Sie den `show-speaker-label` Parameter auf `true`. Weitere Informationen finden Sie unter [Aktivieren der Lautsprecherpartitionierung](#).

Weitere Informationen zum Erstellen vorsignierter URIs finden Sie unter [Einen WebSocket Stream einrichten](#).

## Aktivieren der Lautsprecherpartitionierung

Verwenden Sie die Lautsprechertastatur, um die Lautsprecherpartitionierung in Amazon Transcribe Medical zu aktivieren. Auf diese Weise können Sie in der Transkriptionsausgabe sehen, was der Patient gesagt hat und was der Kliniker gesagt hat.

Wenn Sie die Sprechertasterfassung aktivieren, kennzeichnet Amazon Transcribe Medical jede Äußerung eines Sprechers mit einer eindeutigen Kennung für jeden Sprecher. Eine Äußerung ist eine

Spracheinheit, die typischerweise durch Stille von anderen Äußerungen getrennt ist. Bei der Batch-Transkription könnte eine Äußerung des Kliniklers ein Etikett von `erhaltenspk_0` und eine Äußerung, für die der Patient ein Etikett erhalten könnte `spk_1`.

Wenn sich eine Äußerung eines Sprechers mit einer Äußerung eines anderen Sprechers überschneidet, ordnet Amazon Transcribe Medical sie in der Transkription nach ihren Startzeiten an. Äußerungen, die in der eingegebenen Audiodatei überlappen, überlappen nicht in der ausgegebenen Transkription.

Sie können die Sprechertagebücher aktivieren, wenn Sie eine Audiodatei mit einem Batch-Transkriptionsauftrag oder in einem Echtzeit-Stream transkribieren.

## Themen

- [Aktivierung der Lautsprecherpartitionierung in Batch-Transkriptionen](#)
- [Aktivieren der Lautsprecherpartitionierung in Echtzeit-Streams](#)

## Aktivierung der Lautsprecherpartitionierung in Batch-Transkriptionen

Sie können die Lautsprecherpartitionierung in einem Batch-Transkriptionsjob entweder über die [StartMedicalTranscriptionJob](#) API oder die `aktivierenAWS` Management Console. Auf diese Weise können Sie den Text in einer Konversation zwischen Arzt und Patient pro Sprecher aufteilen und festlegen, wer in der Transkriptionsausgabe was gesagt hat.

### AWS Management Console

Um in Ihrem Transkriptionsjob die Diarisierung von Sprechern in der AWS Management Console zu aktivieren, aktivieren Sie die Audioidentifikation und anschließend die Lautsprecherpartitionierung.

1. Melden Sie sich an der [AWS Management Console](#) an.
2. Wählen Sie im Navigationsbereich unter Amazon Transcribe Medical die Option `Transcription Jobs`.
3. Wählen Sie `Create job (Auftrag erstellen)` aus.
4. Geben Sie auf der Seite „Stellendetails angeben“ Informationen zu Ihrem Transkriptionsjob ein.
5. Wählen Sie `Next (Weiter)`.
6. Aktivieren Sie die Audioidentifikation.
7. Wählen Sie als Audioidentifizierungstyp die Option `Lautsprecherpartitionierung` aus.

8. Geben Sie für Maximale Anzahl von Lautsprechern die maximale Anzahl von Lautsprechern in Ihrer Audiodatei ein, von denen Sie glauben, dass sie sprechen.
9. Wählen Sie Create (Erstellen) aus.

## API

So aktivieren Sie die Lautsprecherpartitionierung mithilfe eines Batch-Transkriptionsjobs (API)

- Geben Sie für die [StartMedicalTranscriptionJob](#)API Folgendes an.
  - a. Geben Sie für `MedicalTranscriptionJobName` einen Namen an, der in Ihrem einzigartig istAWS-Konto.
  - b. Geben Sie für `LanguageCode` den Sprachcode an, der der in der Audiodatei gesprochenen Sprache entspricht.
  - c. Geben Sie für den `MediaFileUri` Parameter des `Media` Objekts den Namen der Audiodatei an, die Sie transkribieren möchten.
  - d. Für `Specialty` geben Sie in der Audiodatei das medizinische Fachgebiet des behandelnden Arztes an.
  - e. Legen Sie für `Type` die Option `CONVERSATION` fest.
  - f. Geben Sie für den Amazon S3 Bucket an `OutputBucketName`, in dem die Transkriptionsergebnisse gespeichert werden sollen.
  - g. Geben Sie für das `Settings` Objekt Folgendes an.
    - i. `ShowSpeakerLabels` – `true`.
    - ii. `MaxSpeakerLabels`— Eine Ganzzahl zwischen 2 und 10, um die Anzahl der Lautsprecher anzugeben, von denen Sie glauben, dass sie in Ihrem Audio sprechen.

Die folgende Anforderung verwendet die AWS SDK for Python (Boto3) um einen Batch-Transkriptionsjob eines Patientendialogs 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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
transcribe.start_medical_transcription_job(
    MedicalTranscriptionJobName = job_name,
    Media={
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'DOC-EXAMPLE-BUCKET',
    OutputKey = 'my-output-files/',
    LanguageCode = 'en-US',
    Specialty = 'PRIMARYCARE',
    Type = 'CONVERSATION',
    OutputBucketName = 'DOC-EXAMPLE-BUCKET',
    Settings = {'ShowSpeakerLabels': True,
                'MaxSpeakerLabels': 2
               }
)
while True:
    status = transcribe.get_medical_transcription_job(MedicalTranscriptionJobName =
job_name)
    if status['MedicalTranscriptionJob']['TranscriptionJobStatus'] in ['COMPLETED',
'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
print(status)
```

Der folgende Beispielcode zeigt die Transkriptionsergebnisse eines Transkriptionsjobs bei aktivierter Lautsprecherpartitionierung.

```
{
  "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 in der Grundversorgung tätigen Arzt 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://DOC-EXAMPLE-BUCKET/my-input-files/my-audio-file.flac"
  },
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",
  "OutputKey": "my-output-files/",
  "LanguageCode": "en-US",
  "Specialty": "PRIMARYCARE",
  "Type": "CONVERSATION",
  "Settings":{

```

```
        "ShowSpeakerLabels": true,  
        "MaxSpeakerLabels": 2  
    }  
}
```

## Aktivieren der Lautsprecherpartitionierung in Echtzeit-Streams

Verwenden Sie die AWS Management Console oder eine Streaming-Anfrage, um Lautsprecher zu partitionieren und ihre Sprache in einem Echtzeit-Stream zu kennzeichnen. Die Lautsprecherpartitionierung funktioniert am besten für zwei bis fünf Lautsprecher in einem Stream. Amazon Transcribe Medical kann zwar mehr als fünf Lautsprecher 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. Die URI enthält die erforderlichen Informationen, um die bidirektionale Kommunikation zwischen Ihrer Anwendung und Amazon Transcribe Medical herzustellen.

Aktivieren der Lautsprecherpartitionierung in Audiodateien, die in Ihr Mikrofon gesprochen werden (AWS Management Console)

Sie können den verwenden AWS Management Console, 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 Management Console](#) an.
2. Wählen Sie im Navigationsbereich für Amazon Transcribe Medical die Option Echtzeit-Transkription aus.
3. Wählen Sie unter Audioeingabetyp die Art der medizinischen Sprache aus, die Sie transkribieren möchten.
4. Wählen Sie für Zusätzliche Einstellungen die Option Lautsprecherpartitionierung.
5. Wählen Sie Streaming starten, um mit der Transkription Ihres Echtzeit-Audios zu beginnen.
6. Sprich in das Mikrofon.

## Aktivieren der Lautsprecherpartitionierung in einem HTTP/2-Stream

Verwenden Sie die [StartMedicalStreamTranscription](#) API und geben Sie Folgendes an, um die Partitionierung von Lautsprechern in einem HTTP/2-Stream einer medizinischen Konversation zu aktivieren:

- Geben Sie für `LanguageCode` den Sprachcode an, der der Sprache im Stream entspricht. Der gültige Wert lautet `en-US`.
- Geben Sie für `MediaSampleHertz` die Samplerate des Audios an.
- Geben Sie für `Specialty` das medizinische Fachgebiet des Anbieters an.
- `ShowSpeakerLabel` – `true`

Weitere Informationen zum Einrichten eines HTTP/2-Streams zur Transkription eines medizinischen Gesprächs finden Sie unter [Einrichten eines HTTP/2-Streams](#).

## Aktivieren der Lautsprecherpartitionierung in einer WebSocket Anfrage

Um Lautsprecher in WebSocket Streams mit der API zu partitionieren, verwenden Sie das folgende Format, um eine vorsignierte URI zum Starten einer WebSocket Anfrageshow-speaker-label zu erstellen und auf festzulegen `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 einer 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": "XXXXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
    "StartTime": 18.74
}

```

```
]
}
}
```

Amazon TranscribeMedical 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 vorherige Code ist ein verkürztes Beispiel für ein vollständig transkribiertes Sprachsegment. Lautsprecherbeschriftungen werden nur für vollständig transkribierte Segmente angezeigt.

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 `isPartial` Feld. Wenn der Wert `0` ist `false`, beziehen sich die zurückgegebenen Ergebnisse auf ein ganzes Sprachsegment.

## Alternatives

Jedes `Results` Objekt hat ein `Alternatives` Objekt.

## Items

Jedes `Alternatives` Objekt hat sein eigenes `Items` Objekt, das Informationen zu jedem Wort und jedem Satzzeichen in der Transkriptionsausgabe enthält. Wenn Sie die Lautsprecherpartitionierung aktivieren, hat jedes Wort eine `Speaker` Bezeichnung für vollständig transkribierte Sprachsegmente. Amazon Transcribe Medical verwendet dieses Label, um jedem Sprecher im Stream eine eindeutige Ganzzahl zuzuweisen. Der `Type` Parameter mit dem Wert `onspeaker-change` gibt an, dass eine Person aufgehört hat zu sprechen und dass eine andere Person im Begriff ist, zu sprechen.

## Transcript

Jedes `Items`-Objekt enthält ein transkribiertes Sprachsegment als Wert des `Transcript` Felds.

Weitere Informationen zu WebSocket Anfragen finden Sie unter [Einen WebSocket Stream einrichten](#).

## Transkription 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 einzelnen Transkriptionen jedes Kanals zu einem einzigen Transkriptionsausgang.

Verwenden Sie die Kanalidentifikation, um die einzelnen Kanäle in Ihrem Audio zu identifizieren und die Sprache von jedem dieser Kanäle zu transkribieren. Aktivieren Sie dies in Situationen wie einem Anrufer- und Agentenszenario. Verwenden Sie diese Option, um in Aufzeichnungen oder Streams von Kontaktzentren, die die Arzneimittelsicherheit überwachen, einen Anrufer von einem Agenten zu unterscheiden.

Sie können die Kanalidentifikation sowohl für die Batchverarbeitung als auch für das Echtzeit-Streaming aktivieren. In der folgenden Liste wird beschrieben, wie Sie es für jede Methode aktivieren.

- Batch-Transkription — AWS Management Console und [StartMedicalTranscriptionJobAPI](#)
- Streaming-Transkription — WebSocket Streaming und [StartMedicalStreamTranscriptionAPI](#)

## 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 ein Satzzeichen. Jedes Wort hat eine Start- und eine Endzeit. Wenn eine Person auf einem Kanal über eine Person auf einem separaten 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 Kontingenterhöhung beantragen, wenn Sie Dateien mit mehr als zwei Kanälen transkribieren müssen. Informationen zur Beantragung einer Kontingenterhöhung finden Sie unter [AWS-ServiceKontingente](#).

Verwenden Sie die API AWS Management Console oder die [StartMedicalTranscriptionJobAPI](#), um Mehrkanal-Audio in einem Batch-Transkriptionsjob zu transkribieren.

## AWS Management Console

AWS Management Console 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 Management Console.

1. Melden Sie sich an der [AWS Management Console](#) an.
2. Wählen Sie im Navigationsbereich unter Amazon Transcribe Medical die Option Transcription Jobs.
3. Wählen Sie Create job (Auftrag erstellen) aus.
4. Geben Sie auf der Seite „Stellendetails angeben“ Informationen zu Ihrem Transkriptionsjob ein.
5. Wählen Sie Next (Weiter).
6. Aktivieren Sie die Audioidentifikation.
7. Wählen Sie als Audioidentifizierungstyp die Option Kanalidentifikation aus.
8. Wählen Sie Create (Erstellen) aus.

## API

Um eine Mehrkanal-Audiodatei (API) zu transkribieren

- Geben Sie für die [StartMedicalTranscriptionJob](#) API Folgendes an.
  - a. Geben Sie für `TranscriptionJobName` einen eindeutigen Namen für Ihren an AWS-Konto.
  - b. Geben Sie für `LanguageCode` den Sprachcode an, der der in der Audiodatei gesprochenen Sprache entspricht. Der gültige Wert lautet `en-US`.
  - c. Geben Sie für den `MediaFileUri` Parameter des `Media` Objekts den Namen der Mediendatei an, die Sie transkribieren möchten.
  - d. Stellen Sie für das `Settings` Objekt `ChannelIdentification` auf `true`.

Im Folgenden finden Sie ein Beispiel für eine Anfrage mit dem AWS SDK for Python (Boto3).

```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
job_name = "my-first-transcription-job"
```

```
job_name = "my-first-med-transcription-job"
job_uri = "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
transcribe.start_medical_transcription_job(
    MedicalTranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'DOC-EXAMPLE-BUCKET',
    OutputKey = 'output-files/',
    LanguageCode = 'en-US',
    Specialty = 'PRIMARYCARE',
    Type = 'CONVERSATION',
    Settings = {
        'ChannelIdentification': True
    }
)
while True:
    status = transcribe.get_transcription_job(MedicalTranscriptionJobName = job_name)
    if status['MedicalTranscriptionJob']['TranscriptionJobStatus'] in ['COMPLETED',
'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
print(status)
```

## AWS CLI

Um eine Mehrkanal-Audiodatei mit einem Batch-Transkriptionsjob 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
```

Das Folgende ist der Code von `example-start-command.json`.

```
{
    "MedicalTranscriptionJobName": "my-first-med-transcription-job",
```

```

    "Media": {
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-audio-
file.flac"
    },
    "OutputBucketName": "DOC-EXAMPLE-BUCKET",
    "OutputKey": "my-output-files/",
    "LanguageCode": "en-US",
    "Specialty": "PRIMARYCARE",
    "Type": "CONVERSATION",

    "Settings":{
      "ChannelIdentification": true
    }
  }
}

```

Der folgende Code zeigt die Transkriptionsausgabe für eine Audiodatei, die eine Konversation 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

Mit der [StartMedicalStreamTranscription](#)API können Sie Audio von separaten Kanälen entweder in HTTP/2 oder in WebSocket Streams transkribieren.

Standardmäßig können Sie Streams mit zwei Kanälen transkribieren. Sie können eine Kontingenterhöhung beantragen, wenn Sie Streams mit mehr als zwei Kanälen transkribieren müssen. Informationen zur Beantragung einer Kontingenterhöhung finden Sie unter [AWSServicekontingente](#).

### Transkription von Mehrkanal-Audio in einem HTTP/2-Stream

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 Streaming-Audio.

Weitere Informationen zum Einrichten eines HTTP/2-Streams zur Transkription eines medizinischen Gesprächs finden Sie unter [Einrichten eines HTTP/2-Streams](#).

### Transkribieren von Mehrkanal-Audio in einem WebSocket Stream

Verwenden Sie das folgende Format, um Lautsprecher in WebSocket Streams zu partitionieren, um eine vorsignierte URI zu erstellen und eine WebSocket Anfrage zu starten. Geben Sie `enable-channel-identification` als `true` und die Anzahl der Kanäle in Ihrem Stream als `number-of-channels`. Eine vorsignierte URI enthält die Informationen, die für die Einrichtung der bidirektionalen Kommunikation zwischen Ihrer Anwendung und Amazon Transcribe Medical erforderlich sind.

```
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-Vorgängen gemeinsam sind, sind im Abschnitt [Allgemeine Parameter](#) aufgeführt.

Weitere Informationen zu WebSocket Anfragen finden Sie unter [Einen WebSocket Stream einrichten](#).

### Mehrkanal-Streaming-Ausgang

Die Ausgabe einer Streaming-Transkription ist für HTTP/2 und WebSocket Anfragen dieselbe. 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 eine `channelId` Flagge, die angibt, zu welchem Kanal die Sprache gehört.

## Ein medizinisches Diktat transkribieren

Sie können Amazon Transcribe Medical verwenden, um vom Arzt diktierte medizinische Notizen zu transkribieren, indem Sie entweder einen Batch-Transkriptionsauftrag oder einen Echtzeit-Stream verwenden. Mit Batch-Transkriptionsaufträgen können Sie Audiodateien transkribieren. In Ihrem Transkriptionsauftrag oder Stream geben Sie das medizinische Fachgebiet des Klinikers an, um sicherzustellen, dass Amazon Transcribe Medical die Transkriptionsergebnisse mit der höchstmöglichen Genauigkeit erstellt.

Sie können ein medizinisches Diktat in den folgenden Fachgebieten transkribieren:

- Kardiologie — nur als Streaming-Transkription verfügbar
- Neurologie — nur als Streaming-Transkription verfügbar
- Onkologie — nur als Streaming-Transkription verfügbar
- Grundversorgung — umfasst die folgenden Arten von Arztpraxen:
  - Allgemeinmedizin
  - Innere Medizin
  - Geburtshilfe und Gynäkologie (OB-GYN)
  - Pädiatrie
- Radiologie — nur als Streaming-Transkription verfügbar
- Urologie — nur als Streaming-Transkription verfügbar

Sie können die Transkriptionsgenauigkeit verbessern, indem Sie benutzerdefinierte Vokabeln verwenden. Informationen zur Funktionsweise benutzerdefinierter medizinischer Vokabeln finden Sie unter [Verbesserung der Transkriptionsgenauigkeit mit maßgeschneiderten medizinischen Vokabeln](#).

Standardmäßig gibt Amazon Transcribe Medical die Transkription mit dem höchsten Konfidenzniveau zurück. Wenn Sie es so konfigurieren möchten, dass alternative Transkriptionen zurückgegeben werden, finden Sie unter [Generierung alternativer Transkriptionen](#).

Hinweise dazu, wie Zahlen und medizinische Messungen in der Transkriptionsausgabe erscheinen, finden Sie unter [Zahlen transkribieren](#) und [Transkription medizinischer Begriffe und Messungen](#).

Themen

- [Transkription einer Audiodatei eines medizinischen Diktats](#)
- [Transkription eines medizinischen Diktats in einem Echtzeit-Stream](#)

## Transkription einer Audiodatei eines medizinischen Diktats

Verwenden Sie einen Batch-Transkriptionsjob, um Audiodateien von medizinischen Gesprächen zu transkribieren. Sie können dies verwenden, um einen Dialog zwischen Arzt und Patient zu transkribieren. Sie können einen Batch-Transkriptionsjob entweder in der [StartMedicalTranscriptionJob](#) API oder in der `start` AWS Management Console.

Wenn Sie einen medizinischen Transkriptionsjob mit der [StartMedicalTranscriptionJob](#)API starten, geben Sie `PRIMARYCARE` ihn als Wert des `Specialty` Parameters an.

## AWS Management Console

Um einen Dialog zwischen Arzt und Patient zu transkribieren (AWS Management Console)

AWS Management Console Um den zum Transkribieren eines Gesprächs zwischen Arzt und Patient zu verwenden, erstellen Sie einen Transkriptionsauftrag und wählen Sie Konversation als Audioeingabetyp.

1. Melden Sie sich an der [AWS Management Console](#) an.
2. Wählen Sie im Navigationsbereich unter Amazon Transcribe Mediziner Knoten aus.
3. Wählen Sie Create job (Auftrag erstellen) aus.
4. Geben Sie auf der Seite „Jobdetails angeben“ unter Auftragseinstellungen Folgendes an.
  - a. Name — der Name des Transkriptionsauftrags.
  - b. Audioeingangstyp — Diktat
5. Geben Sie für die verbleibenden Felder den Amazon S3 Speicherort Ihrer Audiodatei und den Ort an, an dem Sie die Ausgabe Ihres Transkriptionsauftrags speichern möchten.
6. Wählen Sie Next.
7. Wählen Sie Create aus.

## API

Um ein medizinisches Gespräch mithilfe eines Batch-Transkriptionsauftrags (API) zu transkribieren

- Geben Sie für die [StartMedicalTranscriptionJob](#)API Folgendes an.
  - a. Geben Sie für `MedicalTranscriptionJobName` einen eindeutigen Namen in Ihrem AWS-Konto.
  - b. Geben Sie für `LanguageCode` den Sprachcode an, der der in Ihrer Audiodatei gesprochenen Sprache entspricht, und geben Sie die Sprache Ihres Vokabelfilters an.
  - c. Geben Sie im `MediaFileUri` Parameter des `Media` Objekts den Namen der Audiodatei an, die Sie transkribieren möchten.
  - d. Für `Specialty` geben Sie in der Audiodatei das medizinische Fachgebiet des behandelnden Arztes an.

- e. Legen Sie für Type die Option DICTATION fest.
- f. Geben Sie für den Amazon S3 Bucket an `OutputBucketName`, in dem die Transkriptionsergebnisse gespeichert werden sollen.

Im Folgenden finden Sie ein Beispiel für eine Anfrage, bei der das verwendet wird, AWS SDK for Python (Boto3) um ein medizinisches Diktat eines Klinikers des PRIMARYCARE Fachgebiets zu transkribieren.

```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe')
job_name = "my-first-med-transcription-job"
job_uri = "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-audio-file.flac"
transcribe.start_medical_transcription_job(
    MedicalTranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'DOC-EXAMPLE-BUCKET',
    OutputKey = 'my-output-files/',
    LanguageCode = 'en-US',
    Specialty = 'PRIMARYCARE',
    Type = 'DICTATION'
)
while True:
    status = transcribe.get_medical_transcription_job(MedicalTranscriptionJobName =
job_name)
    if status['MedicalTranscriptionJob']['TranscriptionJobStatus'] in ['COMPLETED',
'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
print(status)
```

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-Transkriptionsjob 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://DOC-EXAMPLE-BUCKET/my-input-files/my-audio-file.flac"  
  },  
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",  
  "OutputKey": "my-output-files/",  
  "LanguageCode": "en-US",  
  "Specialty": "PRIMARYCARE",  
  "Type": "DICTATION"  
}
```

## Transkription eines medizinischen Diktats in einem Echtzeit-Stream

Verwenden Sie einen WebSocket Stream, um ein medizinisches Diktat als Audiostream zu transkribieren. Sie können das auch verwenden AWS Management Console, um Sprache, die Sie oder andere sprechen, direkt in ein Mikrofon zu transkribieren.

Für ein HTTP/2 oder einen WebSocket Stream können Sie Audio in den folgenden medizinischen Fachgebieten transkribieren:

- Kardiologie
- Onkologie
- Neurologie
- Primärer Knoten
- Radiologie
- Urologie

Jedes medizinische Fachgebiet umfasst viele Arten von Verfahren und Terminen. Kliniker 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 ein Diktat nach einer Elektrophysiologie oder einem Echokardiogramm wählen Sie `CARDIOLOGY`.
- Für ein Diktat nach einem chirurgischen onkologischen oder radioonkologischen Eingriff wählen Sie `ONCOLOGY`.
- Für einen Arzt, der Notizen diktiert, die auf die Diagnose einer Enzephalitis hinweisen, wählen Sie `NEUROLOGY`.
- Für ein Diktat von Verfahrenshinweisen zum Zerschneiden eines Blasensteins wählen Sie `UROLOGY`.
- Für ein Diktat von Arztnotizen nach einer internistischen Beratung wählen Sie `PRIMARYCARE`.
- Für ein Diktat eines Arztes, der die Ergebnisse eines CT-Scans, PET-Scans, MRT oder Röntgenbildes mitteilt, wählen Sie `RADIOLOGY`.
- Für ein Diktat von Arztnotizen nach einer gynäkologischen Beratung wählen Sie `PRIMARYCARE`.

Verwenden Sie ein benutzerdefiniertes Vokabular, um die Transkriptionsgenauigkeit bestimmter Begriffe in einem Echtzeit-Stream zu verbessern. Um ein benutzerdefiniertes Vokabular zu aktivieren, legen Sie den Wert von `vocabulary-name` auf den Namen des benutzerdefinierten Vokabulars fest, das Sie verwenden möchten.

Transkribieren eines in Ihr Mikrofon gesprochenen Diktats mit dem AWS Management Console

**AWS Management Console** Um das Streaming-Audio eines medizinischen Diktats 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 Management Console)

1. Melden Sie sich an der [AWS Management Console](#) an.
2. Wählen Sie im Navigationsbereich unter Amazon Transcribe Medizinischer Knoten aus.
3. Wählen Sie Diktat.
4. Wählen Sie als medizinisches Fachgebiet das medizinische Fachgebiet des Klinikers aus, der im Stream spricht.
5. Wählen Sie dann Start Streaming (Streamen starten).

## 6. Sprich in das Mikrofon.

### Transkription eines Diktats in einem HTTP/2-Stream

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 Eingangs-Audio verwendete Kodierung. Gültige Werte sind `pcm`, `ogg-opus` und `flac`.
- `Specialty`— Die Spezialisierung des Mediziners.
- `Type` – `DICTATION`

Weitere Informationen zum Einrichten eines HTTP/2-Streams zur Transkription eines medizinischen Diktats finden Sie unter [Einrichten eines HTTP/2-Streams](#).

### 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 Amazon Transcribe Medical erforderlich sind. Weitere Informationen zum Erstellen von WebSocket Anfragen finden Sie unter [Einen WebSocket Stream einrichten](#).

Verwenden Sie die folgende Vorlage, um Ihre vorsignierte 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 zum Erstellen vorsignierter URIs finden Sie unter [Einen WebSocket Stream einrichten](#).

## Verbesserung der Transkriptionsgenauigkeit mit maßgeschneiderten medizinischen Vokabeln

Um die Genauigkeit der Transkription im Amazon Transcribe medizinischen Bereich zu verbessern, sollten Sie ein oder mehrere benutzerdefinierte medizinische Vokabeln erstellen und verwenden. 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 Amazon Transcribe Medical verwenden. Geben Sie keine vertraulichen Informationen, persönlichen Daten (PII) oder geschützten Gesundheitsinformationen (PHI) in ein benutzerdefiniertes Vokabular ein.

Die besten Ergebnisse erzielen Sie, wenn Sie separate kleine benutzerdefinierte Vokabeln erstellen, die jeweils beim Transkribieren einer bestimmten Audioaufnahme helfen. Sie erhalten eine größere Verbesserung der Transkriptionsgenauigkeit, als wenn Sie ein großes benutzerdefiniertes Vokabular für all Ihre Aufnahmen erstellen würden.

Standardmäßig können Sie bis zu 100 benutzerdefinierte Vokabulare in Ihrem AWS-Konto. Ein benutzerdefiniertes Vokabular darf eine Größe von 50 KB nicht überschreiten. Informationen zur Beantragung einer Erhöhung der Anzahl benutzerdefinierter Vokabeln, die Sie in Ihrem System haben können AWS-Konto, finden Sie unter [AWS Servicekontingente](#).

Benutzerdefinierte Vokabeln sind in US-Englisch (en-US) verfügbar.

### Themen

- [Erstellen einer Textdatei für Ihr medizinisches benutzerdefiniertes Vokabular](#)
- [Verwenden einer Textdatei, um ein medizinisches benutzerdefiniertes Vokabular zu erstellen](#)
- [Transkription einer Audiodatei mit einem medizinischen benutzerdefinierten Vokabular](#)

- [Transkription eines Echtzeit-Streams mit einem medizinischen benutzerdefinierten Vokabular](#)
- [Zeichensatz für Amazon Transcribe Medical](#)

## Erstellen einer Textdatei für Ihr medizinisches benutzerdefiniertes Vokabular

Um ein benutzerdefiniertes Vokabular zu erstellen, erstellen Sie eine Textdatei im UTF-8-Format. In dieser Datei erstellen Sie eine Tabelle mit vier Spalten, wobei jede Spalte ein Feld angibt. Jedes Feld zeigt Amazon Transcribe Medical entweder, wie die domänenspezifischen Begriffe ausgesprochen werden oder wie diese Begriffe in Ihren Transkriptionen angezeigt werden. Sie speichern die Textdatei, die diese Felder enthält, in einem Amazon S3 Bucket.

### Grundlegendes zum Formatieren Ihrer Textdatei

Um ein medizinisches benutzerdefiniertes Vokabular zu erstellen, geben Sie die Spaltennamen als Kopfzeile ein. Sie geben die Werte für jede Spalte unter der Kopfzeile 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

Beim Erstellen eines benutzerdefinierten Vokabulars sollten Sie:

- Trennen Sie jede Spalte durch ein einzelnes Tabulatorzeichen. Amazon Transcribe gibt eine Fehlermeldung, wenn Sie versuchen, die Spalten durch
- Stellen Sie sicher, dass nach jedem Wert in einer Spalte keine Leerzeichen oder Leerzeichen am Ende stehen.

Stellen Sie sicher, dass die Werte, die Sie für jede Spalte eingeben,:

- Habe weniger als 256 Zeichen, einschließlich Bindestriche
- Verwenden Sie nur Zeichen aus dem zulässigen Zeichensatz, siehe [Zeichensatz für Amazon Transcribe Medical](#).

## Werte für die Spalten der Tabelle eingeben

Die folgenden Informationen zeigen Ihnen, wie Sie Werte für die vier Spalten der Tabelle angeben:

- **Phrase**— Das Wort oder die 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 einer Phrase 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 eine Phrase in kleinere Segmente unterteilen und für jedes Segment eine Aussprache angeben, indem Sie die Standardorthografie der Sprache verwenden, 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-ty-with-sub-cor-ti-cul-in-farcts-and-lewk-o-en-ce-phul-ah-pu-ty**. 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) — Definiert, wie das Wort oder der Ausdruck 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 in die Spalte DisplayAs beliebige UTF-8-Zeichen eingeben.

Sie können Leerzeichen nur für die Werte in den DisplayAs Spalten IPA und einschließen.

Um die Textdatei Ihres benutzerdefinierten Wortschatzes zu erstellen, platzieren Sie jedes Wort oder jede Phrase in Ihrer Textdatei in einer separaten 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 demselben Bucket,

inAWS-Region dem SieAmazon Transcribe Medical verwenden, um Ihr benutzerdefiniertes Vokabular zu erstellen.

Wenn Sie Ihre Textdatei in Windows bearbeiten, stellen Sie sicher, dass Ihre Datei im LF-Format und nicht im CRLF-Format vorliegt. Andernfalls können Sie Ihr benutzerdefiniertes Vokabular nicht erstellen. Einige Texteditoren ermöglichen es Ihnen, die Formatierung mit den Befehlen Suchen und Ersetzen zu ändern.

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-ular-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-ular-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   ɛ	A.L.L.
	ay-tree-o-ven-trick-ular-node-al-re-entr-ant-tack-ih-card-ia		atrioventricular-nodal-reentrant-tachycardia

## Verwenden einer Textdatei, um ein medizinisches benutzerdefiniertes Vokabular zu erstellen

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 Sie können mithilfe der [CreateMedicalVocabulary](#) API oder der Amazon Transcribe Medical Console ein benutzerdefiniertes Vokabular erstellen.

### AWS Management Console

Um den zu verwenden, AWS Management Console um ein benutzerdefiniertes Vokabular zu erstellen, geben Sie die Amazon S3 URI der Textdatei an, die Ihre Wörter oder Ausdrücke enthält.

1. Melden Sie sich an der [AWS Management Console](#) an.
2. Wählen Sie im Navigationsbereich unter Amazon Transcribe Medizinisch
3. Wählen Sie für Name unter Vokabeleinstellungen einen Namen für Ihr benutzerdefiniertes Vokabular aus.
4. Geben Sie den Speicherort Ihrer Audio- oder Videodatei in Amazon S3 an:
  - Geben Sie für den Speicherort der Vokabel-Eingabedatei auf S3 unter Vokabeleinstellungen die Amazon S3 URI an, die die Textdatei identifiziert, die Sie zum Erstellen Ihres benutzerdefinierten Vokabels verwenden werden.
  - Wählen Sie für den Speicherort der Vokabel-Eingabedatei in S3 die Option S3 durchsuchen aus, um nach der Textdatei zu suchen und sie auszuwählen.
5. Wählen Sie „Wortschatz erstellen“.

Den Verarbeitungsstatus Ihres benutzerdefinierten Vokabulars können Sie in der [AWS Management Console](#) sehen.

### API

Um ein medizinisches benutzerdefiniertes Vokabular (API) zu erstellen

- Geben Sie für die [StartTranscriptionJob](#) API Folgendes an.
  - a. Legen Sie für LanguageCode die Option en-US fest.
  - b. Geben Sie für VocabularyFileUri den Amazon S3 Speicherort der Textdatei an, die Sie verwenden, um Ihr benutzerdefiniertes Vokabular zu definieren.

- c. Geben Sie für `VocabularyName` einen Namen für Ihr benutzerdefiniertes Vokabular an. Der von Ihnen angegebene Name muss innerhalb Ihres eindeutig sein AWS-Konto.

Verwenden Sie die [GetMedicalVocabulary](#) API, um den Verarbeitungsstatus Ihres benutzerdefinierten Vokabulars zu sehen.

Das folgende Beispiel zeigt die Berechtigung AWS SDK for Python (Boto3) zum Erstellen eines benutzerdefinierten Vokabulars.

```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
vocab_name = "my-first-vocabulary"
response = transcribe.create_medical_vocabulary(
    VocabularyName = job_name,
    VocabularyFileUri = 's3://DOC-EXAMPLE-BUCKET/my-vocabularies/my-vocabulary-
table.txt'
    LanguageCode = 'en-US',
)

while True:
    status = transcribe.get_medical_vocabulary(VocabularyName = vocab_name)
    if status['VocabularyState'] in ['READY', 'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
print(status)
```

## AWS CLI

Um die Lautsprecherpartitionierung in einem Batch-Transkriptionsjob zu aktivieren (AWS CLI)

- Führen Sie folgenden Code aus.

```
aws transcribe create-medical-vocabulary \
--vocabulary-name my-first-vocabulary \
--vocabulary-file-uri s3://DOC-EXAMPLE-BUCKET/my-vocabularies/my-vocabulary-
file.txt \
```

```
--language-code en-US
```

## Transkription einer Audiodatei mit einem medizinischen benutzerdefinierten Vokabular

Verwenden Sie das [StartMedicalTranscriptionJob](#) oder die AWS Management Console, um einen Transkriptionsjob zu starten, bei dem ein benutzerdefiniertes Vokabular verwendet wird, um die Transkriptionsgenauigkeit zu verbessern.

### AWS Management Console

1. Melden Sie sich an der [AWS Management Console](#) an.
2. Wählen Sie im Navigationsbereich unter Amazon Transcribe Medical die Option Transcription Jobs.
3. Wählen Sie Create job (Auftrag erstellen) aus.
4. Geben Sie auf der Seite „Stellendetails angeben“ Informationen zu Ihrem Transkriptionsjob ein.
5. Wählen Sie Next (Weiter).
6. Aktivieren Sie unter Anpassung die Option Benutzerdefiniertes Vokabular.
7. Wählen Sie unter Vokabelauswahl ein benutzerdefiniertes Vokabular aus.
8. Wählen Sie Create (Erstellen) aus.

### API

So aktivieren Sie die Lautsprecherpartitionierung in einer Audiodatei mithilfe eines Batch-Transkriptionsauftrags (API)

- Geben Sie für die [StartMedicalTranscriptionJob](#) API Folgendes an.
  - a. Geben Sie für `MedicalTranscriptionJobName` einen Namen an, der in Ihrem einzigartig ist AWS-Konto.
  - b. Geben Sie für `LanguageCode` den Sprachcode an, der der in Ihrer Audiodatei gesprochenen Sprache entspricht, und geben Sie die Sprache Ihres Vokabelfilters an.
  - c. Geben Sie für den `MediaFileUri` Parameter des `Media` Objekts den Namen der Audiodatei an, die Sie transkribieren möchten.

- d. Für `Specialty` geben Sie in der Audiodatei das medizinische Fachgebiet des behandelnden Arztes an.
- e. Geben Sie für `anType`, ob es sich bei der Audiodatei um eine Konversation oder ein Diktat handelt.
- f. Geben Sie für den Amazon S3 Bucket `anOutputBucketName`, in dem die Transkriptionsergebnisse gespeichert werden sollen.
- g. Geben Sie für das `Settings` Objekt Folgendes an.
  - `VocabularyName`— der Name Ihres benutzerdefinierten Wortschatzes.

Die folgende Anfrage verwendet den AWS SDK for Python (Boto3), um einen Batch-Transkriptionsjob mit einem benutzerdefinierten Vokabular 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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
transcribe.start_medical_transcription_job(
    MedicalTranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'DOC-EXAMPLE-BUCKET',
    OutputKey = 'my-output-files/',
    LanguageCode = 'en-US',
    Specialty = 'PRIMARYCARE',
    Type = 'CONVERSATION',
    Settings = {
        'VocabularyName': 'example-med-custom-vocab'
    }
)

while True:
    status = transcribe.get_medical_transcription_job(MedicalTranscriptionJobName =
job_name)
    if status['MedicalTranscriptionJob']['TranscriptionJobStatus'] in ['COMPLETED',
'FAILED']:
```

```
        break
    print("Not ready yet...")
    time.sleep(5)
print(status)
```

## Transkription eines Echtzeit-Streams mit einem medizinischen benutzerdefinierten Vokabular

Um die Transkriptionsgenauigkeit in einem Echtzeit-Stream zu verbessern, können Sie ein benutzerdefiniertes Vokabular verwenden, das entweder HTTP/2 oder WebSocket Streams verwendet. Verwenden Sie die [StartMedicalStreamTranscriptionAPI](#), um eine HTTP/2-Anfrage zu starten. Sie können ein benutzerdefiniertes Vokabular in Echtzeit verwenden AWS Management Console, indem Sie entweder die [StartMedicalStreamTranscriptionAPI](#) oder das WebSocket Protokoll verwenden.

Transkribieren eines in Ihr Mikrofon gesprochenen Diktats (AWS Management Console)

AWS Management Console Um das Streaming-Audio eines medizinischen Diktats 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 Management Console)

1. Melden Sie sich an der [AWS Management Console](#) an.
2. Wählen Sie im Navigationsbereich unter Amazon Transcribe Medical die Option Echtzeit-Transkription.
3. Wählen Sie als medizinisches Fachgebiet das medizinische Fachgebiet des Klinikers aus, der im Stream spricht.
4. Wählen Sie als Audioeingabetyp entweder Konversation oder Diktat.
5. Für Zusätzliche Einstellungen wählen Sie Benutzerdefiniertes Vokabular.
  - Wählen Sie für die Wortschatzauswahl das benutzerdefinierte Vokabular aus.
6. Wählen Sie dann Start Streaming (Streamen starten).
7. Sprich in das Mikrofon.

## Lautsprecherpartitionierung in einem HTTP/2-Stream aktivieren

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
```

### Parameterbeschreibungen:

- **host:** Aktualisiere den AWS-Region ('us-west-2' im vorherigen Beispiel) mit dem, den AWS-Region du anrufst. Eine Liste der gültigen AWS-Regionen Punkte finden Sie unter [AWS-Regionen und Endpoints](#).
- **Autorisierung:** Dies ist ein generiertes Feld. Weitere Informationen zum Erstellen einer Signatur finden Sie unter [Signieren von AWS Anfragen mit Signature Version 4](#).
- **x-amz-target:** Verändern Sie dieses Feld nicht, sondern verwenden Sie den im vorherigen Beispiel gezeigten Inhalt.
- **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, die die Signatur Das Format ist YYYYMMDDTHHMMSSZ, wobei YYYY=Jahr, MM=Monat, DD=Tag, HH=Stunde, MM=Minute, SS=Sekunden und 'T' und 'Z' feste Zeichen sind. Weitere Informationen finden Sie unter [Umgang mit Datern](#).
- **x-amzn-transcribe-session-id:** Der Name für deine Streaming-Sitzung.
- **x-amzn-transcribe-language-code:** Die Kodierung, die für Ihr Eingangs-Audio verwendet wird. Eine Liste der gültigen Werte finden Sie unter

## StartMedicalStreamTranscription oder Unterstützte Sprachen und sprachspezifische Funktionen.

- `x-amzn-transcribe-media-encoding`: Die Kodierung, die für Ihr Eingangs-Audio 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. Audio von geringer Qualität, z. B. Telefonaudio, liegt in der Regel bei etwa 8.000 Hz. Die Audioqualität liegt in der Regel zwischen 16.000 Hz und 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 medizinische Fachgebiet, das transkribiert wird.
- `x-amzn-transcribe-type`: Wählen Sie aus, ob es sich um ein Diktat oder eine Konversation handelt.
- `x-amzn-transcribe-show-speaker-label`: Um die Diarisierung zu aktivieren, muss dieser Wert `sein true`.
- `Inhaltstyp`: Verändern Sie dieses Feld nicht, sondern verwenden Sie den im vorherigen Beispiel gezeigten Inhalt.

### Aktivieren der Lautsprecherpartitionierung in einer WebSocket Anfrage

Um Lautsprecher in WebSocket Streams mit der API zu partitionieren, verwenden Sie das folgende Format, um eine vorsignierte URI zum Starten einer WebSocket Anfrage `vocabulary-name` zu erstellen und auf den Namen des benutzerdefinierten Vokabulars zu setzen.

```
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 Vokabeln 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

Zeichen	Code	Zeichen	Code
g	0067	ɔ	0254
h	0068	ɔɪ	0254 026A
i	0069	ə	0259
j	006A	ɛ	025B
k	006B	ɜ̄	025D
l	006C	g	0261
ɹ	006C 0329	ɪ	026A
i	006D	ɹ	0279
n	006E	ʃ	0283
ŋ	006E 0329	ʊ	028A
ou	006F 028A	ʌ	028C
p	0070	ɹ	028D
S	0073	ʒ	0292
t	0074	ɔʒ	02A4
u	0075	tʃ	02A7
V	0076	θ	03B8

## Identifizieren persönlicher Gesundheitsdaten (PHI) in einer Transkription

Verwenden Sie die persönliche Gesundheitsinformation Identifikation, um persönliche Gesundheitsinformationen (PHI) in Ihren Transkriptionsergebnissen zu kennzeichnen. Anhand der Etiketten können Sie PHI finden, anhand derer ein Patient identifiziert werden könnte.

Sie können PHI entweder mithilfe eines Echtzeit-Stream- oder Batch-Transkriptionsjobs identifizieren.

Sie können Ihre eigene Nachbearbeitung verwenden, um den in der Transkriptionsausgabe identifizierten PHI zu redigieren.

Verwenden Sie die Identifizierung persönlicher Gesundheitsinformationen, um die folgenden Arten von PHI zu identifizieren:

- Persönlicher PHI:
  - Namen — Vollständiger Name oder Nachname und Initiale
  - Geschlecht
  - Alter
  - Phone numbers (Telefonnummern)
  - Daten (ohne das Jahr), die sich direkt auf den Patienten beziehen
  - E-Mail-Adressen
- Geografischer PHI:
  - Physischer Standort
  - PLZ
  - Name des medizinischen Zentrums oder der Praxis
- Konto-PHI:
  - Faxnummern
  - Sozialversicherungsnummern (SSNs)
  - Anzahl der Leistungsempfänger der Health
  - Kontonummern
  - Zertifikat- oder Lizenznummern
- Fahrzeug-PHI:
  - Fahrgestellnummern (VIN)
  - Nummernschild
- Andere PHI:
  - Standort der einheitlichen Webressource (URL)
  - Internetprotokoll (IP) -Adressnummern

Amazon Transcribe Health Insurance Portability and Accountability Act von 1996 (HIPAA) ist Health Insurance Portability and Accountability Act. Weitere Informationen finden Sie unter [Amazon Transcribe Medical](#). Hinweise zur Identifizierung von PHI in einer Audiodatei finden Sie unter [PHI in einer Audiodatei identifizieren](#). Hinweise zur Identifizierung von PHI in einem Stream finden Sie unter [Identifizierung von PHI in einem Echtzeit-Stream](#).

## Themen

- [PHI in einer Audiodatei identifizieren](#)
- [Identifizierung von PHI in einem Echtzeit-Stream](#)

## PHI in einer Audiodatei identifizieren

Verwenden Sie einen Batch-Transkriptionsjob, 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 die PHI, die es in den Transkriptionsergebnissen identifiziert hat. Informationen zu den PHI, die Amazon Transcribe Medical identifizieren kann, finden Sie unter [Identifizieren persönlicher Gesundheitsdaten \(PHI\) in einer Transkription](#).

Sie können einen Batch-Transkriptionsjob entweder über die [StartMedicalTranscriptionJob](#) API oder die [starten AWS Management Console](#).

### AWS Management Console

Um einen Dialog zwischen Arzt und Patient in der AWS Management Console zu transkribieren, erstellen Sie einen Transkriptionsauftrag und wählen Sie Konversation als Audioeingabetyp.

Um eine Audiodatei zu transkribieren und ihre PHI zu identifizieren (AWS Management Console)

1. Melden Sie sich an der [AWS Management Console](#) an.
2. Wählen Sie im Navigationsbereich unter Amazon Transcribe Medical die Option Transcription Jobs aus.
3. Wählen Sie Create job (Auftrag erstellen) aus.
4. Geben Sie auf der Seite „Jobdetails angeben“ unter Auftragseinstellungen Folgendes an.
  - a. Name — Der Name des Transkriptionsauftrags, der für Sie einzigartig ist AWS-Konto.
  - b. Audioeingangstyp — Konversation oder Diktat.

5. Geben Sie für die verbleibenden Felder den Amazon S3 Speicherort Ihrer Audiodatei und den Ort an, an dem Sie die Ausgabe Ihres Transkriptionsauftrags speichern möchten.
6. Wählen Sie Next (Weiter).
7. Wählen Sie unter Audioeinstellungen die Option PHI-Identifizierung aus.
8. Wählen Sie Create (Erstellen) aus.

## API

Um eine Audiodatei zu transkribieren und ihren PHI mithilfe eines Batch-Transkriptionsauftrags (API) zu identifizieren

- Geben Sie für die [StartMedicalTranscriptionJob](#) API Folgendes an.
  - a. Geben Sie für `MedicalTranscriptionJobName` einen Namen an, der für Sie eindeutig ist AWS-Konto.
  - b. Geben Sie für `LanguageCode` den Sprachcode an, der der in Ihrer Audiodatei gesprochenen Sprache entspricht.
  - c. Geben Sie für den `MediaFileUri` Parameter des `Media` Objekts den Namen der Audiodatei an, die Sie transkribieren möchten.
  - d. Für `Specialty` geben Sie das medizinische Fachgebiet des Klinikers an, der in der Audiodatei spricht als `PRIMARYCARE`.
  - e. Geben Sie für `Type` entweder `CONVERSATION` oder `DICTION` an.
  - f. Geben Sie `OutputBucketName` unter den Amazon S3 Bucket an, in dem die Speicherung der Transkriptionsergebnisse gespeichert werden sollen.

Im Folgenden finden Sie ein Beispiel für eine Anfrage, die den verwendet, AWS SDK for Python (Boto3) um eine Audiodatei zu transkribieren und den PHI eines Patienten zu identifizieren.

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

```

    Media = {'MediaFileUri': job_uri},
    LanguageCode = 'en-US',
    ContentIdentificationType = 'PHI',
    Specialty = 'PRIMARYCARE',
    Type = 'type', # Specify 'CONVERSATION' for a medical conversation. Specify
    'DICTATION' for a medical dictation.
    OutputBucketName = 'DOC-EXAMPLE-BUCKET'
)
while True:
    status = transcribe.get_medical_transcription_job(MedicalTranscriptionJobName =
job_name)
    if status['MedicalTranscriptionJob']['TranscriptionJobStatus'] in ['COMPLETED',
'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
print(status)

```

Der folgende Beispielcode zeigt die Transkriptionsergebnisse mit identifiziertem PHI des Patienten.

```

{
  "jobName": "my-medical-transcription-job-name",
  "accountId": "111122223333",
  "results": {
    "transcripts": [{
      "transcript": "The patient's name is Bertrand."
    }],
    "items": [{
      "start_time": "0.0",
      "end_time": "0.37",
      "alternatives": [{
        "confidence": "0.9993",
        "content": "The"
      }],
      "type": "pronunciation"
    }, {
      "start_time": "0.37",
      "end_time": "0.44",
      "alternatives": [{
        "confidence": "0.9981",

```

```
        "content": "patient's"
    }],
    "type": "pronunciation"
}, {
    "start_time": "0.44",
    "end_time": "0.52",
    "alternatives": [{
        "confidence": "1.0",
        "content": "name"
    }],
    "type": "pronunciation"
}, {
    "start_time": "0.52",
    "end_time": "0.92",
    "alternatives": [{
        "confidence": "1.0",
        "content": "is"
    }],
    "type": "pronunciation"
}, {
    "start_time": "0.92",
    "end_time": "0.9989",
    "alternatives": [{
        "confidence": "1.0",
        "content": "Bertrand"
    }],
    "type": "pronunciation"
}, {
    "alternatives": [{
        "confidence": "0.0",
        "content": "."
    }],
    "type": "punctuation"
}],
    "entities": [{
        "content": "Bertrand",
        "category": "PHI*-Personal*",
        "startTime": 0.92,
        "endTime": 1.2,
        "confidence": 0.9989
    }],
},
    "status": "COMPLETED"
}
```

## AWS CLI

Um eine Audiodatei zu transkribieren und PHI mithilfe eines Batch-Transkriptionsjobs 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://DOC-EXAMPLE-BUCKET/my-input-files/my-audio-file.flac" \  
--output-bucket-name DOC-EXAMPLE-BUCKET \  
--specialty PRIMARYCARE \  
--type type \ # Choose CONVERSATION to transcribe a medical conversation.  
                Choose DICTATION to transcribe a medical dictation.  
--content-identification-type PHI
```

## Identifizierung von PHI in einem Echtzeit-Stream

Sie können persönliche Gesundheitsinformationen (PHI) entweder in HTTP/2 oder in WebSocket Streams identifizieren. Wenn Sie die PHI-Identifizierung aktivieren, kennzeichnet Amazon Transcribe Medical das PHI, das es identifiziert, in den Transkriptionsergebnissen. Informationen zu den PHI, die Amazon Transcribe Medical identifizieren kann, finden Sie unter [Identifizieren persönlicher Gesundheitsdaten \(PHI\) in einer Transkription](#).

PHI in einem Diktat identifizieren, das in Ihr Mikrofon gesprochen wird

AWS Management Console Um die von Ihrem Mikrofon aufgenommene Sprache zu transkribieren und PHI zu identifizieren, wählen Sie Diktat als Audioeingangstyp, starten Sie den Stream und beginnen Sie, in das Mikrofon Ihres Computers zu sprechen.

Um PHI in einem Diktat zu identifizieren, verwenden Sie den AWS Management Console

1. Melden Sie sich an der [AWS Management Console](#) an.
2. Wählen Sie im Navigationsbereich Echtzeit-Streaming aus.

3. Wählen Sie als Audioeingabetyp die Option Diktat aus.
4. Wählen Sie für Zusätzliche Einstellungen die PHI-Identifizierung.
5. Wählen Sie Streaming starten und sprechen Sie in das Mikrofon.
6. Wählen Sie Streaming beenden, um das Diktat zu beenden.

### PHI in einem HTTP/2-Stream identifizieren

Um einen HTTP/2-Stream mit aktivierter PHI-Identifikation zu starten, verwenden Sie die [StartMedicalStreamTranscriptionAPI](#) und geben Sie Folgendes an:

- Geben Sie für `LanguageCode` den Sprachcode für die im Stream gesprochene Sprache an. Geben Sie für US-Englisch `en-US` an.
- Geben Sie für `MediaSampleHertz` die Samplerate des Audios an.
- Legen Sie für `content-identification-type` die Option `PHI` fest.

### PHI in einem WebSocket Stream identifizieren

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-Vorgängen gemeinsam sind, sind im Abschnitt [Allgemeine Parameter](#) aufgeführt.

# Generierung 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 niedrigeren Konfidenzniveaus 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 AWS Management Console oder der [StartMedicalTranscriptionJob](#) API generieren.

## AWS Management Console

Um den zu verwenden AWS Management Console, um alternative Transkriptionen zu generieren, aktivieren Sie alternative Ergebnisse, wenn Sie Ihren Job konfigurieren.

1. Melden Sie sich an der [AWS Management Console](#) an.
2. Wählen Sie im Navigationsbereich unter Amazon Transcribe Medical die Option Transkriptionsjobs.
3. Wählen Sie Create job (Auftrag erstellen) aus.
4. Geben Sie auf der Seite „Stellendetails angeben“ Informationen zu Ihrem Transkriptionsjob ein.
5. Wählen Sie Next (Weiter).
6. Aktivieren Sie Alternative Ergebnisse.
7. Geben Sie für Maximale Anzahl an Alternativen eine Ganzzahl zwischen 2 und 10 für die maximale Anzahl alternativer Transkriptionen ein, die Sie in der Ausgabe haben möchten.
8. Wählen Sie Create (Erstellen) aus.

## API

So trennen Sie Text pro Sprecher in einer Audiodatei mithilfe eines Batch-Transkriptionsauftrags (API)

- Geben Sie für die [StartMedicalTranscriptionJob](#) API Folgendes an.

- a. Geben Sie für `MedicalTranscriptionJobName` einen Namen an, der in Ihrem einzigartig ist AWS-Konto.
- b. Geben Sie für `LanguageCode` den Sprachcode an, der der in Ihrer Audiodatei gesprochenen Sprache entspricht, und geben Sie die Sprache Ihres Vokabelfilters an.
- c. Geben Sie im `MediaFileUri` Parameter des `Media` Objekts den Speicherort der Audiodatei an, die Sie transkribieren möchten.
- d. Für `Specialty` geben Sie in der Audiodatei das medizinische Fachgebiet des behandelnden Arztes an.
- e. Geben Sie für `anType`, ob Sie ein medizinisches Gespräch oder ein Diktat transkribieren.
- f. Geben Sie für den Amazon S3 Bucket an `OutputBucketName`, in dem die Transkriptionsergebnisse gespeichert werden sollen.
- g. Geben Sie für das `Settings` Objekt Folgendes an.
  - i. `ShowAlternatives` – `true`.
  - ii. `MaxAlternatives`- Eine Ganzzahl 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 for Python (Boto3), um einen Transkriptionsjob 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://DOC-EXAMPLE-BUCKET/my-input-files/my-audio-file.flac
transcribe.start_medical_transcription_job(
    MedicalTranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'DOC-EXAMPLE-BUCKET',
    OutputKey = 'my-output-files/',
    LanguageCode = 'en-US',
    Specialty = 'PRIMARYCARE',
    Type = 'CONVERSATION',
```

```
    Settings = {
        'ShowAlternatives': True,
        'MaxAlternatives': 2
    }
)

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

## AWS CLI

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": "DOC-EXAMPLE-BUCKET",
    "Media": {
        "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-audio-  
file.flac"
    }
}
```

```
    },  
    "Settings":{  
        "ShowAlternatives": true,  
        "MaxAlternatives": 2  
    }  
}
```

## Amazon Transcribe VPC-Endpunkte für medizinische Anwendungen und Schnittstellen (AWS PrivateLink)

Sie können eine private Verbindung zwischen Ihrer VPC und Amazon Transcribe Medizinisch durch die Erstellung eines Schnittstelle VPC-Endpunkt. Schnittstellen-Endpunkte werden unterstützt von [AWS PrivateLink](#), eine Technologie, mit der Sie privat darauf zugreifen können Amazon Transcribe Medizinische APIs ohne Internet-Gateway, NAT-Gerät, VPN-Verbindung oder AWS Direct Connect Verbindung. Instances in Ihrer VPC benötigen keine öffentlichen IP-Adressen, um mit ihnen zu kommunizieren Amazon Transcribe Medizinische APIs. Verkehr zwischen Ihrer VPC und Amazon Transcribe Medical verlässt das Amazon-Netzwerk nicht.

Jeder Schnittstellenendpunkt wird durch eine oder mehrere [Elastic Network-Schnittstellen](#) in Ihren Subnetzen dargestellt.

Weitere Informationen finden Sie unter [Schnittstelle: VPC-Endpunkte \(AWS PrivateLink\)](#) in der Amazon VPC Benutzerleitfaden.

## Überlegungen für Amazon Transcribe VPC-Endpunkte für den medizinischen Bereich

Bevor Sie einen Schnittstellen-VPC-Endpunkt für einrichten Amazon Transcribe Medizinisch, stellen Sie sicher, dass Sie überprüfen [Eigenschaften und Einschränkungen von Schnittstellenendpunkten](#) in der Amazon VPC Benutzerleitfaden.

Amazon Transcribe Medical unterstützt das Aufrufen aller API-Aktionen von Ihrer VPC aus.

## Erstellen eines VPC-Schnittstellen-Endpunkts für Amazon Transcribe Medizinisch

Sie können einen VPC-Endpunkt für den erstellen Amazon Transcribe Ärztlicher Dienst, der entweder AWS Management Console oder der AWS CLI. Weitere Informationen finden Sie unter [Erstellen eines Schnittstellenendpunkts](#) im Amazon VPC-Benutzerhandbuch.

Für die Batch-Transkription in Amazon Transcribe Medical, erstellen Sie einen VPC-Endpunkt mit dem folgenden Dienstnamen:

- `com.amazonaws.us-west-2.transkribieren`

Für Streaming-Transkription in Amazon Transcribe Medical, erstellen Sie einen VPC-Endpunkt mit dem folgenden Dienstnamen:

- `com.amazonaws.us-west-2.transkribieren und streamen`

Wenn Sie privates DNS für den Endpunkt aktivieren, können Sie API-Anfragen an Amazon Transcribe Medical verwendet seinen Standard-DNS-Namen für AWS-Region, zum Beispiel `transcribestreaming.us-east-2.amazonaws.com`.

Weitere Informationen finden Sie unter [Zugriff auf einen Dienst über einen Schnittstellenendpunkt](#) in der Amazon VPC Benutzerleitfaden.

## Erstellen einer VPC-Endpunkt richtlinie für Amazon Transcribe Medizinisches Streaming

Sie können Ihrem VPC-Endpunkt eine Endpunkt richtlinie anhängen, die den Zugriff auf steuert Amazon Transcribe Medizinisch. 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 Transkriptionsaktionen für medizinisches Streaming

Im Folgenden finden Sie ein Beispiel für eine Endpunktrichtlinie für Streaming-Transkription in Amazon Transcribe Medizinisch. Wenn diese Richtlinie an einen Endpunkt angeschlossen ist, gewährt sie Zugriff auf die aufgelisteten Amazon Transcribe Medizinische Maßnahmen für alle Schulleiter auf allen Ressourcen.

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Allow",
      "Action": [
        "transcribe:StartMedicalStreamTranscription",
      ],
      "Resource": "*"
    }
  ]
}
```

## Beispiel: VPC-Endpunktrichtlinie für Amazon Transcribe Transkriptionsmaßnahmen für medizinische Chargen

Im Folgenden finden Sie ein Beispiel für eine Endpunktrichtlinie für die Batch-Transkription in Amazon Transcribe Medizinisch. Wenn diese Richtlinie an einen Endpunkt angeschlossen ist, gewährt sie Zugriff auf die aufgelisteten Amazon Transcribe Medizinische Maßnahmen für alle Schulleiter auf allen Ressourcen.

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Allow",
      "Action": [
        "transcribe:StartMedicalTranscriptionJob"
      ],
      "Resource": "*"
    }
  ]
}
```

## Gemeinsame Subnetze

Sie können VPC-Endpoints 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 unter [Teilen Sie Ihre VPC mit anderen Konten](#) in der Amazon Virtual Private Cloud Führer.

# AWS-HealthScribe

AWS-HealthScribe ist eine neue, HIPAA-konforme maschinelle Lernfunktion (ML), die Spracherkennung und generative KI kombiniert, um Gespräche zwischen Patient und Arzt zu transkribieren und einfach zu überprüfende klinische Notizen zu erstellen. AWS HealthScribe unterstützt Anbieter von Software für das Gesundheitswesen bei der Entwicklung klinischer Anwendungen, die den Dokumentationsaufwand verringern und die Beratungsqualität verbessern. Der Service liefert automatisch umfangreiche Konversationstranskripte, identifiziert Sprecherrollen, klassifiziert Dialoge, extrahiert medizinische Begriffe und erstellt vorläufige klinische Notizen. AWS HealthScribe kombiniert diese Funktionen und macht die Integration und Optimierung separater KI-Services überflüssig, sodass Sie die Implementierung beschleunigen können.

Häufige Anwendungsfälle:

- Reduzieren des Zeitaufwands für die Dokumentation – Ermöglichen Sie es dem medizinischen Fachpersonal, die klinische Dokumentation schnell mit KI-generierten klinischen Notizen zu vervollständigen, die in Ihrer Anwendung einfach zu überprüfen, anzupassen und abzuschließen sind.
- Steigern der Effizienz von medizinischem Personal – Statten Sie medizinisches Personal mit KI-generierten Transkripten und klinischen Notizen zusammen mit den Audiodaten der Konsultation aus, um die Durchlaufzeit 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 von AWS HealthScribe erzeugten Ergebnisse sind probabilistisch und aufgrund von verschiedenen Faktoren wie der Audioqualität, Hintergrundgeräuschen, der Klarheit des Sprechers, der Komplexität der medizinischen Terminologie, kontextspezifischen Sprachnuancen und dem [Wesen von Machine Learning und generativer KI](#) möglicherweise nicht immer korrekt. AWS HealthScribe ist so konzipiert, dass es Ärzte und medizinische Schreibkräfte unterstützt. Die Ausgaben von AWS HealthScribe sollten nur dann in der Patientenversorgung, einschließlich aber nicht beschränkt auf elektronische Patientenakten, verwendet werden, wenn sie von geschultem medizinischem Fachpersonal auf ihre Richtigkeit überprüft und mit fundiertem medizinischem Urteilsvermögen beurteilt wurden. Die Ausgaben von AWS HealthScribe sind kein Ersatz für professionelle medizinische Beratung,

Diagnose oder Behandlung und nicht dazu bestimmt, Krankheiten oder Gesundheitsprobleme zu heilen, zu behandeln, zu mildern, zu verhindern oder zu diagnostizieren.

AWS-HealthScribe arbeitet nach einem Modell der geteilten Verantwortung, bei dem AWS für den Schutz der Infrastruktur verantwortlich ist, auf der AWS-HealthScribe läuft, und Sie für die Verwaltung Ihrer Daten verantwortlich sind. Weitere Informationen finden Sie unter [Modell der gemeinsamen Verantwortung](#).

AWS HealthScribe ist in der Region US East (N. Virginia) verfügbar.

Der Service ist in US-Englisch (en-US) verfügbar. Die besten Ergebnisse erzielen Sie, wenn Sie ein verlustfreies Audioformat wie FLAC oder WAV mit PCM 16-Bit-Kodierung verwenden. AWS HealthScribe unterstützt Sampleraten von 16.000 Hz oder höher.

AWS HealthScribe unterstützt derzeit die Fachgebiete Allgemeinmedizin und Orthopädie.

Ein AWS-HealthScribe-Auftrag analysiert medizinische Konsultationen und erzeugt zwei JSON-Ausgabedateien: eine [Transkriptdatei](#) und eine Datei mit [klinischer Dokumentation](#).

In der Transkriptionsdatei bietet AWS-HealthScribe neben der standardmäßigen Turn-by-Turn-Transkriptionsausgabe mit Zeitstempeln auf Wortebene weitere Informationen:

- 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.

In der Datei mit klinischer Dokumentation stellt Ihnen AWS HealthScribe folgende Informationen zur Verfügung:

- Zusammenfassungen mit zusammengefassten Notizen zu den wichtigsten Abschnitten der klinischen Dokumentation, wie z. B. Hauptbeschwerde, aktuelle Krankheitsgeschichte, Systemprüfung, vergangene Krankheitsgeschichte, Beurteilung und Plan.
- Links zu Nachweisen, die jeden in der KI-generierten Zusammenfassung verwendeten Satz mit dem Original-Konsultationstranskript verknüpfen, was es den Benutzern erleichtert, die Richtigkeit der Zusammenfassung in Ihrer Anwendung zu überprüfen.

API-Vorgänge speziell für AWS-HealthScribe:

- StartMedicalScribeJob
- ListMedicalScribeJobs
- GetMedicalScribeJob
- DeleteMedicalScribeJob

Beispiele für AWS-HealthScribe-Anforderungen finden Sie unter [Starten eines AWS-HealthScribe-Auftrags](#).

## Transkriptionsdatei

Die Transkriptionsdatei enthält den Inhalt des Gesprächs in einem Format, in dem Sie die einzelnen Schritte nachvollziehen können.

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\_1, CLINICIAN\_2 und PATIENT\_1, PATIENT\_2.
- 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.
  - Flow-Management besuchen – Informationen, die sich auf Smalltalk oder Übergänge beziehen.
- Einblicke – Extrahiert klinisch relevante Entitäten (ClinicalEntity), die in der Konversation vorhanden sind. AWS HealthScribe erkennt alle klinischen Entitäten, die von [Amazon Comprehend Medical](#) unterstützt werden.

Ausführlichere Informationen zur Ausgabe finden Sie unter [Beispiel für die Transkriptausgabe](#).

## Datei mit klinischer Dokumentation

Die Datei mit den Dokumentationseinblicken enthält Zusammenfassungen der folgenden wichtigen Abschnitte der klinischen Dokumentation.

Abschnitt	Beschreibung
HAUPTBESCHWERDE	Kurze Beschreibung des Grundes für den Besuch des Patienten beim Arzt.
VORGESCHICHTE DER AKTUELLEN KRANKHEIT	Notizen, die Informationen über die Krankheit des Patienten enthalten, einschließlich Angaben über Schweregrad, Beginn und Zeitpunkt der Symptome, aktuelle Behandlungen und betroffene Bereiche.
SYSTEMÜBERPRÜFUNG	Bewertung von Symptomen in verschiedenen Körpersystemen nach Angaben der Patienten.
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.

Jeder Satz in der Summary enthält Verweise auf das Original-Transkript der Konsultation, sodass es für Nutzer einfacher ist, die Richtigkeit der Zusammenfassung in Ihrer Anwendung zu überprüfen. Die Nachvollziehbarkeit und Transparenz von KI-generierten Erkenntnissen steht im Einklang mit den Grundsätzen verantwortungsvoller KI, wie z. B. Erklärbarkeit. Die Bereitstellung dieser Referenzen zusammen mit den zusammenfassenden Notizen für Ärzte oder medizinische Schreibkräfte trägt dazu bei, Vertrauen zu schaffen und die sichere Verwendung von KI im klinischen Umfeld zu fördern.

Jeder Satz in der Summary ist mit EvidenceLinks versehen, die SegmentId für die entsprechenden Dialoge im Transkript, die zusammengefasst wurden, bereitstellt.

Ausführlichere Informationen zur Ausgabe finden Sie in der [Beispielausgabe für klinische Dokumentation](#).

## Start eines AWS-HealthScribe-Auftrags

Sie können einen AWS-HealthScribe-Auftrag über die AWS-CLI oder AWS-SDKs starten; Beispiele finden Sie im Folgenden.

### 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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac \  
--output-bucket-name DOC-EXAMPLE-BUCKET \  
--DataAccessRoleArn=arn:aws:iam::111122223333:role/ExampleRole \  
--settings ShowSpeakerLabels=false,ChannelIdentification=true \  
--channel-definitions ChannelId=0,ParticipantRole=CLINICIAN \  
ChannelId=1,ParticipantRole=PATIENT
```

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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
},
"OutputBucketName": "DOC-EXAMPLE-BUCKET",
"DataAccessRoleArn": "arn:aws:iam::111122223333:role/ExampleRole",
"Settings": {
  "ShowSpeakerLabels": false,
  "ChannelIdentification": true
},
"ChannelDefinitions": [
  {
    "ChannelId": 0,
    "ParticipantRole": "CLINICIAN"
  }, {
    "ChannelId": 1,
    "ParticipantRole": "PATIENT"
  }
]
}

```

## AWS SDK for Python (Boto3)

Das folgende Beispiel verwendet die AWS SDK for Python (Boto3), um einen [start\\_medical\\_scribe\\_job](#)-Auftrag zu erteilen. 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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
transcribe.start_medical_scribe_job(
    MedicalScribeJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'DOC-EXAMPLE-BUCKET',
    DataAccessRoleArn = 'arn:aws:iam::111122223333:role/ExampleRole',
    Settings = {
        'ShowSpeakerLabels': false,
        'ChannelIdentification': true
    }
)

```

```
    },
    ChannelDefinitions = [
      {
        'ChannelId': 0,
        'ParticipantRole': 'CLINICIAN'
      }, {
        'ChannelId': 1,
        'ParticipantRole': 'PATIENT'
      }
    ]
  )
while True:
    status = transcribe.get_medical_scribe_job(MedicalScribeJobName = job_name)
    if status['MedicalScribeJob']['MedicalScribeJobStatus'] in ['COMPLETED', 'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
print(status)
```

### Note

Die AWS-Management-Konsole unterstützt derzeit keine AWS-HealthScribe-Aufträge.

## Beispielausgabe

Zusätzlich zu einem Transkript erzeugen `StartMedicalScribeJob`-Anforderungen eine separate Datei mit klinischer Dokumentation. Beide Dateien liegen im JSON-Format vor und werden an dem von Ihnen in der Anfrage angegebenen Ausgabeort gespeichert. Hier finden Sie Beispiele für die einzelnen Ausgabearten:

### Beispiel einer Transkriptionsausgabe

Eine AWS-HealthScribe-Transkriptionsdatei (aus einer `StartMedicalScribeJob`-Anforderung) 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,
    "EndAudioTime": 0,
    "Type": "PUNCTUATION"
  },
  {
    "Alternatives": [
```

```
    {
      "Confidence": 1,
      "Content": "Good"
    }
  ],
  "BeginAudioTime": 0.61,
  "EndAudioTime": 0.92,
  "Type": "PRONUNCIATION"
},
{
  "Alternatives": [
    {
      "Confidence": 1,
      "Content": "afternoon"
    }
  ],
  "BeginAudioTime": 0.92,
  "EndAudioTime": 1.54,
  "Type": "PRONUNCIATION"
},
{
  "Alternatives": [
    {
      "Confidence": 0,
      "Content": "."
    }
  ],
  "BeginAudioTime": 0,
  "EndAudioTime": 0,
  "Type": "PUNCTUATION"
},
{
  "Alternatives": [
    {
      "Confidence": 0.9924,
      "Content": "You"
    }
  ],
  "BeginAudioTime": 1.55,
  "EndAudioTime": 1.88,
  "Type": "PRONUNCIATION"
},
{
  "Alternatives": [
```

```
    {
      "Confidence": 1,
      "Content": "lost"
    }
  ],
  "BeginAudioTime": 1.88,
  "EndAudioTime": 2.19,
  "Type": "PRONUNCIATION"
},
{
  "Alternatives": [
    {
      "Confidence": 1,
      "Content": "one"
    }
  ],
  "BeginAudioTime": 2.19,
  "EndAudioTime": 2.4,
  "Type": "PRONUNCIATION"
},
{
  "Alternatives": [
    {
      "Confidence": 1,
      "Content": "lb"
    }
  ],
  "BeginAudioTime": 2.4,
  "EndAudioTime": 2.97,
  "Type": "PRONUNCIATION"
}
],
"TranscriptSegments": [
  {
    "BeginAudioTime": 0.16,
    "Content": "Okay.",
    "EndAudioTime": 0.6,
    "ParticipantDetails": {
      "ParticipantRole": "CLINICIAN_0"
    },
    "SectionDetails": {
      "SectionName": "SUBJECTIVE"
    },
    "SegmentId": "uuid1"
  }
]
```

```
  },
  {
    "BeginAudioTime": 0.61,
    "Content": "Good afternoon.",
    "EndAudioTime": 1.54,
    "ParticipantDetails": {
      "ParticipantRole": "CLINICIAN_0"
    },
    "SectionDetails": {
      "SectionName": "OTHER"
    },
    "SegmentId": "uuid2"
  },
  {
    "BeginAudioTime": 1.55,
    "Content": "You lost one lb.",
    "EndAudioTime": 2.97,
    "ParticipantDetails": {
      "ParticipantRole": "CLINICIAN_0"
    },
    "SectionDetails": {
      "SectionName": "SUBJECTIVE"
    },
    "SegmentId": "uuid3"
  },
  {
    "BeginAudioTime": 2.98,
    "Content": "Yeah, I think it, uh, do you feel more energy?",
    "EndAudioTime": 6.95,
    "ParticipantDetails": {
      "ParticipantRole": "CLINICIAN_0"
    },
    "SectionDetails": {
      "SectionName": "SUBJECTIVE"
    },
    "SegmentId": "uuid5"
  },
  {
    "BeginAudioTime": 6.96,
    "Content": "Yes.",
    "EndAudioTime": 7.88,
    "ParticipantDetails": {
      "ParticipantRole": "CLINICIAN_0"
    },
  },
```

```

    "SectionDetails": {
      "SectionName": "SUBJECTIVE"
    },
    "SegmentId": "uuid6"
  },
  {
    "BeginAudioTime": 7.89,
    "Content": "Uh, how about craving for the carbohydrate or sugar or fat or anything?",
    "EndAudioTime": 17.93,
    "ParticipantDetails": {
      "ParticipantRole": "CLINICIAN_0"
    },
    "SectionDetails": {
      "SectionName": "SUBJECTIVE"
    },
    "SegmentId": "uuid7"
  }
]
}
}

```

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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
  },
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",
  "DataAccessRoleArn": "arn:aws:iam::111122223333:role/ExampleRole",
  "Settings": {

```

```
"ShowSpeakerLabels": false,
"ChannelIdentification": true
},
"ChannelDefinitions": [
  {
    "ChannelId": 0,
    "ParticipantRole": "CLINICIAN"
  }, {
    "ChannelId": 1,
    "ParticipantRole": "PATIENT"
  }
]
}
```

### 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."
    }
  ]
}
```

```
}  
  ]  
}  
}
```

## Datenverschlüsselung im Ruhezustand für AWS HealthScribe

AWS HealthScribe bietet standardmäßig eine Verschlüsselung, um vertrauliche Kundendaten im Ruhezustand mithilfe von Amazon S3-verwalteten Schlüsseln zu schützen.

- Amazon S3-verwaltete Schlüssel (SSE-S3) – AWS HealthScribe verwendet standardmäßig Amazon S3-verwaltete Schlüssel zum automatischen Verschlüsseln von Zwischendateien. Sie können Amazon S3-verwaltete Schlüssel nicht anzeigen, verwalten oder verwenden und auch nicht ihre Nutzung prüfen. Sie müssen jedoch keine Maßnahmen ergreifen oder Programme ändern, um die Schlüssel zu schützen, die zur Verschlüsselung Ihrer Daten verwendet werden. Weitere Informationen finden Sie unter [SSE-S3](#).

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. Gleichzeitig können Sie damit sichere Anwendungen erstellen, die strenge Verschlüsselungsvorschriften und gesetzliche Auflagen erfüllen.

Sie können diese Verschlüsselungsebene zwar nicht deaktivieren und auch keinen alternativen Verschlüsselungstyp auswählen, doch Sie können eine zweite Verschlüsselungsebene über den vorhandenen Amazon S3-verwalteten Schlüsseln hinzufügen, indem Sie bei der Erstellung eines Auftrags mit AWS HealthScribe einen vom Kunden verwalteten Schlüssel auswählen.

- 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 Pflege von IAM-Richtlinien und -Berechtigungen
  - 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 verwaltete Schlüssel](#) im Entwicklerhandbuch zum AWS Key Management Service.

#### Note

AWS HealthScribe aktiviert automatisch die Verschlüsselung im Ruhezustand mithilfe AWS-eigener Schlüssel, um persönlich identifizierbare Informationen kostenlos zu schützen.

Allerdings fallen AWS KMS-Gebühren für die Verwendung eines kundenverwalteten Schlüssels an. Weitere Informationen zu Preisen finden Sie unter [AWS Key Management Service-Preise](#).

Weitere Informationen zu AWS KMS finden Sie unter [Was ist AWS Key Management Service](#).

## Einen kundenverwalteten Schlüssel erstellen

Sie können einen symmetrischen, vom Kunden verwalteten Schlüssel erstellen, indem Sie die AWS Management Console oder die AWS KMS-APIs verwenden. Folgen Sie dazu den Schritten zum [Erstellen eines symmetrischen kundenverwalteten Schlüssels](#) im Entwicklerhandbuch zum AWS Key Management Service.

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 vom Kunden verwalteten Schlüssel erstellen, können Sie eine Schlüsselrichtlinie angeben. Weitere Informationen finden Sie unter [Verwalten des Zugriffs auf kundenverwaltete Schlüssel](#) im Entwicklerhandbuch zum AWS Key Management Service.

Wenn Sie einen Schlüssel verwenden, der sich in demselben Konto wie die IAM-Rolle befindet, die Sie in Ihrer Anforderung [StartMedicalScribeJob](#) als [DataAccessRoleARN](#) angeben, müssen Sie die Schlüsselrichtlinie nicht aktualisieren. Um Ihren vom Kunden verwalteten Schlüssel in einem anderen Konto als Ihre `DataAccessRole` zu verwenden, müssen Sie `DataAccessRoleARN` in der Schlüsselrichtlinie für die folgenden Aktionen als vertrauenswürdig einstufen:

- [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 die Details zu dem vom Kunden verwalteten Schlüssel bereit, damit AWS HealthScribe den Schlüssel validieren kann

Im Folgenden finden Sie ein Beispiel für eine Richtlinienanweisung, die Sie hinzufügen können, um Ihrer IAM-Rolle kontoübergreifende Berechtigungen für die Verwendung Ihres vom Kunden verwalteten Schlüssels zu erteilen:

```
"Statement" : [
  {
    "Sid": "Allow access to the DataAccessRole for StartMedicalScribeJob",
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::111122223333:role/DataAccessRole"
    },
    "Action": [
      "kms:DescribeKey",
      "kms:Encrypt",
      "kms:Decrypt",
      "kms:GenerateDataKey"
    ],
    "Resource" : "*"
  }
]
```

Unabhängig davon, ob sich Ihr vom Kunden verwalteter Schlüssel und Ihre DataAccessRole in demselben Konto oder in unterschiedlichen Konten befinden, benötigt Ihre DataAccessRole Berechtigungen, um die oben genannten Aktionen mit Ihrem vom Kunden verwalteten Schlüssel ausführen zu können. Im Folgenden sehen Sie ein Beispiel für eine Richtlinienanweisung, die Sie zu Ihrer DataAccessRole hinzufügen können:

```
"Statement" : [
  {
    "Sid": "Allow role to perform AWS KMS actions for customer managed key",
    "Effect": "Allow",
    "Action": [
      "kms:DescribeKey",
```

```
    "kms:Encrypt",
    "kms:Decrypt"
  ],
  "Resource": "*"
}
```

Weitere Informationen zum [Festlegen von Berechtigungen in einer Richtlinie](#) finden Sie im Entwicklerhandbuch zum AWS Key Management Service. Weitere Informationen zur [Fehlerbehebung beim Schlüsselzugriff](#) finden Sie im Entwicklerhandbuch zum AWS Key Management Service.

## Angeben eines vom Kunden verwalteten Schlüssels für AWS HealthScribe

Sie können einen vom Kunden verwalteten Schlüssel als zweite Verschlüsselungsebene für Anforderungen `StartMedicalScribeJob` angeben. Wenn Sie eine Anforderung [StartMedicalScribeJob](#) erstellen, können Sie einen vom Kunden verwalteten Schlüssel angeben, indem Sie das Feld [OutputEncryptionKMSKeyId](#) in Ihre Anforderung aufnehmen.

## AWS KMS-Verschlüsselungskontext

AWS KMS-Verschlüsselungskontext ist eine Karte mit nicht geheimen Schlüssel:Wert-Paaren im Klartext. Diese Karte stellt zusätzliche authentifizierte Daten dar, die als Verschlüsselungskontextpaare bekannt sind und eine zusätzliche Sicherheitsebene für Ihre Daten bieten. AWS HealthScribe benötigt einen symmetrischen Verschlüsselungsschlüssel, um die Ausgabe von AWS HealthScribe in einem vom Kunden angegebenen Amazon S3-Bucket zu verschlüsseln. 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, legen Sie den Parameter [KMSEncryptionContext](#) in der Operation [StartMedicalScribeJob](#) fest. Um einen Verschlüsselungskontext für die Ausgabeverschlüsselung bereitzustellen, muss der Parameter [OutputEncryptionKMSKeyId](#) auf eine symmetrische AWS KMS-Schlüssel-ID verweisen.

Sie können [AWS KMS-Bedingungsschlüssel](#) mit IAM-Richtlinien verwenden, um den Zugriff auf einen symmetrischen AWS KMS-Verschlüsselungsschlüssel auf der Grundlage des Verschlüsselungskontexts zu steuern, der in der Anforderung für eine [kryptografischen Operation](#) 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](#).

# 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
<a href="#">Feature-Update</a>	Aktualisieren Sie die maximale Anzahl an Lautsprechern in der Diarisierung auf 30 statt auf 10.	10. Mai 2024
<a href="#">Abschnitt-Update</a>	Die generative Anrufzusammenfassung wurde aktualisiert und es wurden Details zur Fehlerausgabe hinzugefügt.	30. April 2024
<a href="#">Abschnitt-Update</a>	Aktualisierungen der Spalten mit benutzerdefiniertem Wortschatz — IPA und SoundsLike	30. April 2024
<a href="#">Feature-Update</a>	Amazon Transcribe Call Analytics unterstützt jetzt die generative Anrufzusammenfassung.	29. November 2023
<a href="#">Abschnitt-Update</a>	Aktualisierung des neuen Ausgabeformats mit PII-Schwärzung und Sprachenidentifizierung.	13. November 2023

---

<a href="#">Feature-Update</a>	Die Diarisierung kann nun mit der Kanalidentifizierung kombiniert werden.	6. März 2023
<a href="#">Feature-Update</a>	Die Kanalidentifizierung kann nun mit der Diarisierung kombiniert werden.	6. März 2023
<a href="#">Abschnitt-Update</a>	IAM Die bewährten Verfahren wurden aktualisiert.	13. Februar 2023
<a href="#">Neue Sprachen</a>	Amazon Transcribe unterstützt jetzt Vietnamesisch und Schwedisch.	6. Dezember 2022
<a href="#">Neues Feature</a>	Amazon Transcribe unterstützt jetzt Anrufanalysen in Echtzeit.	28. November 2022
<a href="#">Feature-Update</a>	Streaming-Schwärzung und -Identifizierung ist jetzt auch in Hindi und Thai verfügbar.	11. November 2022
<a href="#">Abschnitt-Update</a>	Für die Schwärzung und Identifizierung von Datenströmen sind neue PII-Kategorien verfügbar.	14. September 2022
<a href="#">Abschnitt-Update</a>	Der Abschnitt über das benutzerdefinierte Sprachmodell wurde überarbeitet.	18. Juni 2022
<a href="#">Abschnitt-Update</a>	Die Batch-Sprachidentifizierung kann nun mehrere Sprachen pro Audiodatei identifizieren.	31. Mai 2022
<a href="#">Leitfaden-Update</a>	Die Amazon Transcribe API-Referenz ist jetzt ein eigenständiger Leitfaden.	1. April 2022

---

<a href="#">Neues Kapitel</a>	Eine neue Vergleichstabelle für Amazon Transcribe Medical Analytics und Amazon Transcribe Call Analytics ist enthalten.	21. März 2022
<a href="#">Neues Kapitel</a>	Ein neues Kapitel mit SDK-Codebeispielen wurde aufgenommen.	21. März 2022
<a href="#">Feature-Update</a>	Call Analytics bietet jetzt eine Anrufzusammenfassung.	21. März 2022
<a href="#">Kapitel-Update</a>	Im Einführungskapitel werden nun Amazon Transcribe Anwendungsfälle vorgestellt.	21. März 2022
<a href="#">Kapitel-Update</a>	Das Kapitel „Erste Schritte“ wurde aktualisiert und ist nun methodenspezifisch.	21. März 2022
<a href="#">Kapitel-Update</a>	Das Kapitel Streaming wurde aktualisiert und umstrukturiert.	21. März 2022
<a href="#">Feature-Update</a>	Die Sprachidentifizierung unterstützt jetzt benutzerdefinierte Vokabulare und benutzerdefinierte Wortschatzfilter mit Streaming-Transkriptionen.	11. März 2022
<a href="#">Neues Ereignis</a>	Es gibt einen neuen Ereignistyp: Vokabularereignisse.	7. Februar 2022
<a href="#">Abschnitt-Update</a>	Der Bereich der benutzerdefinierten Vokabulare wurde aktualisiert.	20. Januar 2022

---

<a href="#">Neues Feature</a>	Die Sprachidentifizierung kann jetzt auch bei Streaming-Transkriptionen verwendet werden.	23. November 2021
<a href="#">Neues Feature</a>	Die Sprachidentifizierung kann jetzt mit benutzerdefinierten Sprachmodellen, benutzerdefinierten Vokabularen, Wortschatzfilterung und Inhaltsredaktion verwendet werden.	29. Oktober 2021
<a href="#">Neues Feature</a>	Amazon Transcribe unterstützt jetzt benutzerdefinierte Sprachmodelle mit Streaming-Transkriptionen.	20. Oktober 2021
<a href="#">Neues Feature</a>	Amazon Transcribe kann jetzt Untertitel für Ihre Videodateien generieren.	16. September 2021
<a href="#">Neues Feature</a>	Amazon Transcribe unterstützt jetzt die Schwärzung und Identifizierung personenbezogener Daten für das Streaming.	14. September 2021
<a href="#">Neues Feature</a>	Amazon Transcribe unterstützt jetzt den AWS KMS Verschlüsselungskontext für ein zusätzliches Maß an Sicherheit für Ihre AWS-Konto Ressourcen.	10. September 2021

---

<a href="#">Neue Sprachen</a>	Amazon Transcribe unterstützt jetzt Afrikaans, Dänisch, Mandarin-Chinesisch (traditionell), Thailändisches, neuseeländisches Englisch und südafrikanisches Englisch.	26. August 2021
<a href="#">Neues Feature</a>	Amazon Transcribe unterstützt jetzt Ressourcen-Tagging.	24. August 2021
<a href="#">Neues Feature</a>	Amazon Transcribe unterstützt jetzt Call Analytics für Batch-Transkriptionsaufträge.	4. August 2021
<a href="#">Neues Feature</a>	Amazon Transcribe unterstützt jetzt die Verwendung von benutzerdefinierten Vokabularen mit benutzerdefinierten Batch-Sprachmodellen.	12. Mai 2021
<a href="#">Neues Feature</a>	Amazon Transcribe unterstützt jetzt die teilweise Stabilisierung der Ergebnisse bei der Streaming-Transkription.	11. Mai 2021
<a href="#">Neues Feature</a>	Amazon Transcribe unterstützt jetzt australisches Englisch, britisches Englisch, Hindi und US-Spanisch für benutzerdefinierte Sprachmodelle.	19. März 2021
<a href="#">Neues Feature</a>	Amazon Transcribe unterstützt jetzt die Codecs OGG/OPUS und FLAC für die Streaming-Audiotranskription.	24. November 2020

---

<a href="#">Neue Sprachen</a>	Amazon Transcribe fügt Unterstützung für Italienisch und Deutsch für die Streaming-Audiotranskription hinzu.	4. November 2020
<a href="#">AWS-Region Erweiterung</a>	Amazon Transcribe ist jetzt in Frankfurt (eu-central-1) und London (eu-west-2) verfügbar.	4. November 2020
<a href="#">Neues Feature</a>	Amazon Transcribe fügt Unterstützung für VPC-Schnittstellen-Endpunkte bei der Batch-Transkription hinzu.	9. Oktober 2020
<a href="#">Neues Feature</a>	Amazon Transcribe fügt Unterstützung für die Kanalidentifikation beim Streaming hinzu.	17. September 2020
<a href="#">Neues Feature</a>	Amazon Transcribe fügt Unterstützung für die automatische Sprachidentifikation bei der Batch-Transkription hinzu.	15. September 2020
<a href="#">Neues Feature</a>	Amazon Transcribe fügt Unterstützung für die Lautsprecherpartitionierung beim Streaming hinzu.	19. August 2020
<a href="#">Neues Feature</a>	Amazon Transcribe fügt Unterstützung für benutzerdefinierte Sprachmodelle hinzu.	5. August 2020
<a href="#">Neues Feature</a>	Amazon Transcribe fügt Unterstützung für VPC-Schnittstellen-Endpunkte beim Streaming hinzu.	26. Juni 2020

---

<a href="#">Neues Feature</a>	Amazon Transcribe fügt Unterstützung für die Vokabelfilterung beim Streaming hinzu.	20. Mai 2020
<a href="#">Neues Feature</a>	Amazon Transcribe fügt Unterstützung für das automatische Redigieren personenbezogener Daten hinzu.	26. Februar 2020
<a href="#">Neues Feature</a>	Amazon Transcribe fügt Unterstützung für die Erstellung eines benutzerdefinierten Wortschatzes hinzu, das aus einer Transkription herausgefiltert werden soll.	20. Dezember 2019
<a href="#">Neues Feature</a>	Amazon Transcribe fügt Unterstützung für das Einreihen von Transkriptionsaufträgen in die Warteschlange hinzu.	19. Dezember 2019
<a href="#">Neue Sprachen</a>	Amazon Transcribe fügt Unterstützung für Golfarabisch, Hebräisch, Japanisch, Malaiisch, Schweizerdeutsch, Telugu und Türkisch hinzu.	21. November 2019
<a href="#">AWS-Region Erweiterung</a>	Amazon Transcribe ist jetzt im asiatisch-pazifischen Raum (Tokio) (ap-northeast-1) erhältlich.	21. November 2019
<a href="#">Neues Feature</a>	Amazon Transcribe fügt Unterstützung für alternative Transkriptionen hinzu.	20. November 2019

---

<a href="#">Neue Sprachen</a>	Amazon Transcribe fügt Unterstützung für Niederländisch, Farsi, Indonesisch, Irisches Englisch, Portugiesisch, Schottisches Englisch, Tamil und Walisisches Englisch hinzu.	12. November 2019
<a href="#">Neue Sprache</a>	Amazon Transcribe unterstützt jetzt Streaming-Transkription für australisches Englisch (en-AU).	25. Oktober 2019
<a href="#">AWS-Region Erweiterung</a>	Amazon Transcribe ist jetzt in China (Peking) (cn-north-1) und China (Ningxia) (cn-north-west-1) verfügbar.	9. Oktober 2019
<a href="#">Neues Feature</a>	Amazon Transcribe ermöglicht es Ihnen, Ihre eigenen Dateien zur Verschlüsselung Ihrer Transkriptionsausgabedateien bereitzustellen. KMS key Weitere Informationen finden Sie im <a href="#">OutputEncryptionKeyIdKMS-Parameter</a> der <a href="#">StartStreamTranscriptionAPI</a> .	24. September 2019
<a href="#">Neue Sprachen</a>	Amazon Transcribe fügt Unterstützung für Chinesisch (Mandarin), vereinfachtes Chinesisch, Festlandchina und Russisch hinzu.	23. August 2019

---

<a href="#">Neues Feature</a>	Amazon Transcribe fügt Unterstützung für die Streaming-Audiotranskription mithilfe des WebSocket Protokolls hinzu.	19. Juli 2019
<a href="#">Neues Feature</a>	AWS CloudTrail zeichnet jetzt Ereignisse für die <a href="#">StartStreamTranscriptionAPI</a> auf.	19. Juli 2019
<a href="#">AWS-Region Erweiterung</a>	Amazon Transcribe ist jetzt in den USA West (Nordkalifornien) (us-west-1) verfügbar.	27. Juni 2019
<a href="#">Neue Sprache</a>	Amazon Transcribe fügt Unterstützung für Modernes Standardarabisch hinzu.	28. Mai 2019
<a href="#">Neues Feature</a>	Amazon Transcribe transkribiert jetzt numerische Wörter in Zahlen für US-Englisch. Beispiel: "zweiundvierzig" wird als "42" transkribiert.	23. Mai 2019
<a href="#">Neue Sprache</a>	Amazon Transcribe fügt Unterstützung für Hindi und indisches Englisch hinzu.	15. Mai 2019
<a href="#">Neues SDK</a>	Das AWS SDK for C++ unterstützt jetzt Amazon Transcribe.	8. Mai 2019
<a href="#">Neue Sprache</a>	Amazon Transcribe fügt Unterstützung für Spanisch hinzu.	19. April 2019

---

<a href="#">AWS-Region Erweiterung</a>	Amazon Transcribe ist jetzt in der EU (Frankfurt) (eu-central-1) und im asiatisch-pazifischen Raum (Seoul) (ap-northeast-2) verfügbar.	18. April 2019
<a href="#">Neue Sprache</a>	Amazon Transcribe fügt Unterstützung für Streaming-Transkription in britischem Englisch, Französisch und kanadischem Französisch hinzu.	5. April 2019
<a href="#">Neues Feature</a>	Das AWS SDK for Ruby V3 unterstützt jetzt Amazon Transcribe	25. März 2019
<a href="#">Neues Feature</a>	Amazon Transcribe ermöglicht benutzerdefinierte Vokabulare, d. h. Listen mit bestimmten Wörtern, die Sie in Ihrer Audioeingabe wiedererkennen Amazon Transcribe möchten.	25. März 2019
<a href="#">Neue Sprachen</a>	Amazon Transcribe fügt Unterstützung für Deutsch und Koreanisch hinzu.	22. März 2019
<a href="#">Neue Sprache</a>	Amazon Transcribe unterstützt jetzt Streaming-Transkription für US-Spanisch (es-US).	7. Februar 2019
<a href="#">AWS-Region Erweiterung</a>	Amazon Transcribe ist jetzt in Südamerika (São Paulo) (sa-east-1) erhältlich.	7. Februar 2019

---

<a href="#">AWS-Region Erweiterung</a>	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
<a href="#">Neue Sprachen</a>	Amazon Transcribe fügt Unterstützung für Französisch, Italienisch und brasilianisches Portugiesisch hinzu.	20. Dezember 2018
<a href="#">Neues Feature</a>	Amazon Transcribe unterstützt jetzt die Transkription von Audiostreams.	19. November 2018
<a href="#">Neue Sprachen</a>	Amazon Transcribe fügt Unterstützung für australisches Englisch, britisches Englisch und kanadisches Französisch hinzu.	15. November 2018
<a href="#">AWS-Region Erweiterung</a>	Amazon Transcribe ist jetzt in Kanada (Zentral) (ca-central-1) und im asiatisch-pazifischen Raum (Sydney) (ap-south east-2) erhältlich.	17. Juli 2018
<a href="#">Neues Feature</a>	Sie können nun zum Speichern der Ausgabe eines Transkriptionsauftrags Ihren eigenen Standort angeben.	11. Juli 2018
<a href="#">Neues Feature</a>	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.