aws

Panduan Developer

Amazon Transcribe

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon Transcribe: Panduan Developer

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara pelanggan, atau dengan cara apa pun yang merendahkan atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan hak milik masing-masing pemiliknya, yang mungkin atau tidak terafiliasi, terkait dengan, atau disponsori oleh Amazon.

Table of Contents

Apa itu Amazon Transcribe? 1
Amazon Transcribe dan kelayakan HIPAA 1
Harga 2
Ketersediaan dan kuota wilayah 2
Fitur yang tersedia 4
Bahasa yang didukung 7
Bahasa pemrograman yang didukung 16
Set karakter 17
Abkhaz 21
Afrikaans 23
Arab 24
Asturian 25
Orang Azerbaijan 26
Orang Armenia 26
Bashkir 28
Basque 30
Belarusia 31
Bengali 32
Orang Bosnia 34
Bulgaria 34
bahasa katala 35
Kurdi Tengah 36
Mandarin, Sederhana 37
Tionghoa, Tradisional 38
Mandarin, Kanton 39
orang Kroasia 40
Bahasa Ceko 40
Orang Denmark 41
Bahasa Belanda 41
Bahasa Inggris 42
Estonia 43
Farsi 43
orang Finlandia 44
Prancis 45

Galicia 46
Orang Georgia 47
Bahasa Jerman 48
Yunani 48
Gujarat 50
Hausa 52
Ibrani 52
bahasa Hindi 53
Bahasa Hungaria 55
Islandia 56
orang Indonesia 57
Bahasa Italia 57
Bahasa Jepang 58
Kabyle 58
Kannada 59
Kazakh 61
Kinyarwanda 62
Bahasa Korea 63
bahasa Kirgiz 63
Latvia 66
Lituania 66
Luganda 67
Makedonia 67
Melayu 70
Malayalam 70
Malta 72
Marathi 73
Padang Rumput Mari 75
Mongolia 77
Bokmål Norwegia 80
Odia/Oriya 80
Pashto 82
Polandia 84
Bahasa Portugis 84
Punjabi 86
Rumania 87

Bahasa Rusia 88
Serbia 89
Sinhala 92
Orang Slovakia 94
Bahasa Slovenia 95
Somalia 95
Bahasa Spanyol 96
Sunda 96
Bahasa Swahili 97
Bahasa Swedia 97
Tagalog/Bahasa Filipino 98
Tamil 98
Tatar 100
Telugu 102
Thai 104
Turki 106
orang Ukraina 107
Uyghur 108
Uzbek 111
Vietnam 112
Welsh 116
Wolof 117
Zulu 118
Cara kerjanya 119
Input dan output data 119
Format media 120
Saluran audio 121
Tarif sampel 122
Output 122
Mentranskripsikan angka 126
Memulai 127
Mendaftar untuk Akun AWS 128
Instalasi AWS CLI dan SDKs 128
Konfigurasikan IAM kredensil 129
Membuat Amazon S3 ember 129
Membuat IAM kebijakan 130

Mentranskripsikan dengan AWS Management Console 132
Mentranskripsikan dengan AWS CLI 141
Memulai pekerjaan transkripsi baru 141
Mendapatkan status pekerjaan transkripsi 143
Daftar pekerjaan transkripsi Anda 144
Menghapus pekerjaan transkripsi Anda 144
Mentranskripsikan dengan AWS SDKs 145
Bekerja dengan AWS SDKs 157
Mentranskripsikan dengan HTTP atau WebSockets 158
Transkripsi streaming 160
Praktik terbaik 161
Streaming dan hasil sebagian 162
Stabilisasi hasil sebagian 163
Menyiapkan transkripsi streaming 167
Pengkodean aliran acara 181
Bingkai data 184
Antrian pekerjaan 185
Mengaktifkan antrian pekerjaan 185
Penandaan pada sumber daya 190
Kontrol akses berbasis tag 191
Menambahkan tag ke Amazon Transcribe sumber daya Anda 192
Mempartisi speaker (diarisasi) 196
Mempartisi speaker dalam transkripsi batch 197
Mempartisi speaker dalam transkripsi streaming 200
Contoh Output 203
Mentranskripsikan audio multi-saluran 210
Menggunakan identifikasi saluran dalam transkripsi batch 211
Menggunakan identifikasi saluran dalam transkripsi streaming 215
Contoh Output 216
Mengidentifikasi bahasa 225
Identifikasi bahasa Batch 225
Mengidentifikasi bahasa dalam audio multi-bahasa 226
Meningkatkan akurasi identifikasi bahasa 227
Menggabungkan identifikasi bahasa dengan Amazon Transcribe fitur lain 228
Menggunakan identifikasi bahasa dengan transkripsi batch 229
Identifikasi bahasa streaming 236

Mengidentifikasi bahasa dalam audio multi-bahasa 237
Menggunakan identifikasi bahasa dengan media streaming 237
Transkripsi alternatif 245
Meminta transkripsi alternatif 247
Meningkatkan akurasi transkripsi 252
Kosakata khusus 253
Tabel kosakata khusus versus daftar 254
Membuat kosakata khusus menggunakan tabel 255
Membuat kosakata khusus menggunakan daftar 266
Menggunakan kosakata khusus 269
Model bahasa khusus 276
Sumber data 276
Pelatihan versus penyetelan data 277
Membuat model bahasa khusus 278
Menggunakan model bahasa khusus 283
Menyaring kata-kata 291
Membuat filter kosakata 292
Membuat filter kosakata khusus 293
Menggunakan filter kosakata khusus 298
Menggunakan filter kosakata khusus dalam transkripsi batch 269
Menggunakan filter kosakata khusus dalam transkripsi streaming 273
Mendeteksi ucapan beracun 307
Menggunakan deteksi ucapan beracun 308
Menggunakan deteksi ucapan beracun dalam transkripsi batch 308
Contoh Output 313
Menyunting transkrip 315
Menyunting PII dalam pekerjaan batch Anda 316
Menyunting atau mengidentifikasi PII dalam streaming waktu nyata 323
Contoh Output 329
Contoh keluaran yang disunting (batch) 329
Contoh output streaming yang disunting 332
Contoh keluaran identifikasi PII 333
Membuat subtitle 336
Menghasilkan file subtitle 337
Menganalisis audio call center 340
Kasus penggunaan umum 340

Pertimbangan dan informasi tambahan 342
Ketersediaan dan kuota wilayah 343
Analisis pasca-panggilan 344
Wawasan pasca-panggilan 344
Membuat kategori 347
Memulai transkripsi 360
Output analitik pasca-panggilan 369
Mengaktifkan ringkasan panggilan generatif 382
Analisis Panggilan Waktu Nyata 387
Wawasan waktu nyata 388
Membuat kategori 390
Analisis pasca-panggilan dengan transkripsi waktu nyata 397
Memulai transkripsi 405
Output Analisis Panggilan Waktu Nyata 413
Mentranskripsikan panggilan Amazon Chime 419
Contoh kode 421
Hal-hal mendasar 422
Tindakan 422
Skenario 485
Membangun aplikasi streaming Amazon Transcribe 486
Mengonversi teks menjadi ucapan dan kembali ke teks 486
Membuat dan menyempurnakan kosakata khusus 487
Transcribe audio dan dapatkan data pekerjaan 497
Keamanan 509
Identity and Access Management 510
Audiens 510
Mengautentikasi dengan identitas 511
Mengelola akses menggunakan kebijakan 514
Bagaimana Amazon Transcribe bekerja dengan IAM 517
Pencegahan Deputi Bingung 524
Contoh kebijakan berbasis identitas 525
Pemecahan Masalah 535
Perlindungan data 537
Privasi lalu lintas antar jaringan 538
Enkripsi data 538
Memilih untuk tidak menggunakan data Anda untuk perbaikan layanan 541

Pemantauan Amazon Transcribe 541
Pemantauan CloudWatch dengan 542
Pemantauan Amazon Transcribe dengan CloudTrail 543
Menggunakan Amazon EventBridge dengan Amazon Transcribe 547
Validasi kepatuhan 556
Ketahanan 557
Keamanan infrastruktur 557
Analisis dan manajemen kerentanan 558
Titik akhir VPC (AWS PrivateLink) 558
Subnet bersama 560
Praktik terbaik keamanan 560
Amazon Transcribe Medis 563
Ketersediaan dan kuota wilayah 564
Spesialisasi medis 565
Mentranskripsikan istilah dan pengukuran medis 566
Mentranskripsikan angka 569
Mentranskripsikan percakapan medis 571
Mentranskripsikan file audio 572
Mentranskripsikan aliran waktu nyata 577
Mengaktifkan partisi speaker 580
Mentranskripsikan audio multi-saluran 589
Mentranskripsikan dikte medis 597
Mentranskripsikan file audio 598
Mentranskripsikan dikte medis streaming 603
Membuat dan menggunakan kosakata khusus medis 605
Membuat file teks untuk kosakata khusus medis Anda 606
Menggunakan file teks untuk membuat kosakata khusus medis 611
Mentranskripsikan file audio menggunakan kosakata khusus medis 613
Mentranskripsikan aliran waktu nyata menggunakan kosakata khusus medis 614
Set karakter untuk Amazon Transcribe Medis 617
Mengidentifikasi PHI dalam transkrip 619
Mengidentifikasi PHI dalam file audio 620
Mengidentifikasi PHI dalam streaming waktu nyata 625
Menghasilkan transkripsi alternatif 627
Titik akhir VPC (AWS PrivateLink) 630
Pertimbangan untuk titik akhir VPC Amazon Transcribe Medis 630

Membuat titik akhir VPC antarmuka untuk Medis Amazon Transcribe 630
Membuat kebijakan titik akhir VPC untuk streaming Medis Amazon Transcribe 631
Subnet bersama 632
AWS HealthScribe 633
Keamanan 634
Ketersediaan layanan 634
Persyaratan teknis 634
Spesialisasi Medis yang Didukung 635
Alur Kerja 635
Berkas transkrip 636
File Dokumentasi Klinis 637
Bagian template HISTORY_AND_PHYSICAL 638
Bagian template GIRPP 639
Bagian template BIRP 640
Bagian template SIRP 641
Bagian template DAP 641
Bagian template BH_SOAP 642
Bagian template PH_SOAP 642
Lowongan kerja transkripsi 643
Memulai AWS HealthScribe pekerjaan transkripsi 643
Streaming 658
Pedoman dan persyaratan 659
ResourceAccessRoleArn izin peran 659
Memulai transkripsi AWS HealthScribe streaming 661
Enkripsi Data saat istirahat untuk AWS HealthScribe 677
Membuat kunci yang dikelola pelanggan 679
Menentukan kunci yang dikelola pelanggan untuk AWS HealthScribe 682
AWS KMS konteks enkripsi 683
Memantau kunci enkripsi Anda untuk AWS HealthScribe 686
Riwayat dokumen 689
AWS Glosarium 701
..... dccii

Apa itu Amazon Transcribe?

Amazon Transcribe adalah layanan pengenalan suara otomatis yang menggunakan model pembelajaran mesin untuk mengonversi audio menjadi teks. Anda dapat menggunakan Amazon Transcribe sebagai layanan transkripsi mandiri atau untuk menambahkan speech-to-text kemampuan ke aplikasi apa pun.

Dengan Amazon Transcribe, Anda dapat meningkatkan akurasi untuk kasus penggunaan spesifik Anda dengan penyesuaian bahasa, memfilter konten untuk memastikan privasi pelanggan atau bahasa yang sesuai dengan audiens, menganalisis konten dalam audio multi-saluran, mempartisi ucapan masing-masing speaker, dan banyak lagi.

Anda dapat mentranskripsikan media secara real time (streaming) atau Anda dapat mentranskripsikan file media yang terletak di Amazon S3 ember (batch). Untuk melihat bahasa mana yang didukung untuk setiap jenis transkripsi, lihat Bahasa yang didukung dan fitur khusus bahasa tabel.

Topik

- Amazon Transcribe dan kelayakan HIPAA
- Harga
- Ketersediaan dan kuota wilayah

Lihat Apa itu Amazon Transcribe? untuk tur video singkat dari layanan ini.

Untuk mempelajari selengkapnya, lihat Bagaimana cara Amazon Transcribe kerja dan Memulai dengan Amazon Transcribe.

> **Tip**
>
> Informasi tentang Amazon Transcribe API terletak di Referensi API.

Amazon Transcribe dan kelayakan HIPAA

Amazon Transcribe tercakup dalam AWS kelayakan HIPAA dan BAA yang mengharuskan pelanggan BAA untuk mengenkripsi semua PHI saat istirahat dan dalam perjalanan saat digunakan. Identifikasi

PHI otomatis tersedia tanpa biaya tambahan dan di semua wilayah tempat Amazon Transcribe beroperasi. Untuk informasi lebih lanjut, lihat kelayakan HIPAA dan BAA.

Harga

Amazon Transcribe adalah pay-as-you-go layanan; harga didasarkan pada detik audio yang ditranskripsi, ditagih setiap bulan.

Penggunaan ditagih dalam kenaikan satu detik, dengan biaya minimum per permintaan 15 detik. Perhatikan bahwa biaya tambahan berlaku untuk fitur seperti redaksi konten PII dan model bahasa khusus.

Untuk informasi biaya masing-masing Wilayah AWS, lihat Amazon Transcribe Harga.

Ketersediaan dan kuota wilayah

Amazon Transcribe didukung sebagai berikut Wilayah AWS:

Wilayah	Jenis transkripsi
af-south-1 (Cape Town)	batch, streaming
ap-east-1 (Hong Kong)	batch
ap-northeast-1 (Tokyo)	batch, streaming
ap-northeast-2 (Seoul)	batch, streaming
ap-south-1 (Mumbai)	batch, streaming
ap-southeast-1 (Singapura)	batch, streaming
ap-southeast-2 (Sydney)	batch, streaming
ca-central-1 (Kanada, Tengah)	batch, streaming
eu-central-1 (Frankfurt am Main)	batch, streaming
eu-north-1 (Stockholm)	batch

Wilayah	Jenis transkripsi
eu-west-1 (Irlandia)	batch, streaming
eu-west-2 (London)	batch, streaming
eu-west-3 (Paris)	batch
me-south-1 (Bahrain)	batch
sa-east-1 (sa-east-1)	batch, streaming
us-east-1 (Virginia Utara)	batch, streaming
us-east-2 (Ohio)	batch, streaming
us-gov-east-1 (GovCloud, AS-Timur)	batch, streaming
us-gov-west-1 (GovCloud, AS-Barat)	batch, streaming
us-west-1 (San Francisco)	batch
us-west-2 (Oregon)	batch, streaming

Important

Dukungan wilayah berbeda untuk Amazon Transcribe, Amazon Transcribe Medical, dan Call Analytics.

Untuk mendapatkan titik akhir untuk setiap Wilayah yang didukung, lihat Titik akhir Layanan di Referensi AWS Umum.

Untuk daftar kuota yang berkaitan dengan transkripsi Anda, lihat kuota Layanan di Referensi Umum.AWS Beberapa kuota dapat diubah berdasarkan permintaan. Jika kolom Adjustable berisi 'Ya', Anda dapat meminta peningkatan. Untuk melakukannya, pilih tautan yang disediakan.

Amazon Transcribe fitur

Untuk membantu Anda memutuskan Amazon Transcribe solusi mana yang paling sesuai dengan kasus penggunaan Anda, tabel berikut menawarkan perbandingan fitur.

Perhatikan bahwa 'batch' dan 'post-call' merujuk untuk menyalin file yang terletak di Amazon S3 ember dan 'streaming' dan 'real-time' merujuk ke media transkripsi secara real time.

Fitur	Amazon Transcribe	Amazon Transcribe Medical[1]	Amazon Transcribe Panggilan Analytics
Opsi konfigurasi			
Transkripsi alternatif	batch, streaming	batch, streaming	tidak
Identifikasi saluran	batch, streaming	batch, streaming	post-call, real-time
Antrian pekerjaan	batch	tidak	post-call
Identifikasi bahasa	batch, streaming	tidak	post-call
Identifikasi multi-bahasa	batch, streaming	tidak	tidak
Diarisasi speaker	batch, streaming	batch, streaming	post-call
Mentranskripsikan digit 2	batch, streaming	batch, streaming	post-call, real-time
Analisis percakapan			
Karakteristik panggilan	tidak	tidak	post-call
Ringkasan panggilan[2]	tidak	tidak	post-call
Kategorisasi kustom	tidak	tidak	post-call
Acara kategori waktu nyata	tidak	tidak	real-time

Fitur	Amazon Transcribe	Amazon Transcribe Medical[1]	Amazon Transcribe Panggilan Analytics
Deteksi masalah waktu nyata [2]	tidak	tidak	real-time
Sentimen pembicara waktu nyata	tidak	tidak	real-time
Sentimen pembicara	tidak	tidak	post-call
Kustomisasi bahasa			
Model bahasa khusus[2]	batch, streaming	tidak	post-call, real-time
Kosakata khusus	batch, streaming	batch, streaming	post-call, real-time
Organisasi sumber daya			
Menandai	batch	batch	post-call
Data sensitif			
Mengidentifikasi informasi kesehatan pribadi [2]	tidak	batch, streaming	tidak
Mengidentifikasi informasi yang dapat diidentifikasi secara pribadi [2]	streaming	tidak	real-time
Menyunting audio 2	tidak	tidak	post-call, real-time
Menyunting transkrip 2	batch, streaming	tidak	post-call, real-time
Pemfilteran kosakata	batch, streaming	tidak	post-call, real-time
Video			

Fitur	Amazon Transcribe	Amazon Transcribe Medical[1]	Amazon Transcribe Panggilan Analytics
Sub judul	batch	tidak	tidak

[1] Amazon Transcribe Medis hanya tersedia dalam bahasa Inggris AS.
[2] Fitur ini tidak tersedia untuk semua bahasa; tinjau Bahasa yang didukung dan fitur khusus bahasa tabel untuk lebih jelasnya.

Bahasa yang didukung dan fitur khusus bahasa

Bahasa yang didukung oleh Amazon Transcribe tercantum dalam tabel berikut; juga tercantum adalah fitur yang khusus bahasa. Harap verifikasi bahwa fitur yang ingin Anda gunakan didukung untuk bahasa di media Anda sebelum melanjutkan dengan transkripsi Anda.

Untuk melihat daftar lengkap Amazon Transcribe fitur, lihat Ringkasan fitur.

Dalam tabel berikut, 'batch' mengacu pada transkripsi file media yang terletak di Amazon S3 ember dan 'streaming' mengacu pada transkripsi media streaming secara real time. Untuk transkripsi Call Analytics, 'post-call' mengacu pada transkripsi file media yang terletak di Amazon S3 bucket dan 'real-time' mengacu pada transkripsi media streaming secara real time.

Bahasa	Kode bahasa #	Masukan data	Mentranskripsikan angka	Akronim	Model bahasa khusus	Redaksi	Analitik Panggilan *
Abkhaz	ab-GE	batch	tidak	batch	tidak	tidak	tidak
Afrikaans	af-ZA	batch, streaming	tidak	batch, streaming	tidak	tidak	tidak
Arab, Teluk	ar-AE	batch, streaming	tidak	tidak	tidak	tidak	post-call
Arab, Standar Modern	ar-SA	batch, streaming	tidak	tidak	tidak	tidak	tidak
Orang Armenia	hy-AM	batch	tidak	batch	tidak	tidak	tidak
Asturian	ast-ES	batch	tidak	batch	tidak	tidak	tidak
Orang Azerbaijan	az-AZ	batch	tidak	batch	tidak	tidak	tidak
Bashkir	ba-RU	batch	tidak	batch	tidak	tidak	tidak

Bahasa	Kode bahasa #	Masukan data	Mentranskripsikan angka	Akronim	Model bahasa khusus	Redaksi	Analitik Panggilan *
Basque	eu-ES	batch, streaming	tidak	batch, streaming	tidak	tidak	tidak
Belarusia	be-BY	batch	tidak	batch	tidak	tidak	tidak
Bengali	bn-IN	batch	tidak	batch	tidak	tidak	tidak
Orang Bosnia	bs-BA	batch	tidak	batch	tidak	tidak	tidak
Bulgaria	bg-BG	batch	tidak	batch	tidak	tidak	tidak
bahasa katala	ca-ES	batch, streaming	streaming	batch, streaming	tidak	tidak	tidak
Kurdi Tengah, Iran	ckb-IR	batch	tidak	batch	tidak	tidak	tidak
Kurdi Tengah, Iraq	ckb-IQ	batch	tidak	batch	tidak	tidak	tidak
Mandarin, Kanton	zh-HK (yue-HK)	batch, streaming	tidak	tidak	tidak	tidak	tidak
Mandarin, Sederhana	zh-CN	batch, streaming	tidak	tidak	tidak	tidak	post-call
Tionghoa, Tradisional	zh-TW	batch, streaming	tidak	tidak	tidak	tidak	tidak

Bahasa	Kode bahasa #	Masukan data	Mentranskripsikan angka	Akronim	Model bahasa khusus	Redaksi	Analitik Panggilan *
orang Kroasia	hr-HR	batch, streaming	tidak	batch, streaming	tidak	tidak	tidak
Bahasa Ceko	cs-CZ	batch, streaming	tidak	batch, streaming	tidak	tidak	tidak
Orang Denmark	da-DK	batch, streaming	tidak	batch, streaming	tidak	tidak	tidak
Bahasa Belanda	nl-NL	batch, streaming	tidak	batch, streaming	tidak	tidak	tidak
Bahasa Inggris, Australia	en-AU	batch, streaming	batch, streaming	batch, streaming	batch, streaming	streaming	post-call, real-time
Bahasa Inggris, Inggris	en-GB	batch, streaming	batch, streaming	batch, streaming	batch, streaming	streaming	post-call, real-time
Bahasa Inggris, India	en-IN	batch, streaming	batch, streaming	batch, streaming	tidak	tidak	post-call
Bahasa Inggris, Irlandia	en-IE	batch, streaming	batch, streaming	batch, streaming	tidak	tidak	post-call
Bahasa Inggris, Selandia Baru	en-NZ	batch, streaming	batch, streaming	batch, streaming	tidak	tidak	tidak

Bahasa	Kode bahasa #	Masukan data	Mentranskripsikan angka	Akronim	Model bahasa khusus	Redaksi	Analitik Panggilan *
Bahasa Inggris, Skotlandia	en-AB	batch, streaming	batch, streaming	batch, streaming	tidak	tidak	post-call
Bahasa Inggris, Afrika Selatan	en-ZA	batch, streaming	batch, streaming	batch, streaming	tidak	tidak	tidak
Bahasa Inggris, AS	en-US	batch, streaming	batch, streaming	batch, streaming	batch, streaming	batch, streaming	post-call, real-time
Bahasa Inggris, Welsh	en-WL	batch, streaming	batch, streaming	batch, streaming	tidak	tidak	post-call
Estonia	et-EE	batch	tidak	batch	tidak	tidak	tidak
Estonia	et-ET	batch	tidak	batch	tidak	tidak	tidak
Farsi	fa-IR	batch, streaming	tidak	tidak	tidak	tidak	tidak
orang Finlandia	fi-FI	batch, streaming	tidak	batch, streaming	tidak	tidak	tidak
Prancis	fr-FR	batch, streaming	batch, streaming	batch, streaming	tidak	tidak	post-call, real-time
Prancis, Kanada	fr-CA	batch, streaming	batch, streaming	batch, streaming	tidak	tidak	post-call, real-time

Bahasa	Kode bahasa #	Masukan data	Mentranskripsikan angka	Akronim	Model bahasa khusus	Redaksi	Analitik Panggilan *
Galicia	gl-ES	batch, streaming	tidak	batch, streaming	tidak	tidak	tidak
Orang Georgia	ka-GE	batch	tidak	batch	tidak	tidak	tidak
Bahasa Jerman	de-DE	batch, streaming	batch, streaming	batch, streaming	batch, streaming	tidak	post-call, real-time
Bahasa Jerman, Swiss	de-CH	batch, streaming	batch, streaming	batch, streaming	tidak	tidak	post-call
Yunani	el-GR	batch, streaming	tidak	batch, streaming	tidak	tidak	tidak
Gujarat	gu-IN	batch	tidak	batch	tidak	tidak	tidak
Hausa	ha-NG	batch	tidak	batch	tidak	tidak	tidak
Ibrani	he-IL	batch, streaming	tidak	tidak	tidak	tidak	tidak
bahasa Hindi, India	hi-IN	batch, streaming	tidak	batch, streaming	batch	tidak	post-call
Bahasa Hungaria	hu-HU	batch	tidak	batch	tidak	tidak	tidak
Islandia	is-IS	batch	tidak	batch	tidak	tidak	tidak
orang Indonesia	id-ID	batch, streaming	tidak	batch, streaming	tidak	tidak	tidak

Bahasa	Kode bahasa #	Masukan data	Mentranskripsikan angka	Akronim	Model bahasa khusus	Redaksi	Analitik Panggilan *
Bahasa Italia	it-IT	batch, streaming	batch, streaming	batch, streaming	tidak	tidak	post-call, real-time
Bahasa Jepang	ja-JP	batch, streaming	batch, streaming	tidak	batch, streaming	tidak	post-call
Kabyle	kab-DZ	batch	tidak	batch	tidak	tidak	tidak
Kannada	kn-IN	batch	tidak	batch	tidak	tidak	tidak
Kazakh	kk-KZ	batch	tidak	batch	tidak	tidak	tidak
Kinyarwanda	rw-RW	batch	tidak	batch	tidak	tidak	tidak
Bahasa Korea	ko-KR	batch, streaming	tidak	tidak	tidak	tidak	post-call
bahasa Kirgiz	ky-KG	batch	tidak	batch	tidak	tidak	tidak
Latvia	lv-LV	batch, streaming	tidak	batch, streaming	tidak	tidak	tidak
Lituania	lt-LT	batch	tidak	batch	tidak	tidak	tidak
Luganda	lg-IN	batch	tidak	batch	tidak	tidak	tidak
Makedonia	mk-MK	batch	tidak	batch	tidak	tidak	tidak
Melayu	ms-MY	batch, streaming	tidak	batch, streaming	tidak	tidak	tidak
Malayalam	ml-IN	batch	tidak	batch	tidak	tidak	tidak
Malta	mt-MT	batch	tidak	batch	tidak	tidak	tidak

Bahasa	Kode bahasa #	Masukan data	Mentranskripsikan angka	Akronim	Model bahasa khusus	Redaksi	Analitik Panggilan *
Marathi	mr-IN	batch	tidak	batch	tidak	tidak	tidak
Padang Rumput Mari	mhr-RU	batch	tidak	batch	tidak	tidak	tidak
Mongolia	mn-MN	batch	tidak	batch	tidak	tidak	tidak
Bokmål Norwegia	no-NO	batch, streaming	tidak	batch, streaming	tidak	tidak	tidak
Odia/Oriya	or-IN	batch	tidak	batch	tidak	tidak	tidak
Pashto	ps-AF	batch	tidak	batch	tidak	tidak	tidak
Polandia	pl-PL	batch, streaming	tidak	batch, streaming	tidak	tidak	tidak
Bahasa Portugis	pt-PT	batch, streaming	batch, streaming	batch, streaming	tidak	tidak	post-call
Bahasa Portugis, Brasil	pt-BR	batch, streaming	batch, streaming	batch, streaming	tidak	tidak	post-call, real-time
Punjabi	pa-IN	batch	tidak	batch	tidak	tidak	tidak
Rumania	ro-RO	batch, streaming	tidak	batch, streaming	tidak	tidak	tidak
Bahasa Rusia	ru-RU	batch, streaming	tidak	tidak	tidak	tidak	tidak
Serbia	sr-RS	batch, streaming	tidak	batch, streaming	tidak	tidak	tidak

Bahasa	Kode bahasa #	Masukan data	Mentranskripsikan angka	Akronim	Model bahasa khusus	Redaksi	Analitik Panggilan *
Sinhala	si-LK	batch	tidak	batch	tidak	tidak	tidak
Orang Slovakia	sk-SK	batch, streaming	tidak	batch, streaming	tidak	tidak	tidak
Bahasa Slovenia	sl-SI	batch	tidak	batch	tidak	tidak	tidak
Somalia	so-SO	batch, streaming	tidak	batch, streaming	tidak	tidak	tidak
Bahasa Spanyol	es-ES	batch, streaming	batch, streaming	batch, streaming	tidak	tidak	post-call
Bahasa Spanyol, AS	es-US	batch, streaming	batch, streaming	batch, streaming	batch, streaming	batch, streaming	post-call, real-time
Sunda	su-ID	batch	tidak	batch	tidak	tidak	tidak
Bahasa Swahili, Kenya	sw-KE	batch	tidak	batch	tidak	tidak	tidak
Bahasa Swahili, Burundi	sw-BI	batch	tidak	batch	tidak	tidak	tidak
Bahasa Swahili, Rwanda	sw-RW	batch	tidak	batch	tidak	tidak	tidak
Bahasa Swahili, Tanzania	sw-TZ	batch	tidak	batch	tidak	tidak	tidak

Bahasa	Kode bahasa #	Masukan data	Mentranskripsikan angka	Akronim	Model bahasa khusus	Redaksi	Analitik Panggilan *
Bahasa Swahili, Uganda	sw-UG	batch	tidak	batch	tidak	tidak	tidak
Bahasa Swedia	sv-SE	batch, streaming	tidak	batch, streaming	tidak	tidak	tidak
Tagalog/ Bahasa Filipino	tl-PH	batch, streaming	tidak	batch, streaming	tidak	tidak	tidak
Tamil	ta-IN	batch	tidak	tidak	tidak	tidak	tidak
Tatar	tt-RU	batch	tidak	batch	tidak	tidak	tidak
Telugu	te-IN	batch	tidak	tidak	tidak	tidak	tidak
Thai	th-TH	batch, streaming	tidak	batch, streaming	tidak	tidak	tidak
Turki	tr-TR	batch	tidak	batch	tidak	tidak	tidak
orang Ukraina	uk-UA	batch, streaming	tidak	batch, streaming	tidak	tidak	tidak
Uyghur	ug-CN	batch	tidak	batch	tidak	tidak	tidak
Uzbek	uz-UZ	batch	tidak	batch	tidak	tidak	tidak
Vietnam	vi-VN	batch, streaming	tidak	batch, streaming	tidak	tidak	tidak
Welsh	cy-WL	batch	tidak	batch	tidak	tidak	tidak
Wolof	wo-SN	batch	tidak	batch	tidak	tidak	tidak

Bahasa	Kode bahasa #	Masukan data	Mentranskripsikan angka	Akronim	Model bahasa khusus	Redaksi	Analitik Panggilan *
Zulu	zu-ZA	batch, streaming	tidak	batch, streaming	tidak	tidak	tidak

*Wawasan Call Analytics berikut hanya didukung dalam dialek bahasa Inggris tertentu:

- Ringkasan panggilan: en-* (semua dialek bahasa Inggris)
- Deteksi masalah:en-AU,en-GB, en-US

Kode Bahasa tidak didukung di wilayah AWS GovCloud (US) (AS-barat, us-gov-west -1), AWS GovCloud (US) (AS-Timur, us-gov-east -1), atau Afrika (Cape Town, af-south-1):

- Kode bahasa: ab-GEast-ES, az-AZba-RU,be-BY,bg-BG, bn-INbs-BA,ca-ES,ckb-IQ,ckb-IR, cs-CZcy-WL,el-GR,et-EE, et-ETeu-ES,fi-FI,gl-ES, gu-INha-NG,hr-HR,hu-HU, hy-AMis-IS,ka-GE,kab-DZ,kk-KZ, kn-INky-KG,lg-IN,lt-LT, lv-LVmhr-RU,mi-NZ,mk-MK, ml-INmn-MN,mr-IN,mt-MT, no-NO,or-IN,pa-IN,pl-PL,ps-AF,ro-RO,rw-RW,si-LK,sk-SK,sl-SI,so-SO,sr-RS,,su-ID,sw-BI,sw-KE,sw-RW,sw-TZ,sw-UG,tl-PH,tt-RU,ug-CN,uk-UA,uz-UZ,wo-SN, zu-ZA

Bahasa pemrograman yang didukung

Amazon Transcribe mendukung yang berikut AWS SDKs:

Transkripsi Batch	Transkripsi streaming
.NET	.NET (Transcribe Medical dan tidak HealthScribe didukung)
AWS Antarmuka Baris Perintah (CLI)	CLI tidak didukung untuk streaming.
C++	C++
Go	Go

Transkripsi Batch	Transkripsi streaming
Java V2	Java V2
JavaScript	JavaScript V3
PHP V3	PHP V3
Python Boto3	Python Streaming SDK untuk Amazon Transcribe
Ruby V3	Ruby V3
Karat	Karat

Untuk informasi tentang penggunaan SDKs dengan Amazon Transcribe, lihatMentranskripsikan dengan AWS SDKs.

Untuk informasi selengkapnya tentang semua alat yang tersedia AWS SDKs dan alat pembangun, lihat Alat untuk Dibangun AWS.

> **Tip**
>
> Anda dapat menemukan contoh kode SDK di GitHub repositori ini:
>
> - AWS Contoh Kode
> - Amazon Transcribe Contoh

Set karakter untuk kosakata khusus dan filter kosakata

Untuk setiap Amazon Transcribe dukungan bahasa, ada seperangkat karakter tertentu yang Amazon Transcribe dapat dikenali. Saat Anda membuat filter kosakata atau kosakata khusus, gunakan hanya karakter yang tercantum dalam kumpulan karakter bahasa Anda. Jika Anda menggunakan karakter yang tidak didukung, filter kosakata atau kosakata khusus Anda gagal.

> **Important**
>
> Pastikan untuk memeriksa bahwa file kosakata khusus Anda hanya menggunakan poin kode Unicode yang didukung dan urutan titik kode yang tercantum dalam kumpulan karakter berikut.

Banyak karakter Unicode dapat tampak identik dalam font populer, bahkan jika mereka menggunakan poin kode yang berbeda. Hanya poin kode yang tercantum dalam panduan ini yang didukung. Misalnya, kata Prancis déjà dapat dirender menggunakan karakter yang telah disusun sebelumnya (di mana satu nilai Unicode mewakili karakter beraksen) atau karakter yang didekomposisi (di mana dua nilai Unicode mewakili karakter beraksen, satu nilai untuk karakter dasar dan satu lagi untuk aksen).

- Versi yang telah disusun sebelumnya: 0064 **`00E9`** 006A **`00E0`** (dirender sebagai deja)
- Versi dekomposisi: 0064 **`0065 0301`** 006A **`0061 0300`** (dirender sebagai déja)

Topik

- Set karakter Abkhaz
- Set karakter Afrikaans
- Set karakter Arab
- Set karakter Asturian
- Set karakter Azerbaijan
- Set karakter Armenia
- Set karakter Bashkir
- Set karakter Basque
- Set karakter Belarusia
- Set karakter Bengali
- Set karakter Bosnia
- Set karakter Bulgaria
- Set karakter Catalan
- Set karakter Kurdi Tengah
- Tionghoa, Mandarin (Tiongkok Daratan), Set karakter yang disederhanakan

- Tionghoa, Mandarin (Taiwan), Set karakter tradisional
- Tionghoa, Kanton (Hong Kong), Set karakter tradisional
- Set karakter Kroasia
- Set karakter Ceko
- Set karakter Denmark
- Set karakter Belanda
- Set karakter bahasa Inggris
- Set karakter Estonia
- Set karakter Farsi
- Set karakter Finlandia
- Set karakter Prancis
- Set karakter Galicia
- Set karakter Georgia
- Set karakter Jerman
- Set karakter Yunani
- Set karakter Gujarati
- Set karakter Hausa
- Set karakter Ibrani
- Set karakter Hindi
- Set karakter Hungaria
- Set karakter Islandia
- Set karakter Indonesia
- Set karakter Italia
- Set karakter Jepang
- Set karakter Kabyle
- Set karakter Kannada
- Set karakter Kazakh
- Set karakter Kinyarwanda
- Set karakter Korea
- Set karakter Kirgistan

- Set karakter Latvia
- Set karakter Lithuania
- Set karakter Luganda
- Set karakter Makedonia
- Set karakter Melayu
- Set karakter Malayalam
- Set karakter Malta
- Set karakter Marathi
- Set karakter Meadow Mari
- Set karakter Mongolia
- Set karakter Bokmål Norwegia
- Set karakter Odia/Oriya
- Set karakter Pashto
- Set karakter Polandia
- Set karakter Portugis
- Set karakter Punjabi
- Set karakter Rumania
- Set karakter Rusia
- Set karakter Serbia
- Set karakter Sinhala
- Set karakter Slovakia
- Set karakter Slovenia
- Set karakter Somalia
- Set karakter Spanyol
- Set karakter Sunda
- Set karakter Swahili
- Set karakter Swedia
- Set karakter Tagalog/Filipina
- Set karakter Tamil
- Set karakter Tatar

- Set karakter Telugu
- Set karakter Thailand
- Set karakter Turki
- Set karakter Ukraina
- Set karakter Uyghur
- Set karakter Uzbek
- Set karakter Vietnam
- Set karakter Welsh
- Set karakter Wolof
- Set karakter Zulu

Set karakter Abkhaz

Untuk kosakata kustom Abkhaz, Anda dapat menggunakan karakter berikut di bidang: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
а	0430	љ	0459
b	0431	њ	045A
в	0432	ћ	045B
г	0433	(b)	045C
(c)	0434	#	045D
е	0435	ў	045E
ж	0436	џ	045F

Karakter	Kode	Karakter	Kode
(b)	0437	ҁ	0491
и	0438	҂	0493
в	0439	ҙ	0497
К	043A	ҙ	0499
л	043B	ҝ	049B
m	043C	ҟ	049F
н	043D	ҡ	04A1
o	043E	ң	04A3
p	043F	ҥ	04A5
p	0440	ҩ	04A9
c	0441	ҫ	04AB
t	0442	ҭ	04AD
y	0443	ү	04AF
÷	0444	ұ	04B1
x	0445	ҳ	04B3
ц	0446	ҵ	04B5
ч	0447	ҷ	04B7
(2)	0448	h	04BB
щ	0449	ҽ	04BD
ъ	044A	ҿ	04BF

Karakter	Kode	Karakter	Kode
ы	044B	#	04CA
ь	044C	ă	04D1
э	044D	ä	04D3
ю	044E	ĕ	04H7
я	044F	ə	04H9
#	0450	ʒ	04E1
ё	0451	ӣ	04E3
ђ	0452	ö	04E7
ѓ	0453	ө	04E9
c	0454	ӯ	04EF
s	0455	ӱ	04F1
i	0456	ӳ	04F3
c	0457	#	04F7
j	0458	ӹ	04F9
#	0525		

Set karakter Afrikaans

Untuk kosakata kustom Afrikaans, Anda dapat menggunakan karakter berikut di lapangan: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

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

Set karakter Arab

Untuk kosakata kustom Arab, Anda dapat menggunakan karakter Unicode berikut di lapangan. `Phrase` Anda juga dapat menggunakan karakter tanda hubung (-) untuk memisahkan kata-kata.

Karakter	Kode	Karakter	Kode
ء	0621	س	0633
4	0622	ش	0634
أ	0623	ص	0635
ؤ	0624	ض	0636
إ	0625	ط	0637
ئ	0626	ظ	0638
ا	0627	ع	0639
م	0628	غ	063A

Karakter	Kode	Karakter	Kode
س	0629	ف	0641
ت	062A	ق	0642
ث	062B	ك	0643
ج	062C	ل	0644
ح	062D	م	0645
خ	062E	م	0646
د	062F	ه	0647
ذ	0630	و	0648
ر	0631	ى	0649
4	0632	1	064A

Set karakter Asturian

Untuk kosakata khusus Asturian, Anda dapat menggunakan karakter berikut di lapangan: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
á	00E1	ñ	00F1
é	00E9	ó	00F3
í	00ED	ú	00FA

Karakter	Kode	Karakter	Kode
ü	00FC		

Set karakter Azerbaijan

Untuk kosakata kustom Azerbaijan, Anda dapat menggunakan karakter berikut di lapangan: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
ä	00E4	ğ	011F
ç	00E7	ı	0131
ö	00F6	ş	015F
ü	00FC	ə	0259
◌̇	0307		

Set karakter Armenia

Untuk kosakata kustom Armenia, Anda dapat menggunakan karakter berikut di lapangan: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
ա	0561	ն	0574
բ	0562	շ	0575
գ	0563	ո	0576
դ	0564	չ	0577
ե	0565	պ	0578
զ	0566	ջ	0579
է	0567	ռ	057A
ը	0568	ս	057B
թ	0569	վ	057C
ժ	056A	տ	057D
ի	056B	ր	057E
լ	056C	ց	057F
խ	056D	ւ	0580
ծ	056E	փ	0581
կ	056F	ք	0582
հ	0570	և	0583
ձ	0571	ֆ	0584
ղ	0572	օ	0585
ճ	0573	֏	0586

Set karakter Bashkir

Untuk kosakata khusus Bashkir, Anda dapat menggunakan karakter berikut di bidang: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
а	0430	љ	0459
b	0431	њ	045A
в	0432	ћ	045B
г	0433	(b)	045C
д	0434	#	045D
е	0435	ў	045E
ж	0436	џ	045F
з	0437	ґ	0491
и	0438	ғ	0493
в	0439	җ	0497
К	043A	ҙ	0499
л	043B	қ	049B
m	043C	ҝ	049F
н	043D	ҡ	04A1

Karakter	Kode	Karakter	Kode
о	043E	ӈ	04A3
р	043F	ҥ	04A5
р	0440	ҩ	04A9
с	0441	ҫ	04AB
т	0442	ҭ	04AD
у	0443	ү	04AF
÷	0444	ұ	04B1
х	0445	ҳ	04B3
ц	0446	ҵ	04B5
ч	0447	ҷ	04B7
÷	0448	h	04BB
щ	0449	ҽ	04BD
ъ	044A	ҿ	04BF
ы	044B	#	04CA
ь	044C	ӑ	04D1
э	044D	ӓ	04D3
ю	044E	ӗ	04H7
я	044F	ә	04H9
#	0450	ӡ	04E1
g	0451	ӣ	04E3

Karakter	Kode	Karakter	Kode
ђ	0452	ӧ	04E7
ѓ	0453	ө	04E9
c	0454	ӯ	04EF
s	0455	ӱ	04F1
i	0456	ӳ	04F3
c	0457	#	04F7
j	0458	ӹ	04F9

Set karakter Basque

Untuk kosakata kustom Basque, Anda dapat menggunakan karakter berikut di lapangan: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
á	00E1	ñ	00F1
é	00E9	ó	00F3
í	00ED	ú	00FA
ü	00FC		

Set karakter Belarusia

Untuk kosakata khusus Belarusia, Anda dapat menggunakan karakter berikut di lapangan: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
а	0430	с	0441
b	0431	т	0442
в	0432	у	0443
г	0433	ў	0444
(c)	0434	х	0445
е	0435	ц	0446
ж	0436	ч	0447
(b)	0437	(2)	0448
в	0439	ы	044B
К	043A	ь	044C
л	043B	э	044D
m	043C	ю	044E
н	043D	я	044F
о	043E	g	0451

Karakter	Kode	Karakter	Kode
p	043F	i	0456
p	0440	ў	045E

Set karakter Bengali

Untuk kosakata kustom Bengali, Anda dapat menggunakan karakter berikut di bidang: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
ঁ	0981	দ	09A6
ং	0982	ধ	09A7
ঃ	0983	ন	09A8
অ	0985	প	09AA
আ	0986	ফ	09AB
ই	0987	ব	09AC
ঈ	0988	ভ	09AD
উ	0989	ম	09AE
ঊ	098A	য	09AF
ঋ	098B	র	09B0
এ	098F	ল	09B2

Karakter	Kode	Karakter	Kode
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
ঞ	099E	ৄ	09C4
ট	099F	ে	09C7
ঠ	09A0	ৈ	09C8
ড	09A1	ো	09CB
ঢ	09A2	ৌ	09CC
ণ	09A3	্	09CD
ত	09A4	#	09CE
থ	09A5	ৗ	09D7

Set karakter Bosnia

Untuk kosakata kustom Bosnia, Anda dapat menggunakan karakter berikut di bidang: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
ć	0107	đ	0111
č	010D	š	0161
ž	017E		

Set karakter Bulgaria

Untuk kosakata kustom Bulgaria, Anda dapat menggunakan karakter berikut di lapangan: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
а	0430	р	043F
б	0431	р	0440
в	0432	с	0441
г	0433	т	0442

Karakter	Kode	Karakter	Kode
д	0434	у	0443
е	0435	÷	0444
ж	0436	х	0445
з	0437	ц	0446
и	0438	ч	0447
с	0439	⋙	0448
К	043A	щ	0449
л	043B	ъ	044A
m	043C	ь	044C
н	043D	ю	044E
о	043E	я	044F

Set karakter Catalan

Untuk kosakata kustom Catalan, Anda dapat menggunakan karakter berikut di bidang: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
à	00E0	ï	00EF
ç	00E7	ò	00F2

Karakter	Kode	Karakter	Kode
è	00E8	ó	00F3
é	00E9	ú	00FA
í	00ED	ü	00FC
l·	0140		

Set karakter Kurdi Tengah

Untuk kosakata kustom Kurdi Tengah, Anda dapat menggunakan karakter berikut di lapangan: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
ئ	0626	م	0645
ا	0627	م	0646
م	0628	و	0648
ت	062A	پ	067E
ج	062C	چ	0686
ح	062D	ڕ	0695
خ	062E	ژ	0698
د	062F	ڤ	06A4

Karakter	Kode	Karakter	Kode
ر	0631	ک	06A9
4	0632	گ	06AF
س	0633	ڵ	06B5
ش	0634	ھ	06BE
ع	0639	ۆ	06C6
غ	063A	ۇ	06C7
ف	0641	4	06CC
ق	0642	ێ	06CE
ل	0644	ه	06D5

Tionghoa, Mandarin (Tiongkok Daratan), Set karakter yang disederhanakan

Untuk kosakata khusus bahasa Mandarin (Sederhana), `Phrase` bidang ini dapat menggunakan salah satu karakter yang tercantum dalam file berikut:

- zh-cn-character-set

`SoundsLike`Bidang dapat berisi suku kata pinyin yang tercantum dalam file berikut:

- pinyin-character-set

Saat Anda menggunakan suku kata pinyin di `SoundsLike` lapangan, pisahkan suku kata dengan tanda hubung (-).

Amazon Transcribe mewakili empat nada dalam bahasa Mandarin (Sederhana) menggunakan angka. Tabel berikut menunjukkan bagaimana tanda nada dipetakan untuk kata 'ma'.

Nada	Tanda nada	Nomor nada
Nada 1	mā	ma1
Nada 2	má	ma2
Nada 3	m	ma3
Nada 4	mà	ma4

> **Note**
>
> Untuk nada ke-5 (netral), Anda dapat menggunakan Tone 1, dengan pengecualian 'er', yang harus dipetakan ke Tone 2. Misalnya, akan direpresentasikan sebagai 'da3-zhuan4-er2'.

Kosakata kustom bahasa Mandarin (Sederhana) tidak menggunakan `IPA` bidang ini, tetapi Anda tetap harus menyertakan `IPA` header dalam tabel kosakata khusus.

Contoh berikut adalah file input dalam format teks. Contoh menggunakan spasi untuk menyelaraskan kolom. File input Anda harus menggunakan karakter TAB untuk memisahkan kolom. Sertakan spasi hanya di `DisplayAs` kolom.

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

Tionghoa, Mandarin (Taiwan), Set karakter tradisional

Untuk kosakata khusus bahasa Mandarin (Tradisional), `Phrase` bidang ini dapat menggunakan salah satu karakter yang tercantum dalam file berikut:

- zh-tw-character-set

`SoundsLike`Bidang dapat berisi suku kata zhuyin yang tercantum dalam file berikut:

- zhuyin-character-set

Saat Anda menggunakan suku kata zhuyin di `SoundsLike` lapangan, pisahkan suku kata dengan tanda hubung (-).

Amazon Transcribe mewakili empat nada dalam bahasa Mandarin (Tradisional) menggunakan angka. Tabel berikut menunjukkan bagaimana tanda nada dipetakan untuk kata.

Nada	Tanda nada
Nada 1	ㄇㄚ
Nada 2	ㄇㄚˊ
Nada 3	ㄇㄚˇ
Nada 4	ㄇㄚˋ

Kosakata kustom bahasa Mandarin (Tradisional) tidak menggunakan `IPA` bidang ini, tetapi Anda tetap harus menyertakan `IPA` tajuk dalam tabel kosakata khusus.

Contoh berikut adalah file input dalam format teks. Contoh menggunakan spasi untuk menyelaraskan kolom. File input Anda harus menggunakan karakter TAB untuk memisahkan kolom. Sertakan spasi hanya di `DisplayAs` kolom.

```
Phrase        SoundsLike                    IPA        DisplayAs
##           ###ˋ-##
##           ###ˇ-##ˊ
####        ###ˊ-##ˊ-##ˋ-##ˇ
#####     #ˋ-###ˋ-##ˊ-##ˇ-###ˋ            ###
```

Tionghoa, Kanton (Hong Kong), Set karakter tradisional

Untuk kosakata khusus bahasa Mandarin (Kanton) Hong Kong, `Phrase` bidang ini dapat menggunakan salah satu karakter yang tercantum dalam file berikut:

- zh-hk-character-set

Set karakter Kroasia

Untuk kosakata kustom Kroasia, Anda dapat menggunakan karakter berikut di lapangan: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
ć	0107	đ	0111
č	010D	š	0161
ž	017E		

Set karakter Ceko

Untuk kosakata kustom Ceko, Anda dapat menggunakan karakter berikut di `Phrase` lapangan:

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
á	00E1	ď	010F
é	00E9	ě	011B
í	00ED	ň	0148
ó	00F3	ř	0159

Karakter	Kode	Karakter	Kode
ú	00FA	š	0161
ý	00FD	ť	0165
č	010D	ů	016F
ž	017E		

Set karakter Denmark

Untuk kosakata kustom Denmark, Anda dapat menggunakan karakter berikut di `Phrase` lapangan:

- a - z
- A - Z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
Å	00C5	æ	00E6
Æ	00C6	é	00E9
Ø	00D8	ø	00F8
å	00E5		

Set karakter Belanda

Untuk kosakata khusus bahasa Belanda, Anda dapat menggunakan karakter berikut di `Phrase` lapangan:

- a - z

- A - Z
- '(apostrof)
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

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

Set karakter bahasa Inggris

Untuk kosakata kustom bahasa Inggris, Anda dapat menggunakan karakter berikut di `Phrase` lapangan:

- a - z
- A - Z

- '(apostrof)
- - (tanda hubung)
- . (periode)

Set karakter Estonia

Untuk kosakata kustom Estonia, Anda dapat menggunakan karakter berikut di lapangan: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
ä	00E4	ü	00FC
õ	00F5	š	0161
ö	00F6	ž	017E

Set karakter Farsi

Untuk kosakata kustom Farsi, Anda dapat menggunakan karakter berikut di lapangan. `Phrase`

Karakter	Kode	Karakter	Kode
ء	0621	ظ	0638
4	0622	ع	0639
أ	0623	غ	063A
ؤ	0624	ف	0641
ئ	0626	ق	0642

Karakter	Kode	Karakter	Kode
ا	0627	ل	0644
م	0628	م	0645
ت	062A	م	0646
ث	062B	ه	0647
ج	062C	و	0648
ح	062D	ـَ	064E
خ	062E	ـُ	064F
د	062F	ـِ	0650
ذ	0630	ـّ	0651
ر	0631	پ	067E
4	0632	چ	0686
س	0633	ژ	0698
ش	0634	ک	06A9
ص	0635	گ	06AF
ض	0636	4	06CC
ط	0637		

Set karakter Finlandia

Untuk kosakata kustom Finlandia, Anda dapat menggunakan karakter berikut di lapangan: `Phrase`

- a - z
- - (tanda hubung)

- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
ä	00E4	ö	00F6
å	00E5	š	0161
ž	017E		

Set karakter Prancis

Untuk kosakata khusus bahasa Prancis, Anda dapat menggunakan karakter berikut di `Phrase` lapangan:

- a - z
- A - Z
- '(apostrof)
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
À	00C0	à	00E0
Â	00C2	â	00E2
Ç	00C7	ç	00E7
È	00C8	è	00E8
É	00C9	é	00E9

Karakter	Kode	Karakter	Kode
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
DAN	00DB	dan	00FB
Ü	00DC	ü	00FC

Set karakter Galicia

Untuk kosakata kustom Galicia, Anda dapat menggunakan karakter berikut di lapangan: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
á	00E1	ñ	00F1
é	00E9	ó	00F3
í	00ED	ú	00FA
ü	00FC		

Set karakter Georgia

Untuk kosakata khusus Georgia, Anda dapat menggunakan karakter berikut di lapangan: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

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

Karakter	Kode	Karakter	Kode
ƺ	10DE	ƾ	10EE
ƻ	10DF	ƿ	10EF
ƽ	10F0		

Set karakter Jerman

Untuk kosakata khusus Jerman, Anda dapat menggunakan karakter berikut di `Phrase` lapangan:

- a - z
- A - Z
- '(apostrof)
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
ä	00E4	Ä	00C4
ö	00F6	Ö	00D6
ü	00FC	Ü	00DC
ß	00DF		

Set karakter Yunani

Untuk kosakata kustom Yunani, Anda dapat menggunakan karakter berikut di `Phrase` lapangan:

- a - z
- - (tanda hubung)

- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
ά	03AC	ν	03BD
έ	03AD	ξ	03MENJADI
?	03AE	??	03BF
ί	03AF	π	03C0
ΰ	03B0	ρ	03C1
α	03B1	ς	03C2
β	03B2	σ	03C3
γ	03B3	τ	03C4
δ	03B4	?	03C5
ε	03B5	φ	03C6
θ	03B6	χ	03C7
η	03B7	ψ	03C8
θ	03B8	ω	03C9
ι	03B9	ϊ	03CA
κ	03BA	ϋ	03CB
λ	03BB	ό	03CC
μ	03BC	11.	03CE
ΐ	0390		

Set karakter Gujarati

Untuk kosakata kustom Gujarati, Anda dapat menggunakan karakter berikut di bidang: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
ઁ	0A81	દ	0 AA6
ં	0A82	ધ	0 AA7
ઃ	0A83	ન	0 AA8
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
ઍ	0A8D	લ	0 AB2
એ	0A8F	ળ	0 AB3
ઐ	0A90	વ	0 AB5
ઑ	0A91	શ	0 AB6

Karakter	Kode	Karakter	Kode
ઓ	0A93	ષ	0 AB7
ઔ	0A94	સ	0 AB8
ક	0A95	હ	0 AB9
ખ	0A96	઼	0ABC
ગ	0A97	ા	0ABE
ઘ	0A98	િ	0ABF
ઙ	0A99	ી	0AC0
ચ	0A9A	ુ	0 AC1
છ	0A9B	ૂ	0 AC2
જ	0A9C	ૃ	0 AC3
ઝ	0A9D	ૅ	0 AC5
ઞ	0A9E	ે	0 AC7
ટ	0A9F	ૈ	0 AC8
ઠ	0AA0	ૉ	0 AC9
ડ	0 AA1	ો	0ACB
ઢ	0 AA2	ૌ	0ACC
ણ	0 AA3	્	0ACD
ત	0 AA4	ૐ	0AD0
થ	0 AA5	ૠ	0AE0

Set karakter Hausa

Untuk kosakata kustom Hausa, Anda dapat menggunakan karakter berikut di bidang: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
ƙ	0199	ɓ	0253
ƴ	01B4	ɗ	0257
̃	0303		

Set karakter Ibrani

Untuk kosakata kustom Ibrani, Anda dapat menggunakan karakter Unicode berikut di bidang: `Phrase`

Karakter	Kode	Karakter	Kode
-	002D	4	05DD
4	05D0	ם	05DE
2	05D1	ן	05DF
ג	05D2	נ	05E0
4	05D3	ס	05E1
ה	05D4	ע	05E2
4	05D5	ף	05E3

Karakter	Kode	Karakter	Kode
ר	05D6	פ	05E4
ח	05D7	ץ	05E5
ט	05D8	צ	05E6
י	05D9	ק	05E7
ך	05DA	4	05E8
כ	05DB	ש	05E9
1	05DC	ת	05EA

Set karakter Hindi

Untuk kosakata kustom Hindi, Anda dapat menggunakan karakter Unicode berikut di bidang: `Phrase`

Karakter	Kode	Karakter	Kode
-	002D	थ	0925
.	002E	द	0926
ँ	0901	ध	0927
ं	0902	न	0928
ः	0903	प	092A
अ	0905	फ	092B
आ	0906	ब	092C
इ	0907	भ	092D
ई	0908	म	092E

Karakter	Kode	Karakter	Kode
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	ा	093E
ख	0916	ि	093F
ग	0917	ी	0940
घ	0918	ु	0941
ङ	0919	ू	0942
च	091A	ृ	0943
छ	091B	ॅ	0945
ज	091C	ॉ	0947
झ	091D	ॊ	0948
ञ	091E	ॉ	0949
ट	091F	ो	094B
ठ	0920	ौ	094C

Karakter	Kode	Karakter	Kode
ड	0921	़	094D
ढ	0922	ज़	095B
ण	0923	ड़	095C
त	0924	ढ़	095D

Amazon Transcribe memetakan karakter berikut:

Karakter	Dipetakan ke
(0929)	(0928)
(0931)	(0930)
(0958)	(0915)
(0959)	(0916)
(095A)	(0917)
(095E)	(092B)
(095F)	(092F)

Set karakter Hungaria

Untuk kosakata khusus Hongaria, Anda dapat menggunakan karakter berikut di lapangan: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

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

Set karakter Islandia

Untuk kosakata kustom Islandia, Anda dapat menggunakan karakter berikut di lapangan: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
á	00E1	ú	00FA
é	00E9	ý	00FD
ð	00F0	?	00FE
í	00ED	æ	00E6
ó	00F3	ö	00F6

Set karakter Indonesia

Untuk kosakata kustom bahasa Indonesia, Anda dapat menggunakan karakter berikut di `Phrase` lapangan:

- a - z
- A - Z
- '(apostrof)
- - (tanda hubung)
- . (periode)

Set karakter Italia

Untuk kosakata khusus Italia, Anda dapat menggunakan karakter berikut di `Phrase` lapangan:

- a - z
- A - Z
- '(apostrof)
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
À	00C0	à	00E0
Ä	00C4	ä	00E4
Ç	00C7	ç	00E7
È	00C8	è	00E8
É	00C9	é	00E9
Ê	00CA	ê	00EA

Karakter	Kode	Karakter	Kode
Ë	00CB	ë	00EB
Ì	00CC	ì	00EC
Ò	00D2	ò	00F2
Ù	00D9	ù	00F9
Ü	00DC	ü	00FC

Set karakter Jepang

Untuk kosakata kustom Jepang, `DisplayAs` bidang ini mendukung semua karakter hiragana, katakana, dan kanji, dan huruf kapital romaji fullwidth.

`Phrase`Bidang mendukung karakter yang tercantum dalam file berikut:

- ja-jp-character-set

Set karakter Kabyle

Untuk kosakata kustom Kabyle, Anda dapat menggunakan karakter berikut di bidang: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
ï	00EF	ḍ	1E0D
č	010D	ḥ	1E25
ř	0159	ṛ	1E5B

Karakter	Kode	Karakter	Kode
ǧ	01E7	ṣ	1E63
ɛ	025B	ṭ	1E6D
ɣ	0263	ẓ	1E93

Set karakter Kannada

Untuk kosakata kustom Kannada, Anda dapat menggunakan karakter berikut di bidang: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
ಂ	0C82	ಧ	0 CA7
ಃ	0C83	ನ	0 CA8
ಅ	0C85	ಪ	0CAA
ಆ	0C86	ಫ	0TAKSI
ಇ	0C87	ಬ	0CAC
ಈ	0C88	ಭ	0CAD
ಉ	0C89	ಮ	0CAE
ಊ	0C8A	ಯ	0CAF
ಋ	0C8B	ರ	0CB0
ಎ	0C8E	ಲ	0 CB2

Karakter	Kode	Karakter	Kode
ಏ	0C8F	ಳ	0 CB3
ಐ	0C90	ವ	0 CB5
ಒ	0C92	ಶ	0 CB6
ಓ	0C93	ಷ	0 CB7
ಔ	0C94	ಸ	0 CB8
ಕ	0C95	ಹ	0 CB9
ಖ	0C96	#	0CBC
ಗ	0C97	#	0CBD
ಘ	0C98	ಾ	0CBE
ಙ	0C99	ಿ	0CBF
ಚ	0C9A	ೀ	0CC0
ಛ	0C9B	ು	0 CC1
ಜ	0C9C	ೂ	0 CC2
ಝ	0C9D	ೃ	0 CC3
ಞ	0C9E	ೆ	0 CC6
ಟ	0C9F	ೇ	0 CC7
ಠ	0CA0	ೈ	0 CC8
ಡ	0 CA1	ೊ	0CCA
ಢ	0 CA2	ೋ	0CCB
ಣ	0 CA3	್	0CCC

Karakter	Kode	Karakter	Kode
ತ	0 CA4	್	0CCD
ಥ	0 CA5	ೀ	0 CD5
ದ	0 CA6	ೆ	0 CD6
ಖಾ	0CE0		

Set karakter Kazakh

Untuk kosakata khusus Kazakh, Anda dapat menggunakan karakter berikut di lapangan: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
t	0442	ы	044B
b	0431	я	044F
o	043E	c	0441
p	043F	h	04BB
□	0448	d	0434
и	0438	p	0440
ч	0447	г	0433
н	043D	g	0451
қ	049B	в	0439

Karakter	Kode	Karakter	Kode
і	0456	ө	04E9
щ	0449	в	0432
е	0435	э	044D
ә	04H9	ң	04A3
ю	044E	л	043B
÷	0437	□	0444
х	0445	К	043A
ц	0446	у	0443
ү	04AF	ж	0436
m	043C	ғ	0493
ь	044C	а	0430
ъ	044A	ұ	04B1

Set karakter Kinyarwanda

Untuk kosakata kustom Kinyarwanda, Anda dapat menggunakan karakter berikut di bidang: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
á	00E1	ó	00F3

Karakter	Kode	Karakter	Kode
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

Set karakter Korea

Untuk kosakata kustom Korea, Anda dapat menggunakan salah satu suku kata Hangul di lapangan. `Phrase` Untuk informasi lebih lanjut, lihat Suku Kata Hangul di Wikipedia.

Set karakter Kirgistan

Untuk kosakata khusus Kirgistan, Anda dapat menggunakan karakter berikut di lapangan: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
а	0430	љ	0459

Karakter	Kode	Karakter	Kode
b	0431	њ	045A
в	0432	ћ	045B
г	0433	(b)	045C
д	0434	#	045D
е	0435	ў	045E
ж	0436	џ	045F
з	0437	ґ	0491
и	0438	ғ	0493
в	0439	ҷ	0497
К	043A	ҙ	0499
л	043B	қ	049B
m	043C	ҝ	049F
н	043D	ҡ	04A1
о	043E	ң	04A3
р	043F	ҥ	04A5
р	0440	ҩ	04A9
с	0441	ҫ	04AB
t	0442	ҭ	04AD
у	0443	ү	04AF
⋙	0444	ұ	04B1

Karakter	Kode	Karakter	Kode
х	0445	ҳ	04B3
ц	0446	ҵ	04B5
ч	0447	ҷ	04B7
÷	0448	h	04BB
щ	0449	ҽ	04BD
ъ	044A	ҿ	04BF
ы	044B	#	04CA
ь	044C	ӑ	04D1
э	044D	ӓ	04D3
ю	044E	ӗ	04H7
я	044F	ә	04H9
#	0450	ӡ	04E1
g	0451	ӣ	04E3
ђ	0452	ӧ	04E7
ѓ	0453	ө	04E9
c	0454	ӯ	04EF
s	0455	ӱ	04F1
i	0456	ӳ	04F3
c	0457	#	04F7
j	0458	ӹ	04F9

Set karakter Latvia

Untuk kosakata kustom Latvia, Anda dapat menggunakan karakter berikut di bidang: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

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

Set karakter Lithuania

Untuk kosakata khusus Lituania, Anda dapat menggunakan karakter berikut di lapangan: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
ą	0105	į	012F

Karakter	Kode	Karakter	Kode
č	010D	š	0161
ę	0119	ų	0173
ė	0117	ū	016B
ž	017E		

Set karakter Luganda

Untuk kosakata kustom Luganda, Anda dapat menggunakan karakter berikut di bidang: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
??	00FF	ŋ	014B

Set karakter Makedonia

Untuk kosakata kustom Makedonia, Anda dapat menggunakan karakter berikut di lapangan: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
а	0430	љ	0459
b	0431	њ	045A
в	0432	ћ	045B
г	0433	(b)	045C
д	0434	#	045D
е	0435	ў	045E
ж	0436	џ	045F
з	0437	ґ	0491
и	0438	ғ	0493
с	0439	җ	0497
К	043A	ҙ	0499
л	043B	қ	049B
m	043C	ҝ	049F
н	043D	ҡ	04A1
о	043E	ң	04A3
р	043F	ҥ	04A5
р	0440	ҩ	04A9
с	0441	ҫ	04AB
n	0442	ҭ	04AD
у	0443	ү	04AF

Karakter	Kode	Karakter	Kode
Ф	0444	ұ	04B1
х	0445	ҳ	04B3
ц	0446	ҵ	04B5
ч	0447	ҷ	04B7
(2)	0448	h	04BB
щ	0449	ҽ	04BD
ъ	044A	ҿ	04BF
ы	044B	#	04CA
ь	044C	ӑ	04D1
э	044D	ӓ	04D3
ю	044E	ӗ	04H7
я	044F	ә	04H9
#	0450	ӡ	04E1
ё	0451	ӣ	04E3
ђ	0452	ӧ	04E7
ѓ	0453	ө	04E9
c	0454	ӯ	04EF
s	0455	ӱ	04F1
i	0456	ӳ	04F3
c	0457	#	04F7

Karakter	Kode	Karakter	Kode
j	0458	ӹ	04F9

Set karakter Melayu

Untuk kosakata kustom Melayu, Anda dapat menggunakan karakter berikut di `Phrase` lapangan:

- a - z
- A - Z
- '(apostrof)
- - (tanda hubung)
- . (periode)

Set karakter Malayalam

Untuk kosakata kustom Malayalam, Anda dapat menggunakan karakter berikut di lapangan: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
ം	0D02	ന	0D28
ഃ	0D03	പ	0D2A
അ	0D05	ഫ	0D2B
ആ	0D06	ബ	0D2C
ഇ	0D07	ഭ	0D2D

Karakter	Kode	Karakter	Kode
ഈ	0D08	മ	0D2E
ഉ	0D09	യ	0D2F
ഊ	0D0A	ര	0D30
ഋ	0D0B	റ	0D31
എ	0D0E	ല	0D32
ഏ	0D0F	ള	0D33
ഐ	0D10	ഴ	0D34
ഒ	0D12	വ	0D35
ഓ	0D13	ശ	0D36
ഔ	0D14	ഷ	0D37
ക	0D15	സ	0D38
ഖ	0D16	ഹ	0D39
ഗ	0D17	ാ	0D3E
ഘ	0D18	ി	0D3F
ങ	0D19	ീ	0D40
ച	0D1A	ു	0D41
ഛ	0D1B	ൂ	0D42
ജ	0D1C	ൃ	0D43
ഝ	0D1D	െ	0D46
ഞ	0D1E	േ	0D47

Karakter	Kode	Karakter	Kode
ടു	0D1F	ൈ	0D48
ഠ	0D20	ൊ	0D4A
ഡ	0D21	ോ	0D4B
ഢ	0D22	ൌ	0D4C
ണ	0D23	്	0D4D
ത	0D24	#	0D7A
ഥ	0D25	#	0D7B
ദ	0D26	#	0D7C
ധ	0D27	#	0D7D

Set karakter Malta

Untuk kosakata kustom Malta, Anda dapat menggunakan karakter berikut di lapangan: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
à	00E0	ù	00F9
è	00E8	4	010B
ì	00EC	ġ	0121
ò	00F2	ħ	0127

Karakter	Kode	Karakter	Kode
ż	017C		

Set karakter Marathi

Untuk kosakata kustom Marathi, Anda dapat menggunakan karakter berikut di bidang: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
ँ	0901	थ	0925
ं	0902	द	0926
ः	0903	ध	0927
अ	0905	न	0928
आ	0906	प	092A
इ	0907	फ	092B
ई	0908	ब	092C
उ	0909	भ	092D
ऊ	090A	म	092E
ऋ	090B	य	092F
ऍ	090D	र	0930
ए	090F	ल	0932

Karakter	Kode	Karakter	Kode
ऐ	0910	ळ	0933
ऑ	0911	व	0935
ओ	0913	श	0936
औ	0914	ष	0937
क	0915	स	0938
ख	0916	ह	0939
ग	0917	़	093C
घ	0918	ा	093E
ङ	0919	ि	093F
च	091A	ी	0940
छ	091B	ु	0941
ज	091C	ू	0942
झ	091D	ृ	0943
ञ	091E	ॅ	0945
ट	091F	ॉ	0947
ठ	0920	ॊ	0948
ड	0921	ॅ	0949
ढ	0922	ो	094B
ण	0923	ौ	094C
त	0924	्	094D

Karakter	Kode	Karakter	Kode
ॐ	0950		

Set karakter Meadow Mari

Untuk kosakata khusus Meadow Mari, Anda dapat menggunakan karakter berikut di lapangan: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
а	0430	љ	0459
b	0431	њ	045A
в	0432	ћ	045B
г	0433	(b)	045C
d	0434	#	045D
е	0435	ў	045E
ж	0436	џ	045F
з	0437	ґ	0491
и	0438	ғ	0493
в	0439	ҷ	0497
К	043A	ҙ	0499

Karakter	Kode	Karakter	Kode
л	043B	қ	049B
m	043C	ҝ	049F
н	043D	ҡ	04A1
o	043E	ң	04A3
p	043F	ҥ	04A5
p	0440	ҩ	04A9
c	0441	ҫ	04AB
tk	0442	ҭ	04AD
y	0443	ү	04AF
÷	0444	ұ	04B1
x	0445	ҳ	04B3
ц	0446	ҵ	04B5
ч	0447	ҷ	04B7
÷	0448	h	04BB
щ	0449	ҽ	04BD
ъ	044A	ҿ	04BF
ы	044B	#	04CA
ь	044C	ӑ	04D1
э	044D	ӓ	04D3
ю	044E	ӗ	04H7

Karakter	Kode	Karakter	Kode
я	044F	ә	04H9
#	0450	ӡ	04E1
ё	0451	ӣ	04E3
ђ	0452	ӧ	04E7
ѓ	0453	ө	04E9
c	0454	ӯ	04EF
s	0455	ӱ	04F1
i	0456	ӳ	04F3
c	0457	#	04F7
j	0458	ӹ	04F9

Set karakter Mongolia

Untuk kosakata kustom Mongolia, Anda dapat menggunakan karakter berikut di lapangan: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
a	0430	љ	0459
b	0431	њ	045A
в	0432	ћ	045B

Karakter	Kode	Karakter	Kode
г	0433	(b)	045C
д	0434	#	045D
е	0435	ў	045E
ж	0436	џ	045F
з	0437	ґ	0491
и	0438	ғ	0493
в	0439	җ	0497
К	043A	ҙ	0499
л	043B	қ	049B
m	043C	ҝ	049F
н	043D	ҡ	04A1
о	043E	ң	04A3
р	043F	ҥ	04A5
р	0440	ҩ	04A9
с	0441	ҫ	04AB
t	0442	ҭ	04AD
у	0443	ү	04AF
÷	0444	ұ	04B1
х	0445	ҳ	04B3
ц	0446	ҵ	04B5

Karakter	Kode	Karakter	Kode
ч	0447	ҷ	04B7
÷	0448	h	04BB
щ	0449	ҽ	04BD
ъ	044A	ҿ	04BF
ы	044B	#	04CA
ь	044C	ӑ	04D1
э	044D	ӓ	04D3
ю	044E	ӗ	04H7
я	044F	ә	04H9
#	0450	ӡ	04E1
ё	0451	ӣ	04E3
ђ	0452	ӧ	04E7
ѓ	0453	ө	04E9
ї	0454	ӯ	04EF
s	0455	ӱ	04F1
і	0456	ӳ	04F3
ї	0457	#	04F7
ј	0458	ӹ	04F9

Set karakter Bokmål Norwegia

Untuk kosakata kustom Bokmål Norwegia, Anda dapat menggunakan karakter berikut di lapangan: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
å	00E5	æ	00E6
ø	00F8		

Set karakter Odia/Oriya

Untuk kosakata khusus Odia/Oriya, Anda dapat menggunakan karakter berikut di lapangan: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
◌	0B01	ଦ	0B26
ଂ	0B02	ଧ	0B27
ଃ	0B03	ନ	0B28
ଅ	0B05	ପ	0B2A

Karakter	Kode	Karakter	Kode
ଆ	0B06	ଫ	0B2B
ଇ	0B07	ବ	0B2C
ଈ	0B08	ଭ	0B2D
ଉ	0B09	ମ	0B2E
ଊ	0B0A	ଯ	0B2F
ଋ	0B0B	ର	0B30
ଏ	0B0F	ଲ	0B32
ଐ	0B10	ଳ	0B33
ଓ	0B13	ଶ	0B36
ଔ	0B14	ଷ	0B37
କ	0B15	ସ	0B38
ଖ	0B16	ହ	0B39
ଗ	0B17	଼	0B3C
ଘ	0B18	ା	0B3E
ଙ	0B19	ି	0B3F
ଚ	0B1A	ୀ	0B40
ଛ	0B1B	ୁ	0B41
ଜ	0B1C	ୂ	0B42
ଝ	0B1D	ୃ	0B43
ଞ	0B1E	େ	0B47

Karakter	Kode	Karakter	Kode
ଟ	0B1F	ୈ	0B48
ଠ	0B20	ୋ#	0B4B
ଡ	0B21	ୌ	0B4C
ଢ	0B22	୍	0B4D
ଣ	0B23	୍	0B56
ତ	0B24	ୟ	0B5F
ଥ	0B25	ୠ	0B60
#	0B71		

Set karakter Pashto

Untuk kosakata kustom Pashto, Anda dapat menggunakan karakter berikut di lapangan: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
4	0622	و	0648
أ	0623	1	064A
ؤ	0624	ً	064B
ئ	0626	ٌ	064C
ا	0627	ٍ	064D

Karakter	Kode	Karakter	Kode
م	0628	ـَ	064E
ت	062A	ـُ	064F
ث	062B	ـِ	0650
ج	062C	ـّ	0651
ح	062D	ـْ	0652
خ	062E	#	0654
د	062F	ٱ	0670
ذ	0630	ټ	067C
ر	0631	پ	067E
4	0632	ځ	0681
س	0633	څ	0685
ش	0634	چ	0686
ص	0635	ډ	0689
ض	0636	ړ	0693
ط	0637	ږ	0696
ظ	0638	ژ	0698
ع	0639	ښ	069A
غ	063A	ک	06A9
ف	0641	ګ	06AB
ق	0642	گ	06AF

Karakter	Kode	Karakter	Kode
ل	0644	ں	06BC
م	0645	ھ	06CC
م	0646	ی	06CD
ہ	0647	ے	06D0

Set karakter Polandia

Untuk kosakata kustom Polandia, Anda dapat menggunakan karakter berikut di `Phrase` lapangan:

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

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

Set karakter Portugis

Untuk kosakata kustom Portugis, Anda dapat menggunakan karakter berikut di `Phrase` lapangan:

- a - z

- A - Z
- '(apostrof)
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

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

Karakter	Kode	Karakter	Kode
Ú	00DA	ú	00FA
Ü	00DC	ü	00FC

Set karakter Punjabi

Untuk kosakata kustom Punjabi, Anda dapat menggunakan karakter berikut di bidang: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
ਅ	0A05	ਧ	0A27
ਆ	0A06	ਨ	0A28
ਇ	0A07	ਪ	0A2A
ਈ	0A08	ਫ	0A2B
ਉ	0A09	ਬ	0A2C
ਊ	0A0A	ਭ	0A2D
ਏ	0A0F	ਮ	0A2E
ਐ	0A10	ਯ	0A2F
ਓ	0A13	ਰ	0A30
ਔ	0A14	ਲ	0A32
ਕ	0A15	ਵ	0A35

Karakter	Kode	Karakter	Kode
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
ਢ	0A22	੍	0A4D
ਣ	0A23	ੜ	0A5C
ਤ	0A24	ੰ	0A70
ਥ	0A25	ੱ	0A71
ਦ	0A26	ੲ	0A72
ਓ	0A73		

Set karakter Rumania

Untuk kosakata kustom Rumania, Anda dapat menggunakan karakter berikut di lapangan: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
ă	0103	#	0219
â	00E2	ţ	021B
î	00EE	ș	015F
ţ	0163		

Set karakter Rusia

Untuk kosakata khusus Rusia, Anda dapat menggunakan karakter berikut di `Phrase` lapangan:

Karakter	Kode	Karakter	Kode
'	0027	p	043F
-	002D	p	0440
.	002E	c	0441
a	0430	t	0442
b	0431	y	0443
в	0432	≡	0444
г	0433	x	0445
d	0434	ц	0446
e	0435	ч	0447

Karakter	Kode	Karakter	Kode
ж	0436	÷	0448
÷	0437	щ	0449
и	0438	ъ	044A
в	0439	ы	044B
К	043A	ь	044C
л	043B	э	044D
m	043C	ю	044E
н	043D	я	044F
о	043E	g	0451

Set karakter Serbia

Untuk kosakata kustom Serbia, Anda dapat menggunakan karakter berikut di bidang: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
ć	0107	i	0456
č	010D	c	0457
đ	0111	j	0458
š	0161	љ	0459

Karakter	Kode	Karakter	Kode
ž	017E	њ	045A
a	0430	ћ	045B
b	0431	(b)	045C
в	0432	#	045D
г	0433	ў	045E
д	0434	џ	045F
е	0435	ґ	0491
ж	0436	ғ	0493
з	0437	ҷ	0497
и	0438	ҙ	0499
в	0439	қ	049B
К	043A	ҟ	049F
л	043B	ҡ	04A1
m	043C	ң	04A3
н	043D	ҥ	04A5
o	043E	ҩ	04A9
p	043F	ҫ	04AB
p	0440	ҭ	04AD
c	0441	ү	04AF
t	0442	ұ	04B1

Karakter	Kode	Karakter	Kode
у	0443	ҳ	04B3
÷	0444	ҵ	04B5
х	0445	ҷ	04B7
ц	0446	һ	04BB
ч	0447	ҽ	04BD
÷	0448	ҿ	04BF
щ	0449	#	04CA
ъ	044A	ӑ	04D1
ы	044B	ӓ	04D3
ь	044C	ӗ	04H7
э	044D	ә	04H9
ю	044E	ӡ	04E1
я	044F	ӣ	04E3
#	0450	ӧ	04E7
ё	0451	ө	04E9
ђ	0452	ӯ	04EF
ѓ	0453	ӱ	04F1
є	0454	ӳ	04F3
s	0455	#	04F7
ӹ	04F9		

Set karakter Sinhala

Untuk kosakata kustom Sinhala, Anda dapat menggunakan karakter berikut di bidang: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
#	0D82	#	0DAF
#	0D83	#	0DB0
#	0D85	#	0 DB1
#	0D86	#	0 DB3
#	0D87	#	0 DB4
#	0D88	#	0 DB5
#	0D89	#	0 DB6
#	0D8A	#	0 DB7
#	0D8B	#	0 DB8
#	0D8C	#	0 DB9
#	0D8D	#	0DBA
#	0D91	#	0DBB
#	0D92	#	0DBD
#	0D93	#	0DC0

Karakter	Kode	Karakter	Kode
#	0D94	#	0 DC1
#	0D95	#	0 DC2
#	0D96	#	0 DC3
#	0D9A	#	0 DC4
#	0D9B	#	0 DC5
#	0D9C	#	0 DC6
#	0D9D	#	0DCA
#	0D9E	#	0DCF
#	0D9F	#	0DD0
#	0DA0	#	0 DD1
#	0 DA1	#	0 DD2
#	0 DA2	#	0 DD3
#	0 DA3	#	0 DD4
#	0 DA4	#	0 DD6
#	0 DA5	#	0 DD8
#	0 DA7	#	0 DD9
#	0 DA8	##	0DDA
#	0 DA9	#	0DDB
#	0DAA	##	0DDC
#	0DAB	###	0DDD

Karakter	Kode	Karakter	Kode
#	0DAC	##	0DDE
#	0AYAH	#	0DDF
#	0DAE	#	0 DF2

Set karakter Slovakia

Untuk kosakata kustom Slowakia, Anda dapat menggunakan karakter berikut di lapangan: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

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

Set karakter Slovenia

Untuk kosakata kustom Slovenia, Anda dapat menggunakan karakter berikut di lapangan: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
č	010D	š	0161
ž	017E		

Set karakter Somalia

Untuk kosakata kustom Somalia, Anda dapat menggunakan karakter berikut di lapangan: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
detik	0073	d	0064
t	0074	a	0061
a	0061	r	0072
n	006E	d	0064

Set karakter Spanyol

Untuk kosakata kustom Spanyol, Anda dapat menggunakan karakter berikut di `Phrase` lapangan:

- a - z
- A - Z
- '(apostrof)
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

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

Set karakter Sunda

Untuk kosakata kustom bahasa Sunda, Anda dapat menggunakan karakter berikut di lapangan: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
detik	0073	d	0064
t	0074	a	0061
a	0061	r	0072
n	006E	d	0064

Set karakter Swahili

Untuk kosakata kustom Swahili, Anda dapat menggunakan karakter berikut di bidang: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
detik	0073	d	0064
t	0074	a	0061
a	0061	r	0072
n	006E	d	0064

Set karakter Swedia

Untuk kosakata kustom Swedia, Anda dapat menggunakan karakter berikut di `Phrase` lapangan:

- a - z

- A - Z
- '(apostrof)
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
Ä	00C4	ä	00E4
Å	00C5	å	00E5
Ö	00D6	ö	00F6

Set karakter Tagalog/Filipina

Untuk kosakata khusus Tagalog/Filipina, Anda dapat menggunakan karakter berikut di bidang: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode		
ñ	00F1		

Set karakter Tamil

Untuk kosakata kustom Tamil, Anda dapat menggunakan karakter berikut di bidang: `Phrase`

Karakter	Kode	Karakter	Kode
அ	0B85	ர	0BB0
ஆ	0B86	ல	0 BB2
இ	0B87	வ	0 BB5
ஈ	0B88	ழ	0 BB4
உ	0B89	ள	0 BB3
ஊ	0B8A	ற	0 BB1
எ	0B8E	ன	0 BA9
ஏ	0B8F	ஜ	0B9C
ஐ	0B90	#	0 BB6
ஒ	0B92	ஷ	0 BB7
ஓ	0B93	ஸ	0 BB8
ஔ	0B94	ஹ	0 BB9
ஃ	0B83	◌்	0BCD
க	0B95	ா	0BBE
ங	0B99	ி	0BBF
ச	0B9A	ீ	0BC0
ஞ	0B9E	ு	0 BC1
ட	0B9F	ூ	0 BC2
ண	0 BA3	ெ	0 BC6
த	0 BA4	ே	0 BC7

Karakter	Kode	Karakter	Kode
ந	0 BA8	ன	0 BC8
ப	0BAA	ஔ	0BCA
ம	0BAE	ே	0BCB
ய	0BAF	ை	0BCC

Set karakter Tatar

Untuk kosakata kustom Tatar, Anda dapat menggunakan karakter berikut di bidang: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
а	0430	љ	0459
b	0431	њ	045A
в	0432	ћ	045B
г	0433	ќ	045C
d	0434	#	045D
е	0435	ў	045E
ж	0436	џ	045F
÷	0437	ґ	0491
и	0438	ғ	0493

Karakter	Kode	Karakter	Kode
с	0439	җ	0497
К	043A	ҙ	0499
л	043B	қ	049B
m	043C	ҝ	049F
н	043D	ҡ	04A1
о	043E	ң	04A3
р	043F	ҥ	04A5
р	0440	ҩ	04A9
с	0441	ҫ	04AB
n	0442	ҭ	04AD
у	0443	ү	04AF
Ф	0444	ұ	04B1
х	0445	ҳ	04B3
ц	0446	ҵ	04B5
ч	0447	ҷ	04B7
ш	0448	h	04BB
щ	0449	ҽ	04BD
ъ	044A	ҿ	04BF
ы	044B	#	04CA
ь	044C	ӑ	04D1

Karakter	Kode	Karakter	Kode
э	044D	ӓ	04D3
ю	044E	ӗ	04H7
я	044F	ә	04H9
#	0450	ӡ	04E1
ё	0451	ӣ	04E3
ђ	0452	ӧ	04E7
ѓ	0453	ө	04E9
с	0454	ӯ	04EF
ѕ	0455	ӱ	04F1
і	0456	ӳ	04F3
с	0457	#	04F7
ј	0458	ы	04F9

Set karakter Telugu

Untuk kosakata kustom Telugu, Anda dapat menggunakan karakter berikut di lapangan: `Phrase`

Karakter	Kode	Karakter	Kode
-	002D	ఴ	0C24
ఁ	0C01	఺	0C25
ం	0C02	ఽ	0C26
ః	0C03	ా	0C27

Karakter	Kode	Karakter	Kode
అ	0C05	న	0C28
ఆ	0C06	ప	0C2A
ఇ	0C07	ఫ	0C2B
ఈ	0C08	బ	0C2C
ఉ	0C09	భ	0C2D
ఊ	0C0A	మ	0C2E
ఋ	0C0B	య	0C2F
ర	0C30	ఎ	0C0E
ఱ	0C31	ఏ	0C0F
ల	0C32	ఐ	0C10
ళ	0C33	ఒ	0C12
వ	0C35	ఓ	0C13
శ	0C36	ౌ	0C14
ష	0C37	క	0C15
స	0C38	ఖ	0C16
హ	0C39	గ	0C17
ా	0C3E	ఘ	0C18
ి	0C3F	ఙ	0C19
ీ	0C40	చ	0C1A
ు	0C41	ఛ	0C1B

Karakter	Kode	Karakter	Kode
ా	0C42	ఙ	0C1C
ి	0C43	ఛ	0C1D
ీ	0C44	ఝ	0C1E
ు	0C47	ఞ	0C1F
ూ	0C48	ఠ	0C20
ృ	0C4A	డ	0C21
ౄ	0C4B	ఢ	0C22
ె	0C4C	ణ	0C23
ే	0C4D		

Set karakter Thailand

Untuk kosakata khusus Thailand, Anda dapat menggunakan karakter berikut di `Phrase` lapangan:

- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
ก	0E01	ล	0E25
ข	0E02	ฦ	0E26
ฃ	0E03	ว	0E27
ค	0E04	ศ	0E28

Karakter	Kode	Karakter	Kode
ค	0E05	ษ	0E29
ฆ	0E06	ส	0E2A
ง	0E07	ห	0E2B
จ	0E08	ฬ	0E2C
ฉ	0E09	อ	0E2D
ช	0E0A	ฮ	0E2E
ซ	0E0B	ฯ	0E2F
ฌ	0E0C	ะ	0E30
ญ	0E0D	ั	0E31
ฎ	0E0E	า	0E32
ฏ	0E0F	ิ	0E34
ฐ	0E10	ี	0E35
ฑ	0E11	ึ	0E36
ฒ	0E12	ื	0E37
ณ	0E13	ุ	0E38
ด	0E14	ู	0E39
ต	0E15	ฺ	0E3A
ถ	0E16	เ	0E40
ท	0E17	แ	0E41
ธ	0E18	โ	0E42

Karakter	Kode	Karakter	Kode
น	0E19	ใ	0E43
บ	0E1A	ไ	0E44
ป	0E1B	ๅ	0E45
ผ	0E1C	ๆ	0E46
ฝ	0E1D	็	0E47
พ	0E1E	่	0E48
ฟ	0E1F	้	0E49
ภ	0E20	๊	0E4A
ม	0E21	๋	0E4B
ย	0E22	์	0E4C
ร	0E23	๎	0E4D
ฤ	0E24		

Set karakter Turki

Untuk kosakata kustom Turki, Anda dapat menggunakan karakter berikut di `Phrase` lapangan:

- a - z
- A - Z
- '(apostrof)
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

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

Set karakter Ukraina

Untuk kosakata khusus Ukraina, Anda dapat menggunakan karakter berikut di `Phrase` lapangan:

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
а	0430	р	0440
b	0431	с	0441
в	0432	т	0442
г	0433	у	0443
д	0434	ω	0444
е	0435	х	0445
ж	0436	ц	0446
з	0437	ч	0447
и	0438	÷	0448
в	0439	щ	0449
К	043A	ь	044C
л	043B	ю	044E
m	043C	я	044F
н	043D	с	0454
о	043E	і	0456
р	043F	с	0457
ґ	0491		

Set karakter Uyghur

Untuk kosakata kustom Uyghur, Anda dapat menggunakan karakter berikut di lapangan: `Phrase`

- a - z

- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
#	0611	و	0648
#	0613	ى	0649
#	0614	1	064A
ء	0621	ً	064B
4	0622	ٌ	064C
أ	0623	ٍ	064D
ؤ	0624	َ	064E
إ	0625	ُ	064F
ئ	0626	ِ	0650
ا	0627	ّ	0651
م	0628	ْ	0652
س	0629	#	0653
ت	062A	#	0654
ث	062B	#	0657
ج	062C	ٰ	0670
ح	062D	ٹ	0679
خ	062E	ٺ	067A

Karakter	Kode	Karakter	Kode
د	062F	ٻ	067B
ذ	0630	ټ	067C
ر	0631	ٽ	067D
4	0632	پ	067E
س	0633	ٿ	067F
ش	0634	ڀ	0680
ص	0635	ځ	0681
ض	0636	ڄ	0683
ط	0637	ڃ	0684
ظ	0638	څ	0685
ع	0639	چ	0686
غ	063A	ڇ	0687
ـ	0640	ڈ	0688
ف	0641	ډ	0689
ق	0642	ڊ	068A
ك	0643	ڌ	068C
ل	0644	ڍ	068D
م	0645	ڎ	068F
م	0646	ڑ	0691
ه	0647	ړ	0693

Karakter	Kode	Karakter	Kode
ڕ	0695		

Set karakter Uzbek

Untuk kosakata khusus Uzbek, Anda dapat menggunakan karakter berikut di lapangan: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
t	0442	я	044F
b	0431	c	0441
o	043E	ҳ	04B3
p	043F	д	0434
÷	0448	p	0440
и	0438	ў	045E
ч	0447	г	0433
н	043D	g	0451
қ	049B	в	0439
e	0435	в	0432
ю	044E	э	044D
з	0437	л	043B

Karakter	Kode	Karakter	Kode
х	0445	÷	0444
ц	0446	К	043A
m	043C	у	0443
ь	044C	ж	0436
ъ	044A	ғ	0493
а	0430		

Set karakter Vietnam

Amazon Transcribe mewakili enam nada dalam bahasa Vietnam menggunakan angka. Tabel berikut menunjukkan bagaimana tanda nada dipetakan untuk kata 'ma'.

Nama nada	Tanda nada	Nomor nada
ngang	ma	ma1
sắc	má	ma2
huyn	mà	ma3
hi	m	ma4
ngã	mã	ma5
nng	mạ	ma6

Untuk kosakata kustom Vietnam, Anda dapat menggunakan karakter berikut di bidang: `Phrase`

- a - z
- A - Z
- '(apostrof)

- - (tanda hubung)
- . (periode)
- & (Ampersand)
- ; (titik koma)
- _ (garis rendah)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
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

Karakter	Kode	Karakter	Kode
ý	00FD	Ý	00DD
ă	0103	Ă	0102
đ	0111	Đ	0110
ĩ	0129	Ĩ	0128
ũ	0169	Ũ	0168
ơ	01A1	Ơ	01A0
ư	01B0	Ư	01AF
ạ	1 EA1	Ạ	1EA0
ả	1 EA3	Ả	1 EA2
ấ	1 EA5	Ấ	1 EA4
ầ	1 EA7	Ầ	1 EA6
ẩ	1 EA9	Ẩ	1 EA8
ẫ	1EAB	Ẫ	1EAA
ậ	1EAD	Ậ	1EAC
ắ	1EAF	Ắ	1EAE
ằ	1 EB1	Ằ	1EB0
ẳ	1 EB3	Ẳ	1 EB2
ẵ	1 EB5	Ẵ	1 EB4
ặ	1 EB7	Ặ	1 EB6
ẹ	1 EB9	Ẹ	1 EB8

Karakter	Kode	Karakter	Kode
ẻ	1PASANG SURUT	Ẻ	1EBA
ẽ	1EBD	Ẽ	1EBC
ế	1EBF	Ế	1EBE
11	1 EC1	11	1EC0
ể	1 EC3	Ể	1 EC2
ễ	1 EC5	Ễ	1 EC4
ệ	1 EC7	Ệ	1 EC6
ỉ	1 EC9	Ỉ	1 EC8
ị	1ECB	Ị	1ECA
ọ	1ECD	Ọ	1ECC
11	1ECF	11	1ECE
ố	1 ED1	Ố	1ED0
ồ	1 ED3	Ồ	1 ED2
ổ	1 ED5	Ổ	1 ED4
ỗ	1 ED7	Ỗ	1 ED6
ộ	1 ED9	Ộ	1 ED8
ớ	1EDB	Ớ	1EDA
ờ	1EDD	Ờ	1EDC
11	1EDF	11	1EDE
ỡ	1 EE1	Ỡ	1EE0

Karakter	Kode	Karakter	Kode
ợ	1 EE3	Ợ	1 EE2
ụ	1 EE5	Ụ	1 EE4
ủ	1 EE7	Ủ	1 EE6
ứ	1 EE9	Ứ	1 EE8
ừ	1EEB	Ừ	1EEA
ử	1EED	Ử	1EEC
ữ	1EEF	Ữ	1EEE
ự	1 EF1	Ự	1EF0
ỳ	1 EF3	Ỳ	1 EF2
ỵ	1 EF5	Ỵ	1 EF4
ỷ	1 EF7	Ỷ	1 EF6
ỹ	1 EF9	Ỹ	1 EF8

Set karakter Welsh

Untuk kosakata kustom Welsh, Anda dapat menggunakan karakter berikut di bidang: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
à	00E0	ò	00F2

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

Set karakter Wolof

Untuk kosakata khusus Wolof, Anda dapat menggunakan karakter berikut di bidang: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
à	00E0	ê	00EA

Karakter	Kode	Karakter	Kode
ã	00E3	ë	00EB
ç	00E7	ñ	00F1
è	00E8	ó	00F3
é	00E9	ô	00F4
ŋ	014B		

Set karakter Zulu

Untuk kosakata khusus Zulu, Anda dapat menggunakan karakter berikut di lapangan: `Phrase`

- a - z
- - (tanda hubung)
- . (periode)

Anda juga dapat menggunakan karakter Unicode berikut di `Phrase` bidang:

Karakter	Kode	Karakter	Kode
detik	0073	d	0064
t	0074	a	0061
a	0061	r	0072
n	006E	d	0064

Bagaimana cara Amazon Transcribe kerja

Amazon Transcribe menggunakan model pembelajaran mesin untuk mengubah ucapan menjadi teks.

Selain teks yang ditranskripsi, transkrip berisi data tentang konten yang ditranskripsi, termasuk skor kepercayaan dan stempel waktu untuk setiap kata atau tanda baca. Untuk melihat contoh keluaran, lihat bagian Input dan output data. Untuk daftar lengkap fitur yang dapat Anda terapkan ke transkripsi Anda, lihat ringkasan fitur.

Metode transkripsi dapat dipisahkan menjadi dua kategori utama:

- Transkripsi Batch: Transcribe file media yang telah diunggah ke dalam ember. Amazon S3 Anda dapat menggunakan AWS CLI, AWS Management Console, dan berbagai AWS SDKsuntuk transkripsi batch.
- Transkripsi streaming: Transkripsikan aliran media secara real time. Anda dapat menggunakan AWS Management Console, HTTP/2, WebSockets, dan berbagai AWS SDKsuntuk streaming transkripsi.

Perhatikan bahwa dukungan fitur dan bahasa berbeda untuk transkripsi batch dan streaming. Untuk informasi selengkapnya, lihat Amazon Transcribe fitur dan Bahasa yang didukung.

Topik

- Input dan output data
- Mentranskripsikan angka dan tanda baca

> Operasi API untuk membantu Anda memulai
>
> Batch: `StartTranscriptionJob`
> Streaming: `StartStreamTranscription`, StartStreamTranscriptionWebSocket

Input dan output data

Amazon Transcribe mengambil data audio, sebagai file media dalam Amazon S3 ember atau aliran media, dan mengubahnya menjadi data teks.

Jika Anda mentranskripsikan file media yang disimpan dalam Amazon S3 ember, Anda melakukan transkripsi batch. Jika Anda mentranskripsikan aliran media, Anda melakukan transkripsi streaming. Kedua proses ini memiliki aturan dan persyaratan yang berbeda.

Dengan transkripsi batch, Anda dapat menggunakan Antrian pekerjaan jika Anda tidak perlu memproses semua pekerjaan transkripsi Anda secara bersamaan. Ini memungkinkan Amazon Transcribe untuk melacak pekerjaan transkripsi Anda dan memprosesnya saat slot tersedia.

> **Note**
>
> Amazon Transcribe dapat menyimpan konten Anda untuk sementara waktu untuk terus meningkatkan kualitas model analisisnya. Lihat Amazon Transcribe FAQ untuk mempelajari lebih lanjut. Untuk meminta penghapusan konten yang mungkin telah disimpan oleh Amazon Transcribe, buka kasing dengan. Dukungan

Topik

- Format media
- Saluran audio
- Tarif sampel
- Output

Format media

Jenis media yang didukung berbeda antara transkripsi batch dan transkripsi streaming, meskipun format lossless direkomendasikan untuk keduanya. Lihat tabel berikut untuk detailnya:

	Batch	Streaming
Format yang didukung	• AMR • FLAC • M4A • MP3 • MP4 • Ogg	• FLAC • Ogg Opus • Pengkodean PCM

	Batch	Streaming
	• WebM • WAV	
Format yang direkomendasikan	• FLAC • WAV dengan pengkodean PCM 16-bit	• FLAC • PCM menandatangani audio endian kecil 16-bit (perhatikan bahwa ini tidak termasuk WAV)

Untuk hasil terbaik, gunakan format lossless, seperti FLAC atau WAV dengan pengkodean PCM 16-bit.

> **Note**
>
> Transkripsi streaming tidak didukung dengan semua bahasa. Lihat kolom 'Input data' dalam tabel bahasa yang didukung untuk detailnya.

Saluran audio

Amazon Transcribe mendukung media saluran tunggal dan saluran ganda. Media dengan lebih dari dua saluran saat ini tidak didukung.

Jika audio Anda berisi beberapa speaker pada satu saluran dan Anda ingin mempartisi dan memberi label pada setiap speaker dalam output transkripsi Anda, Anda dapat menggunakan partisi Speaker (diarisasi).

Jika audio Anda berisi ucapan di dua saluran terpisah, Anda dapat menggunakan identifikasi Saluran untuk mentranskripsikan setiap saluran secara terpisah dalam transkrip Anda.

Kedua opsi ini menghasilkan satu file transkrip.

> **Note**
>
> Jika Anda tidak mengaktifkan partisi Speaker atau identifikasi Saluran, teks transkrip Anda disediakan sebagai satu bagian berkelanjutan.

Tarif sampel

Dengan pekerjaan transkripsi batch, Anda dapat memilih untuk memberikan laju sampel, meskipun parameter ini opsional. Jika Anda memasukkannya dalam permintaan Anda, pastikan nilai yang Anda berikan cocok dengan laju sampel aktual dalam audio Anda. Jika Anda memberikan sample rate yang tidak sesuai dengan audio Anda, pekerjaan Anda mungkin gagal.

Dengan transkripsi streaming, Anda harus menyertakan laju sampel dalam permintaan Anda. Seperti halnya pekerjaan transkripsi batch, pastikan nilai yang Anda berikan cocok dengan laju sampel aktual dalam audio Anda.

Kecepatan sampel untuk audio dengan kesetiaan rendah, seperti rekaman telepon, biasanya menggunakan 8.000 Hz. Untuk audio dengan kesetiaan tinggi, Amazon Transcribe mendukung nilai antara 16.000 Hz dan 48.000 Hz.

Output

Output transkripsi dalam format JSON. Bagian pertama transkrip Anda berisi transkrip itu sendiri dalam bentuk paragraf, diikuti dengan data tambahan untuk setiap kata dan tanda baca. Data yang diberikan tergantung pada fitur yang Anda sertakan dalam permintaan Anda. Minimal, transkrip Anda berisi waktu mulai, waktu akhir, dan skor kepercayaan untuk setiap kata. Bagian berikut menunjukkan contoh keluaran dari permintaan transkripsi dasar yang tidak menyertakan opsi atau fitur tambahan apa pun.

Semua transkrip batch disimpan dalam Amazon S3 ember. Anda dapat memilih untuk menyimpan transkrip di Amazon S3 bucket Anda sendiri, atau Amazon Transcribe menggunakan bucket default yang aman. Untuk mempelajari lebih lanjut tentang membuat dan menggunakan Amazon S3 bucket, lihat Bekerja dengan bucket.

Jika Anda ingin transkrip Anda disimpan dalam Amazon S3 ember yang Anda miliki, tentukan URI bucket dalam permintaan transkripsi Anda. Pastikan Anda memberikan izin Amazon Transcribe menulis untuk bucket ini sebelum memulai pekerjaan transkripsi batch Anda. Jika Anda menentukan bucket Anda sendiri, transkrip Anda tetap berada di ember itu sampai Anda menghapusnya.

Jika Anda tidak menentukan Amazon S3 bucket, Amazon Transcribe gunakan bucket yang dikelola layanan aman dan memberi Anda URI sementara yang dapat Anda gunakan untuk mengunduh transkrip Anda. Perhatikan bahwa sementara URIs berlaku selama 15 menit. Jika Anda mendapatkan `AccessDenied` kesalahan saat menggunakan URI yang disediakan, buat `GetTranscriptionJob` permintaan untuk mendapatkan URI sementara baru untuk transkrip Anda.

Jika Anda memilih bucket default, transkrip Anda akan dihapus ketika pekerjaan Anda kedaluwarsa (90 hari). Jika Anda ingin menyimpan transkrip Anda melewati tanggal kedaluwarsa ini, Anda harus mengunduhnya.

Transkrip streaming dikembalikan melalui metode yang sama yang Anda gunakan untuk streaming Anda.

> **Tip**
>
> Jika Anda ingin mengonversi output JSON Anda menjadi turn-by-turn transkrip dalam format Word, lihat GitHub contoh ini (untuk Python3). Skrip ini berfungsi dengan transkrip analitik pasca-panggilan dan transkrip batch standar dengan diarisasi diaktifkan.

Contoh Output

Transkrip memberikan transkripsi lengkap dalam bentuk paragraf, diikuti dengan word-for-word perincian, yang menyediakan data untuk setiap kata dan tanda baca. Ini termasuk waktu mulai, waktu akhir, skor kepercayaan diri, dan tipe (`pronunciation`atau`punctuation`).

Contoh berikut adalah dari pekerjaan transkripsi batch sederhana yang tidak menyertakan fitur tambahan apa pun. Dengan setiap fitur tambahan yang Anda terapkan pada permintaan transkripsi Anda, Anda mendapatkan data tambahan dalam file keluaran transkrip Anda.

Transkrip batch dasar berisi dua bagian utama:

1. `transcripts`: Berisi seluruh transkrip dalam satu blok teks.
2. `items`: Berisi informasi tentang setiap kata dan tanda baca dari bagian. `transcripts`
3. `audio_segments`Segmen audio adalah bagian tertentu dari rekaman audio yang berisi bahasa lisan tanpa gangguan, dengan jeda atau jeda minimal. Segmen ini menangkap aliran bicara alami dan ditangkap `audio_segments` dengan waktu mulai dan waktu akhir. `items`Elemen dalam segmen audio adalah urutan pengidentifikasi yang sesuai dengan setiap item dalam segmen.

Setiap fitur tambahan yang Anda sertakan dalam permintaan transkripsi menghasilkan informasi tambahan dalam transkrip Anda.

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

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

```
                "id": 3,
                "start_time": "1.74",
                "end_time": "2.56",
                "alternatives": [
                    {
                        "confidence": "1.0",
                        "content": "Transcribe"
                    }
                ],
                "type": "pronunciation"
            },
            {
                "id": 4,
                "alternatives": [
                    {
                        "confidence": "0.0",
                        "content": "."
                    }
                ],
                "type": "punctuation"
            }
        ],
        "audio_segments": [
            {
                "id": 0,
                "transcript": "Welcome to Amazon Transcribe.",
                "start_time": "0.64",
                "end_time": "2.56",
                "items": [
                    0,
                    1,
                    2,
                    3,
                    4
                ]
            }
        ]
    },
    "status": "COMPLETED"
}
```

Mentranskripsikan angka dan tanda baca

Amazon Transcribe secara otomatis menambahkan tanda baca ke semua bahasa yang didukung, dan menggunakan huruf kapital dengan tepat untuk bahasa yang menggunakan perbedaan kasus dalam sistem penulisan mereka.

Untuk sebagian besar bahasa, angka ditranskripsikan ke dalam bentuk kata mereka. Namun, untuk bahasa dengan dukungan untuk menyalin angka, Amazon Transcribe memperlakukan angka secara berbeda tergantung pada konteks di mana mereka digunakan.

Misalnya, jika seorang pembicara mengatakan”Temui saya pukul delapan tiga puluh pagi pada bulan Juni pertama di seratus Main Street dengan three-dollars-and-fifty -sen dan cokelat batangan one-point-five,” ini ditranskripsikan sebagai:

- Bahasa dengan dukungan transkripsi angka: Meet me at 8:30 a.m. on June 1st at 100 Main Street with $3.50 and 1.5 chocolate bars
- Semua bahasa lainnya: 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

Untuk melihat bahasa dengan dukungan untuk menyalin angka, lihat. Bahasa yang didukung dan fitur khusus bahasa

Memulai dengan Amazon Transcribe

Sebelum Anda dapat membuat transkripsi, Anda memiliki beberapa prasyarat:

- Mendaftar untuk Akun AWS
- Instal AWS CLI dan SDKs (jika Anda menggunakan AWS Management Console untuk transkripsi Anda, Anda dapat melewati langkah ini)
- Konfigurasikan IAM kredensil
- Siapkan Amazon S3 ember
- Buat IAM kebijakan

Setelah Anda menyelesaikan prasyarat ini, Anda siap untuk menyalin. Pilih metode transkripsi pilihan Anda dari daftar berikut untuk memulai.

- AWS CLI
- AWS Management Console
- AWS SDK
- HTTP
- WebSockets

> **Tip**
>
> Jika Anda baru Amazon Transcribe atau ingin menjelajahi fitur kami, kami sarankan untuk menggunakan AWS Management Console. Ini juga merupakan opsi termudah jika Anda ingin memulai streaming menggunakan mikrofon komputer Anda.

Karena streaming menggunakan HTTP/2 dan WebSockets lebih rumit daripada metode transkripsi lainnya, kami sarankan untuk meninjau Menyiapkan transkripsi streaming bagian sebelum memulai dengan metode ini. Perhatikan bahwa kami sangat menyarankan menggunakan SDK untuk streaming transkripsi.

Mendaftar untuk Akun AWS

Anda dapat mendaftar untuk akun tingkat gratis atau akun berbayar. Kedua opsi memberi Anda akses ke semua Layanan AWS. Tingkat gratis memiliki masa uji coba di mana Anda dapat menjelajahi Layanan AWS dan memperkirakan penggunaan Anda. Setelah masa uji coba berakhir, Anda dapat bermigrasi ke akun berbayar. Biaya dikenakan atas pay-as-you-use dasar; lihat Amazon Transcribe Harga untuk detailnya.

> **Tip**
>
> Saat menyiapkan akun Anda, catat Akun AWS ID Anda karena Anda memerlukannya untuk membuat IAM entitas.

Instalasi AWS CLI dan SDKs

Untuk menggunakan Amazon Transcribe API, Anda harus menginstal file terlebih dahulu AWS CLI. Saat ini AWS CLI adalah versi 2. Anda dapat menemukan petunjuk penginstalan untuk Linux, Mac, Windows, dan Docker di Panduan AWS Command Line Interface Pengguna.

Setelah Anda AWS CLI menginstal, Anda harus mengkonfigurasinya untuk kredensi keamanan Anda dan. Wilayah AWS

Jika Anda ingin menggunakan Amazon Transcribe SDK, pilih bahasa pilihan Anda untuk petunjuk penginstalan:

- .NET
- C++
- Go
- Jawa V2
- JavaScript
- PHP V3
- AWS SDK untuk Python (Boto3)(transkripsi batch)
- Python (transkripsi streaming)
- Ruby V3
- Rust (transkripsi batch)

- Rust (transkripsi streaming)

Konfigurasikan IAM kredensil

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang memiliki akses lengkap ke semua AWS layanan dan sumber daya di akun Anda. Identitas ini disebut pengguna Akun AWS root dan diakses dengan masuk dengan alamat email dan kata sandi yang Anda gunakan untuk membuat akun.

Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari Anda. Lindungi kredensial pengguna root Anda dan gunakan kredensial tersebut untuk melakukan tugas yang hanya dapat dilakukan pengguna root.

Sebagai praktik terbaik, mewajibkan pengguna—termasuk yang memerlukan akses administrator—untuk menggunakan federasi dengan penyedia identitas untuk mengakses AWS layanan dengan menggunakan kredensi sementara.

Identitas federasi adalah setiap pengguna yang mengakses AWS layanan dengan menggunakan kredensi yang disediakan melalui sumber identitas. Ketika identitas terfederasi mengakses Akun AWS, identitas tersebut mengambil peran, dan peran ini memberikan kredensial sementara.

Untuk manajemen akses terpusat, kami sarankan Anda menggunakan AWS IAM Identity Center. Anda dapat membuat pengguna dan grup di IAM Identity Center. Atau Anda dapat menghubungkan dan menyinkronkan ke sekumpulan pengguna dan grup di sumber identitas Anda sendiri untuk digunakan di semua aplikasi Akun AWS dan aplikasi Anda. Untuk informasi selengkapnya, lihat Identity and Access Management untuk Amazon Transcribe.

Untuk mempelajari lebih lanjut tentang praktik IAM terbaik, lihat Praktik terbaik keamanan di IAM.

Membuat Amazon S3 ember

Amazon S3 adalah layanan penyimpanan objek yang aman. Amazon S3 menyimpan file Anda (disebut objek) dalam wadah (disebut ember).

Untuk menjalankan transkripsi batch, Anda harus terlebih dahulu mengunggah file media Anda ke dalam Amazon S3 ember. Jika Anda tidak menentukan Amazon S3 bucket untuk keluaran transkripsi Anda, Amazon Transcribe masukkan transkrip Anda ke dalam bucket yang AWS dikelola Amazon S3 sementara. Output transkripsi dalam bucket AWS yang dikelola secara otomatis dihapus setelah 90 hari.

Pelajari cara Membuat bucket S3 pertama Anda dan Unggah objek ke bucket Anda.

Membuat IAM kebijakan

Untuk mengelola akses AWS, Anda harus membuat kebijakan dan melampirkannya ke IAM identitas (pengguna, grup, atau peran) atau AWS sumber daya. Kebijakan mendefinisikan izin entitas yang dilampirkan padanya. Misalnya, peran hanya dapat mengakses file media yang terletak di Amazon S3 bucket jika Anda telah melampirkan kebijakan ke peran yang memberinya akses. Jika Anda ingin membatasi peran itu lebih lanjut, Anda dapat membatasi aksesnya ke file tertentu dalam Amazon S3 bucket.

Untuk mempelajari lebih lanjut tentang menggunakan AWS kebijakan, lihat:

- Kebijakan dan izin di IAM
- Membuat IAM kebijakan
- Bagaimana Amazon Transcribe bekerja dengan IAM

Misalnya kebijakan yang dapat Anda gunakan Amazon Transcribe, lihatAmazon Transcribe contoh kebijakan berbasis identitas. Jika Anda ingin membuat kebijakan khusus, pertimbangkan untuk menggunakan AWS Policy Generator.

Anda dapat menambahkan kebijakan menggunakan AWS Management Console, AWS CLI, atau AWS SDK. Untuk petunjuk, lihat Menambahkan dan menghapus izin IAM identitas.

Kebijakan memiliki format:

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

```
    ]
}
```

Amazon Resource Names (ARNs) secara unik mengidentifikasi semua AWS sumber daya, seperti bucket. Amazon S3 Anda dapat menggunakan ARNs kebijakan Anda untuk memberikan izin untuk tindakan tertentu guna menggunakan sumber daya tertentu. Misalnya, jika Anda ingin memberikan akses baca ke Amazon S3 bucket dan sub-foldernya, Anda dapat menambahkan kode berikut ke `Statement` bagian kebijakan kepercayaan Anda:

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

Berikut adalah contoh kebijakan yang memberikan izin Amazon Transcribe read (`GetObject`,`ListBucket`) dan write (`PutObject`) ke Amazon S3 bucket`amzn-s3-demo-bucket`, dan sub-foldernya:

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

```
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        }
    ]
}
```

Mentranskripsikan dengan AWS Management Console

Anda dapat menggunakan AWS konsol untuk transkripsi batch dan streaming. Jika Anda mentranskripsikan file media yang terletak di Amazon S3 ember, Anda sedang melakukan transkripsi batch. Jika Anda menyalin aliran data audio waktu nyata, Anda melakukan transkripsi streaming.

Sebelum memulai transkripsi batch, Anda harus terlebih dahulu mengunggah file media Anda ke Amazon S3 bucket. Untuk streaming transkripsi menggunakan AWS Management Console, Anda harus menggunakan mikrofon komputer Anda.

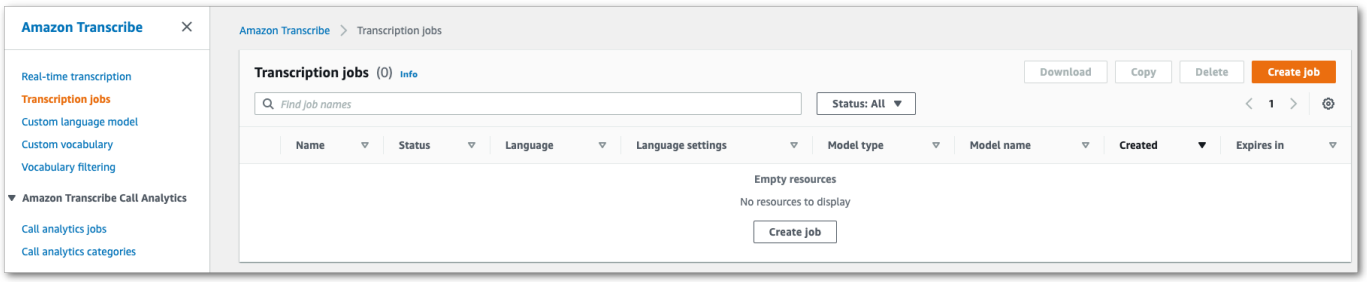
Untuk melihat format media yang didukung serta persyaratan dan kendala media lainnya, lihat. Input dan output data

Perluas bagian berikut untuk penelusuran singkat dari setiap metode transkripsi.

Transkripsi Batch

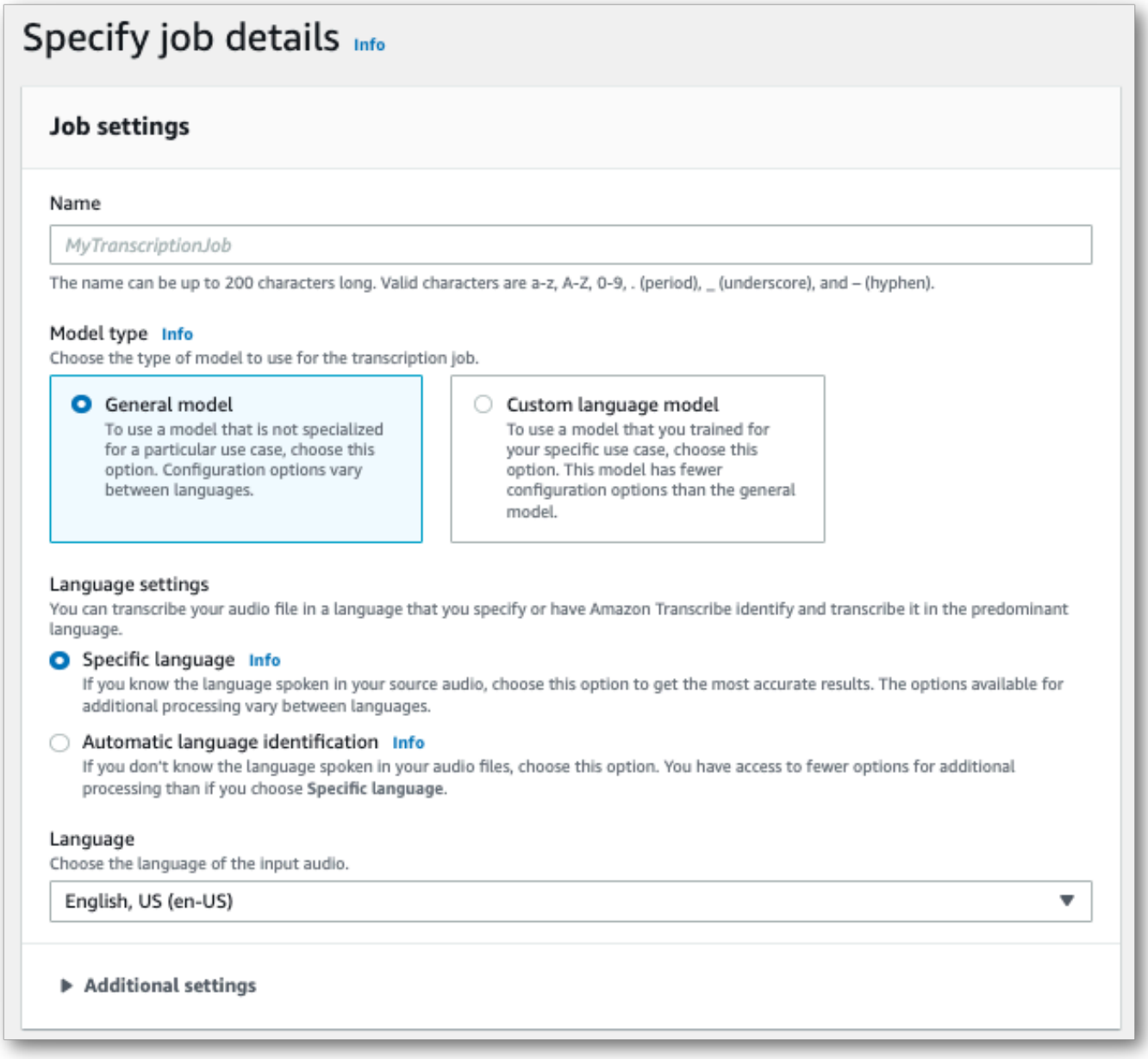
Pertama, pastikan Anda telah mengunggah file media yang ingin Anda transkripsikan ke dalam ember. Amazon S3 Jika Anda tidak yakin bagaimana melakukannya, lihat Panduan Amazon S3 Pengguna: Unggah objek ke bucket Anda.

1. Dari AWS Management Console, pilih Pekerjaan transkripsi di panel navigasi kiri. Ini membawa Anda ke daftar pekerjaan transkripsi Anda.



Pilih Buat pekerjaan.

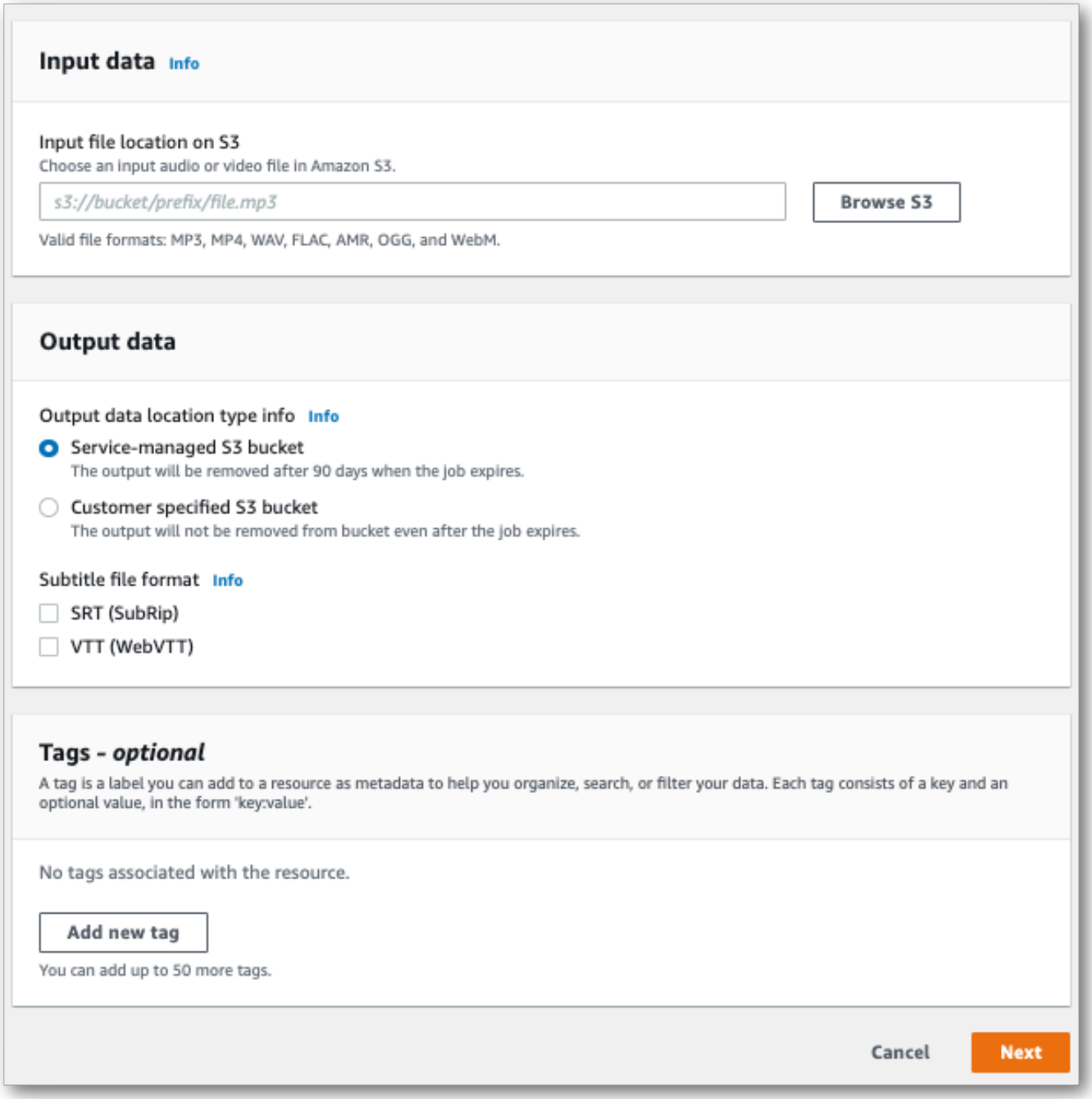
2. Lengkapi bidang pada halaman Tentukan detail pekerjaan.



Lokasi input harus berupa objek di dalam Amazon S3 ember. Untuk lokasi keluaran, Anda dapat memilih bucket yang Amazon S3 dikelola layanan yang aman atau Anda dapat menentukan bucket Anda sendiri Amazon S3 .

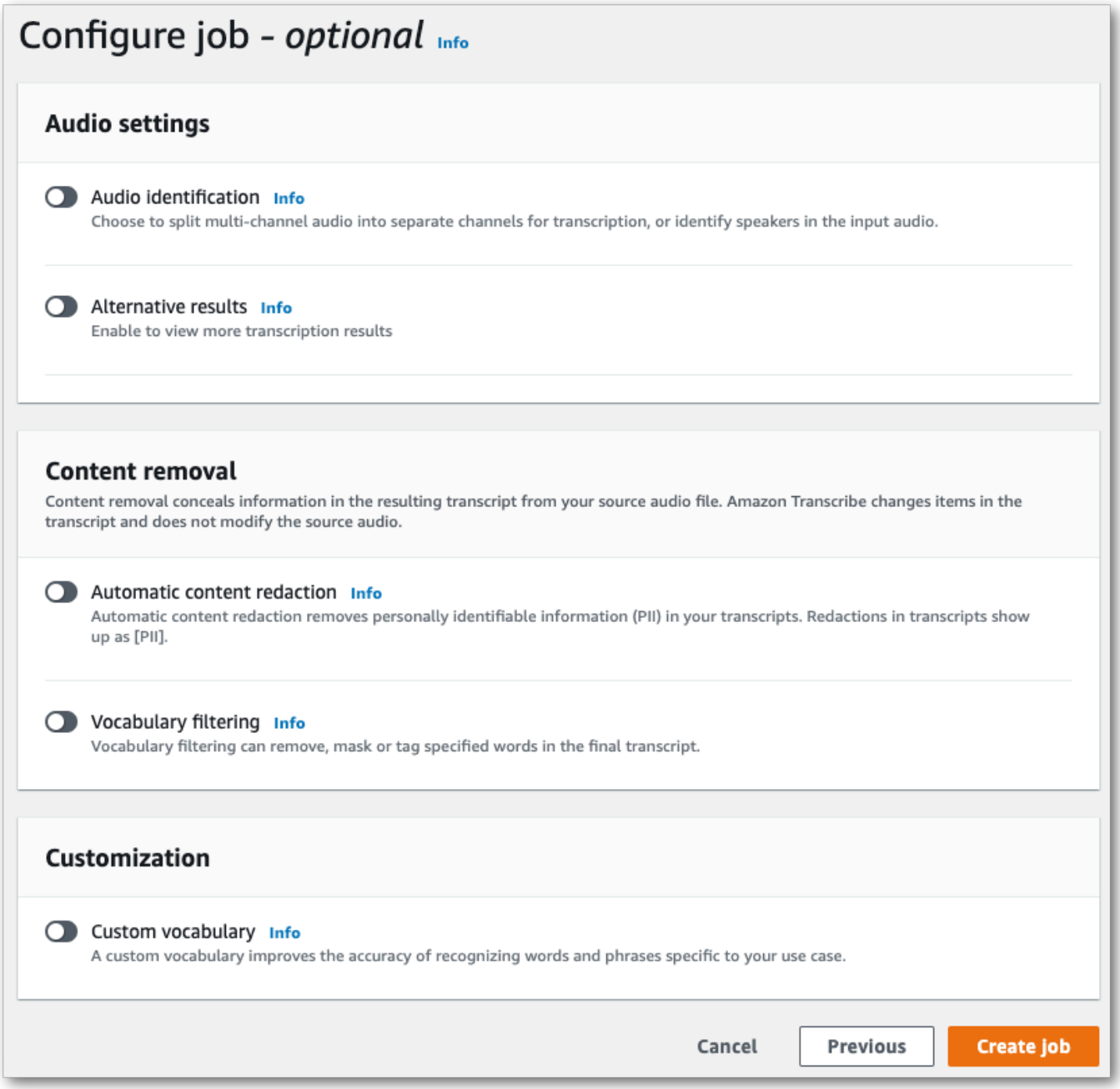
Jika Anda memilih bucket yang dikelola layanan, Anda dapat melihat pratinjau transkrip di AWS Management Console dan Anda dapat mengunduh transkrip Anda dari halaman detail pekerjaan (lihat di bawah).

Jika Anda memilih Amazon S3 bucket Anda sendiri, Anda tidak dapat melihat pratinjau di AWS Management Console dan harus pergi ke Amazon S3 ember untuk mengunduh transkrip Anda.



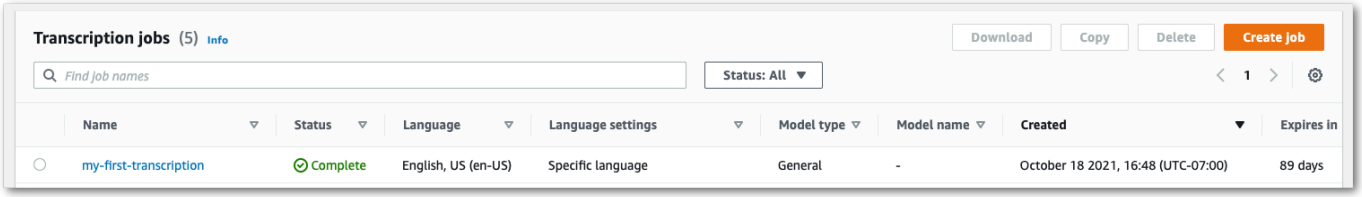
Pilih Selanjutnya.

3. Pilih opsi yang diinginkan di halaman Konfigurasi pekerjaan. Jika Anda ingin menggunakan Kosakata khusus atau Model bahasa khusus dengan transkripsi Anda, Anda harus membuatnya sebelum memulai pekerjaan transkripsi Anda.



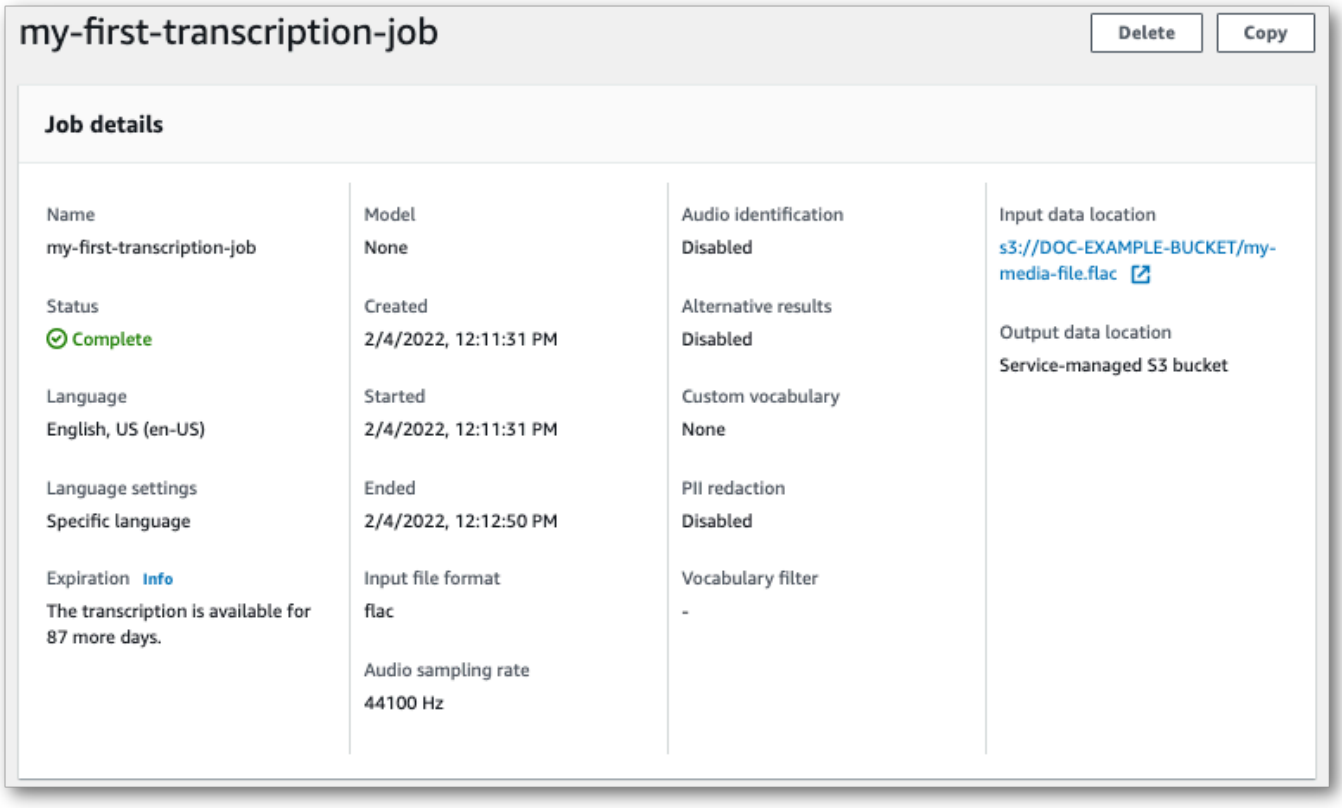
Pilih Buat pekerjaan.

4. Anda sekarang berada di halaman pekerjaan Transkripsi. Di sini Anda dapat melihat status pekerjaan transkripsi. Setelah selesai, pilih transkripsi Anda.



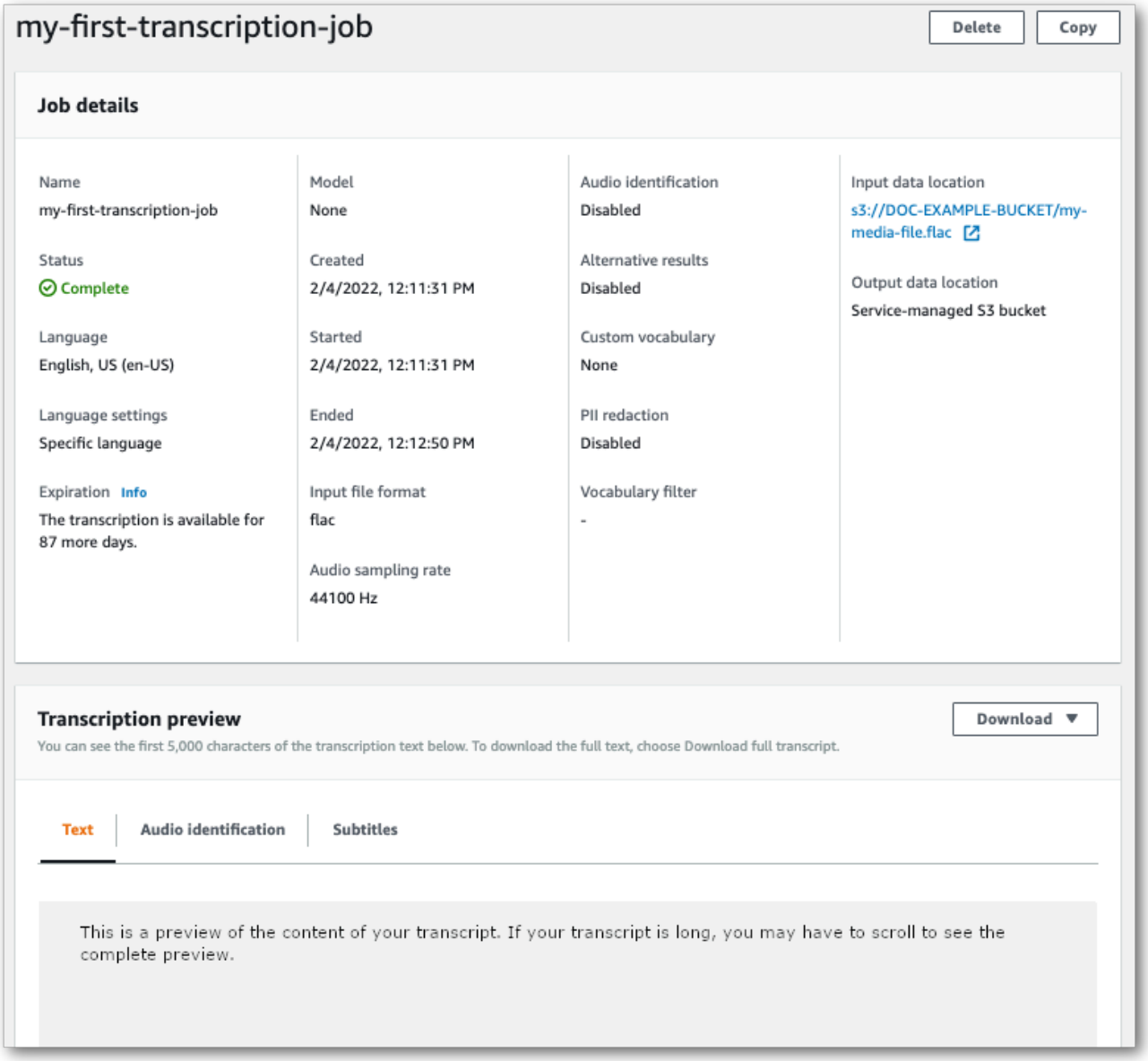
5. Anda sekarang melihat halaman Detail Pekerjaan untuk transkripsi Anda. Di sini Anda dapat melihat semua opsi yang Anda tentukan saat menyiapkan pekerjaan transkripsi Anda.

   Untuk melihat transkrip Anda, pilih jalur file yang ditautkan di kolom kanan di bawah Lokasi data keluaran. Ini membawa Anda ke folder Amazon S3 output yang Anda tentukan. Pilih file output Anda, yang sekarang memiliki ekstensi.json.



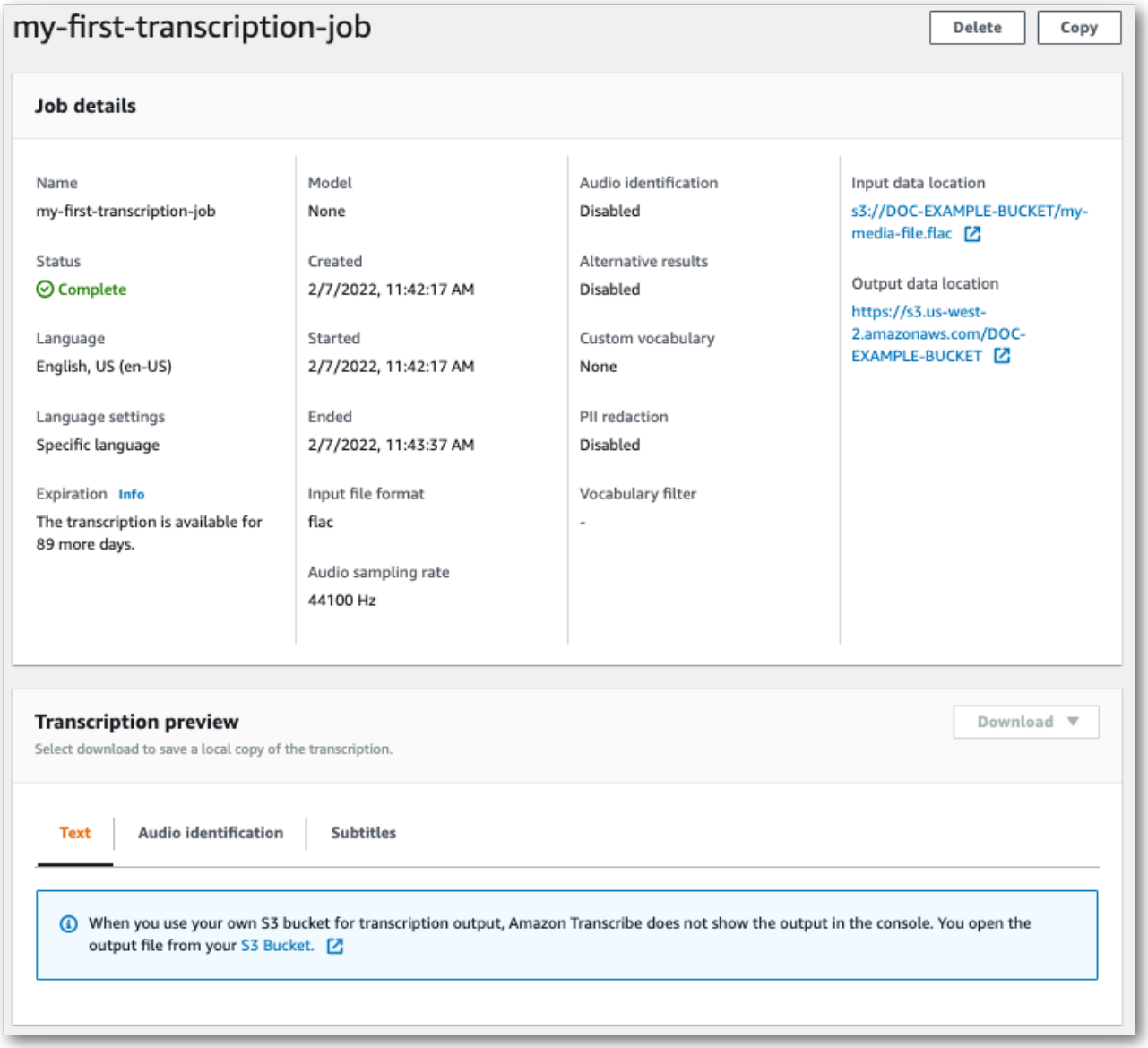
6. Cara Anda mengunduh transkrip tergantung pada apakah Anda memilih bucket yang dikelola layanan atau Amazon S3 bucket Anda sendiri. Amazon S3

   a. Jika memilih bucket yang dikelola layanan, Anda dapat melihat panel pratinjau Transkripsi di halaman informasi pekerjaan transkripsi, bersama dengan tombol Unduh.



Pilih Unduh dan pilih Unduh transkrip.

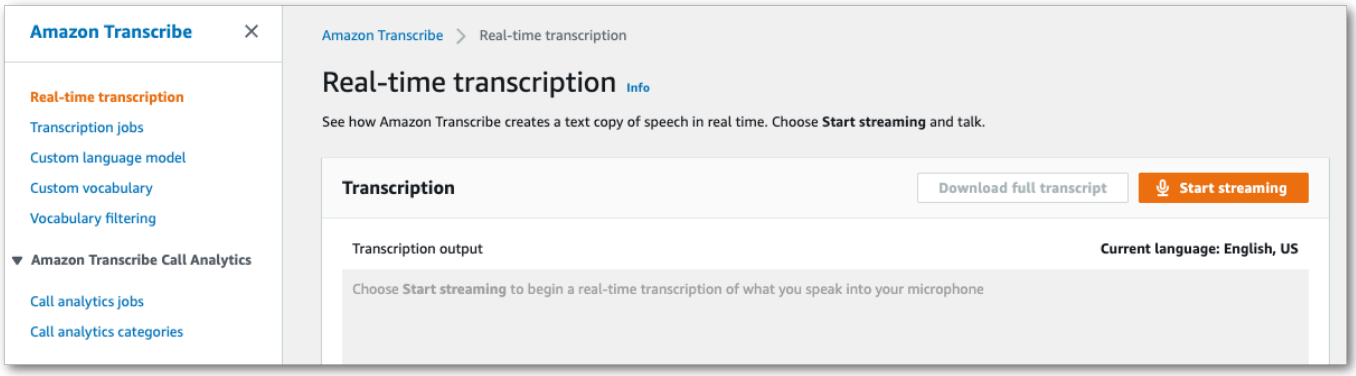
b. Jika Anda memilih Amazon S3 bucket sendiri, Anda tidak akan melihat teks apa pun di panel pratinjau Transkripsi di halaman informasi pekerjaan transkripsi Anda. Sebagai gantinya, Anda melihat kotak informasi biru dengan tautan ke Amazon S3 ember yang Anda pilih.



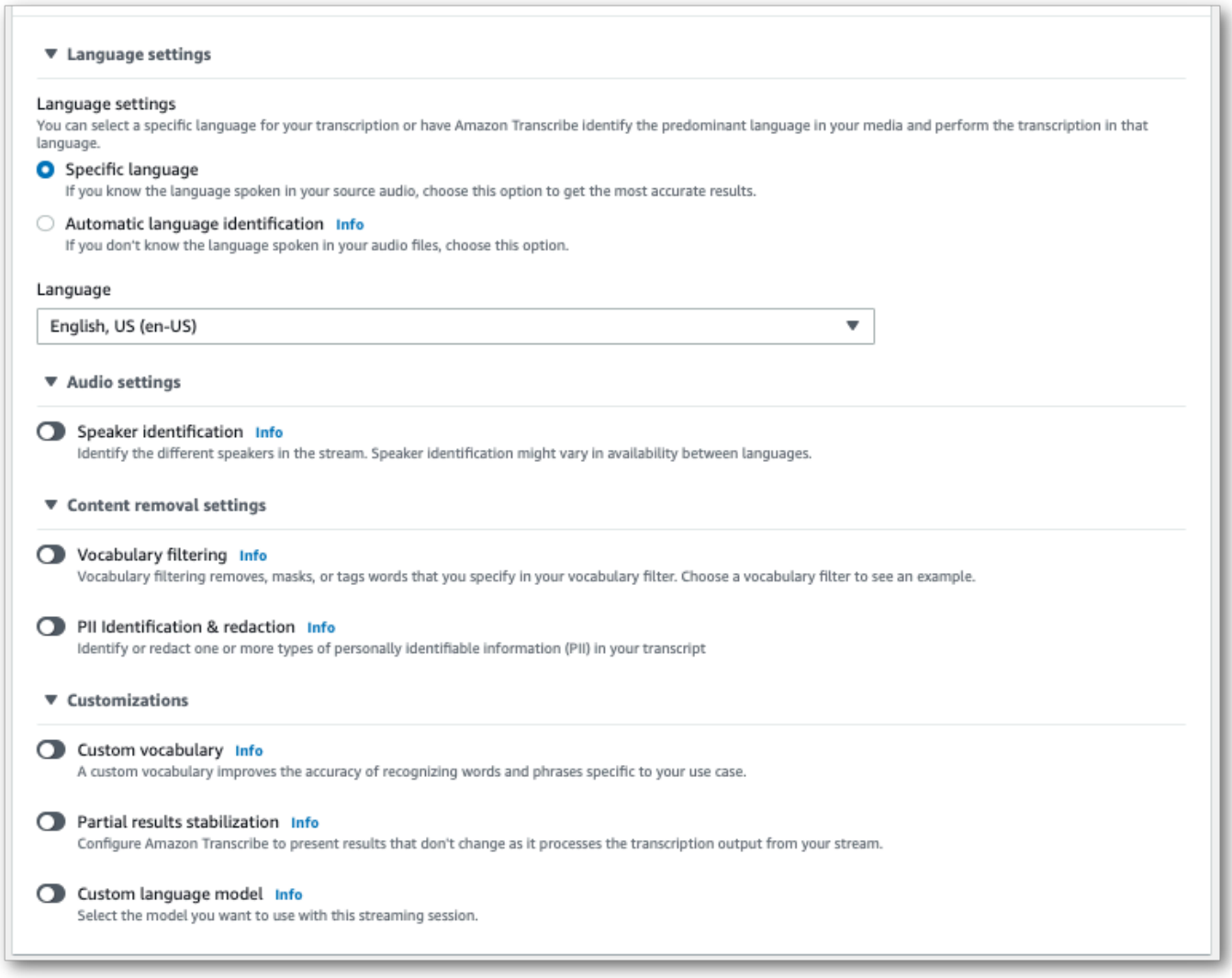
Untuk mengakses transkrip Anda, buka Amazon S3 bucket yang ditentukan menggunakan tautan di bawah Lokasi data keluaran di panel Detail pekerjaan atau tautan Bucket S3 di dalam kotak informasi biru di panel pratinjau Transkripsi.

Transkripsi streaming

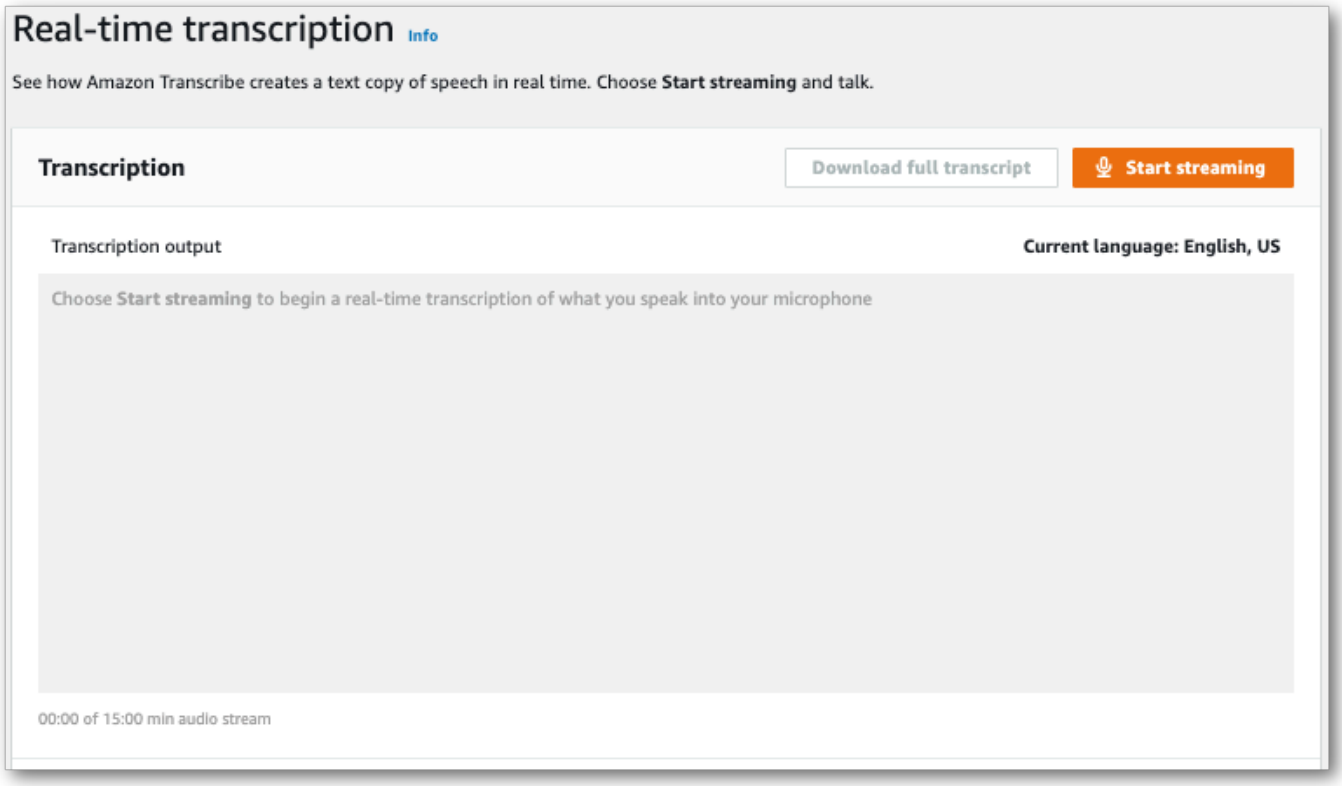
1. Dari AWS Management Console, pilih Transkripsi waktu nyata di panel navigasi kiri. Ini membawa Anda ke halaman streaming utama di mana Anda dapat memilih opsi sebelum memulai streaming Anda.



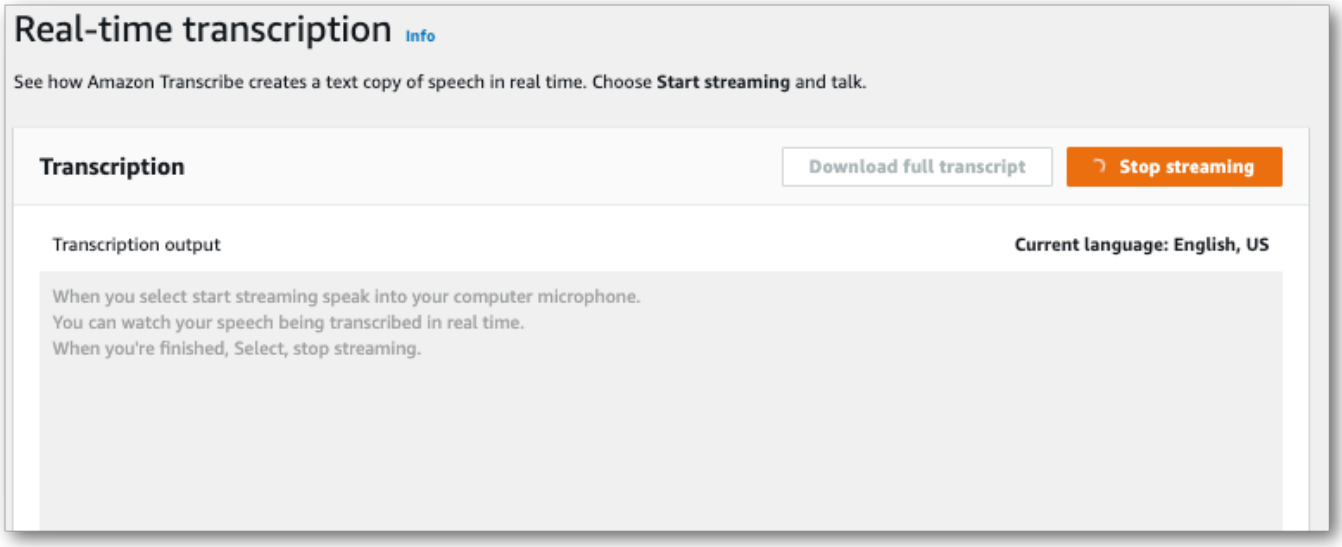
2. Di bawah kotak output Transkripsi, Anda memiliki opsi untuk memilih berbagai pengaturan bahasa dan audio.



3. Setelah Anda memilih pengaturan yang sesuai, gulir ke bagian atas halaman dan pilih Mulai streaming, lalu mulailah berbicara ke mikrofon komputer Anda. Anda dapat melihat pidato Anda ditranskripsikan secara real time.



4. Setelah selesai, pilih Hentikan streaming.



Anda sekarang dapat mengunduh transkrip Anda dengan memilih Unduh transkrip lengkap.

Mentranskripsikan dengan AWS CLI

Saat menggunakan AWS CLI untuk memulai transkripsi, Anda dapat menjalankan semua perintah di tingkat CLI. Atau Anda dapat menjalankan perintah yang ingin Anda gunakan, diikuti oleh Wilayah AWS dan lokasi file JSON yang berisi badan permintaan. Contoh di seluruh panduan ini menunjukkan kedua metode; Namun, bagian ini berfokus pada metode sebelumnya.

AWS CLI Itu tidak mendukung transkripsi streaming.

Sebelum Anda melanjutkan, pastikan Anda telah:

- Mengunggah file media Anda ke dalam Amazon S3 ember. Jika Anda tidak yakin cara membuat Amazon S3 bucket atau mengunggah file, lihat Buat Amazon S3 bucket pertama Anda dan Unggah objek ke bucket Anda.
- Menginstal AWS CLI.

Anda dapat menemukan semua AWS CLI perintah Amazon Transcribe di AWS CLI Command Reference.

Memulai pekerjaan transkripsi baru

Untuk memulai transkripsi baru, gunakan `start-transcription-job` perintah.

1. Di jendela terminal, ketik berikut ini:

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

   A '>' muncul di baris berikutnya, dan Anda sekarang dapat terus menambahkan parameter yang diperlukan, seperti yang dijelaskan pada langkah berikutnya.

   Anda juga dapat menghilangkan '\' dan menambahkan semua parameter, memisahkan masing-masing dengan spasi.

2. Dengan `start-transcription-job` perintah, Anda harus menyertakan`region`,, `transcription-job-name``media`, dan salah satu `language-code` atau`identify-language`.

   Jika Anda ingin menentukan lokasi keluaran, sertakan `output-bucket-name` dalam permintaan Anda; jika Anda ingin menentukan sub-folder dari bucket keluaran yang ditentukan, sertakan `output-key` juga.

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

Jika menambahkan semua parameter, permintaan ini terlihat seperti:

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

Jika Anda memilih untuk tidak menentukan bucket keluaran yang digunakan`output-bucket-name`, Amazon Transcribe tempatkan keluaran transkripsi Anda dalam bucket yang dikelola layanan. Transkrip yang disimpan dalam bucket yang dikelola layanan akan kedaluwarsa setelah 90 hari.

Amazon Transcribe merespons dengan:

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

Pekerjaan transkripsi Anda berhasil jika `TranscriptionJobStatus`berubah dari `IN_PROGRESS` ke`COMPLETED`. Untuk melihat yang diperbarui `TranscriptionJobStatus`, gunakan `list-transcription-job` perintah `get-transcription-job` or, seperti yang ditunjukkan pada bagian berikut.

Mendapatkan status pekerjaan transkripsi

Untuk mendapatkan informasi tentang pekerjaan transkripsi Anda, gunakan `get-transcription-job` perintah.

Satu-satunya parameter yang diperlukan untuk perintah ini adalah Wilayah AWS di mana pekerjaan itu berada dan nama pekerjaan.

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

Amazon Transcribe merespons dengan:

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

Jika Anda telah memilih Amazon S3 bucket Anda sendiri untuk keluaran transkripsi Anda, bucket ini terdaftar dengan`TranscriptFileUri`. Jika Anda telah memilih bucket yang dikelola layanan, URI sementara disediakan; gunakan URI ini untuk mengunduh transkrip Anda.

> **Note**
>
> Sementara URIs untuk Amazon S3 bucket yang dikelola layanan hanya berlaku selama 15 menit. Jika Anda mendapatkan `AccesDenied` kesalahan saat menggunakan URI, jalankan `get-transcription-job` permintaan lagi untuk mendapatkan URI sementara baru.

Daftar pekerjaan transkripsi Anda

Untuk membuat daftar semua pekerjaan transkripsi Anda di tempat tertentu Wilayah AWS, gunakan `list-transcription-jobs` perintah.

Satu-satunya parameter yang diperlukan untuk perintah ini adalah Wilayah AWS di mana pekerjaan transkripsi Anda berada.

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

Amazon Transcribe merespons dengan:

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

Menghapus pekerjaan transkripsi Anda

Untuk menghapus pekerjaan transkripsi Anda, gunakan `delete-transcription-job` perintah.

Satu-satunya parameter yang diperlukan untuk perintah ini adalah Wilayah AWS di mana pekerjaan itu berada dan nama pekerjaan.

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

Untuk mengonfirmasi permintaan penghapusan Anda berhasil, Anda dapat menjalankan `list-transcription-jobs` perintah. Pekerjaan Anda seharusnya tidak lagi muncul dalam daftar.

Mentranskripsikan dengan AWS SDKs

Anda dapat menggunakan SDKs transkripsi batch dan streaming. Jika Anda mentranskripsikan file media yang terletak di Amazon S3 ember, Anda sedang melakukan transkripsi batch. Jika Anda menyalin aliran data audio waktu nyata, Anda melakukan transkripsi streaming.

Untuk daftar bahasa pemrograman yang dapat Anda gunakan Amazon Transcribe, lihatBahasa pemrograman yang didukung. Perhatikan bahwa transkripsi streaming tidak didukung dengan semua AWS SDKs. Untuk melihat format media yang didukung serta persyaratan dan kendala media lainnya, lihat. Input dan output data

Untuk informasi selengkapnya tentang semua alat yang tersedia AWS SDKs dan alat pembangun, lihat Alat untuk Dibangun AWS.

> **Tip**
>
> Untuk contoh tambahan menggunakan AWS SDKs, termasuk contoh khusus fitur, skenario, dan lintas layanan, lihat bagian ini. Contoh kode untuk Amazon Transcribe menggunakan AWS SDKs
>
> Anda juga dapat menemukan contoh kode SDK di GitHub repositori ini:
>
> - AWS Contoh Kode
> - Amazon Transcribe Contoh

Transkripsi Batch

Anda dapat membuat transkripsi batch menggunakan URI file media yang terletak di Amazon S3 bucket. Jika Anda tidak yakin cara membuat Amazon S3 bucket atau mengunggah file, lihat Buat bucket S3 pertama Anda dan Unggah objek ke bucket Anda.

Java

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

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

    public static void main(String args[]) {

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

        String transcriptionJobName = "my-first-transcription-job";
        String mediaType = "flac"; // can be other types
        Media myMedia = Media.builder()
                .mediaFileUri("s3://amzn-s3-demo-bucket/my-input-files/my-media-
file.flac")
                .build();

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

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

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

```
        System.out.println(startJobResponse.transcriptionJob());

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

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

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

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

}
```

JavaScript

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

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

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

async function startTranscriptionRequest() {
```

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

startTranscriptionRequest();
```

Python

```
import time
import boto3

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

    max_tries = 60
    while max_tries > 0:
        max_tries -= 1
        job = transcribe_client.get_transcription_job(TranscriptionJobName =
 job_name)
        job_status = job['TranscriptionJob']['TranscriptionJobStatus']
        if job_status in ['COMPLETED', 'FAILED']:
            print(f"Job {job_name} is {job_status}.")
            if job_status == 'COMPLETED':
                print(
                    f"Download the transcript from\n"
```

```
                    f"\t{job['TranscriptionJob']['Transcript']
['TranscriptFileUri']}.")
            break
        else:
            print(f"Waiting for {job_name}. Current status is {job_status}.")
        time.sleep(10)


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


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

Transkripsi streaming

Anda dapat membuat transkripsi streaming menggunakan file media streaming atau streaming media langsung.

Perhatikan bahwa standar AWS SDK untuk Python (Boto3) tidak didukung untuk Amazon Transcribe streaming. Untuk memulai transkripsi streaming menggunakan Python, gunakan SDK Python async ini untuk. Amazon Transcribe

Java

Contoh berikut adalah program Java yang mentranskripsikan audio streaming.

Untuk menjalankan contoh ini, perhatikan hal berikut:

- Anda harus menggunakan AWS SDK for Java 2.x.
- Klien harus menggunakan Java 1.8 agar kompatibel dengan AWS SDK for Java 2.x.
- Tingkat sampel yang Anda tentukan harus sesuai dengan laju sampel aktual dari aliran audio Anda.

Lihat juga: Coba lagi klien untuk Amazon Transcribe streaming (Java SDK). Kode ini mengelola koneksi ke Amazon Transcribe dan mencoba lagi mengirim data ketika ada kesalahan pada

koneksi. Misalnya, jika ada kesalahan sementara pada jaringan, klien ini mengirim ulang permintaan yang gagal.

```
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

Contoh berikut adalah program Python yang mentranskripsikan audio streaming.

Untuk menjalankan contoh ini, perhatikan hal berikut:

- Anda harus menggunakan SDK ini untuk Python.
- Tingkat sampel yang Anda tentukan harus sesuai dengan laju sampel aktual dari aliran audio Anda.

```
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++

Lihat Bab contoh Kode untuk contoh streaming C++ SDK.

Menggunakan layanan ini dengan AWS SDK

AWS kit pengembangan perangkat lunak (SDKs) tersedia untuk banyak bahasa pemrograman populer. Setiap SDK menyediakan API, contoh kode, dan dokumentasi yang memudahkan developer untuk membangun aplikasi dalam bahasa pilihan mereka.

Dokumentasi SDK	Contoh kode
AWS SDK untuk C++	AWS SDK untuk C++ contoh kode
AWS CLI	AWS CLI contoh kode
AWS SDK untuk Go	AWS SDK untuk Go contoh kode
AWS SDK untuk Java	AWS SDK untuk Java contoh kode
AWS SDK untuk JavaScript	AWS SDK untuk JavaScript contoh kode
AWS SDK untuk Kotlin	AWS SDK untuk Kotlin contoh kode
AWS SDK untuk .NET	AWS SDK untuk .NET contoh kode
AWS SDK untuk PHP	AWS SDK untuk PHP contoh kode
Alat AWS untuk PowerShell	Alat AWS untuk PowerShell contoh kode
AWS SDK untuk Python (Boto3)	AWS SDK untuk Python (Boto3) contoh kode
AWS SDK untuk Ruby	AWS SDK untuk Ruby contoh kode
AWS SDK for Rust	AWS SDK for Rust contoh kode
AWS SDK untuk SAP ABAP	AWS SDK untuk SAP ABAP contoh kode
AWS SDK for Swift	AWS SDK for Swift contoh kode

Untuk contoh khusus untuk layanan ini, lihat Contoh kode untuk Amazon Transcribe menggunakan AWS SDKs.

> **Ketersediaan contoh**
>
> Tidak dapat menemukan apa yang Anda butuhkan? Minta contoh kode menggunakan tautan Berikan umpan balik di bagian bawah halaman ini.

Mentranskripsikan dengan HTTP atau WebSockets

Amazon Transcribe mendukung HTTP untuk transkripsi batch (HTTP/1.1) dan streaming (HTTP/2). WebSockets didukung untuk transkripsi streaming.

Jika Anda menyalin file media yang terletak di Amazon S3 ember, Anda sedang melakukan transkripsi batch. Jika Anda menyalin aliran data audio waktu nyata, Anda melakukan transkripsi streaming.

Baik HTTP dan WebSockets mengharuskan Anda untuk mengautentikasi permintaan Anda menggunakan header AWS Signature Version 4. Lihat permintaan AWS API Penandatanganan untuk informasi selengkapnya.

Transkripsi Batch

Anda dapat membuat permintaan HTTP batch menggunakan header berikut:

- host
- x-amz-target
- tipe konten
- x-amz-content-sha256
- x-amz-date
- otorisasi

Berikut adalah contoh `StartTranscriptionJob` permintaan:

```
POST /transcribe HTTP/1.1
host: transcribe.us-west-2.amazonaws.com
x-amz-target: com.amazonaws.transcribe.Transcribe.StartTranscriptionJob
```

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

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

Operasi dan parameter tambahan tercantum dalam Referensi API; parameter yang umum untuk semua operasi AWS API tercantum di bagian Parameter Umum. Elemen tanda tangan lainnya dirinci dalam Elemen permintaan AWS Signature Version 4.

Transkripsi streaming

Transkripsi streaming menggunakan HTTP/2 dan lebih WebSockets terlibat daripada menggunakan. SDKs Sebaiknya tinjau Menyiapkan transkripsi streaming bagian ini sebelum menyiapkan streaming pertama Anda.

Untuk informasi lebih lanjut tentang metode ini, lihat Menyiapkan aliran HTTP/2 atauMenyiapkan WebSocket aliran.

> **Note**
>
> Kami sangat menyarankan menggunakan SDK untuk streaming transkripsi. Untuk daftar yang didukung SDKs, lihatBahasa pemrograman yang didukung.

Mentranskripsikan audio streaming

Dengan menggunakan Amazon Transcribe streaming, Anda dapat menghasilkan transkripsi waktu nyata untuk konten media Anda. Tidak seperti transkripsi batch, yang melibatkan pengunggahan file media, media streaming dikirimkan Amazon Transcribe secara real time. Amazon Transcribe kemudian mengembalikan transkrip, juga secara real time.

Streaming dapat mencakup media pra-rekaman (film, musik, dan podcast) dan media real-time (siaran berita langsung). Kasus penggunaan streaming umum Amazon Transcribe termasuk teks tertutup langsung untuk acara olahraga dan pemantauan audio call center secara real-time.

Konten streaming dikirimkan sebagai serangkaian paket data berurutan, atau 'potongan', yang mentranskripsikan secara instan. Amazon Transcribe Keuntungan menggunakan streaming melalui batch termasuk speech-to-text kemampuan real-time dalam aplikasi Anda dan waktu transkripsi yang lebih cepat. Namun, peningkatan kecepatan ini mungkin memiliki keterbatasan akurasi dalam beberapa kasus.

Amazon Transcribe menawarkan opsi berikut untuk streaming:

- SDKs(lebih disukai)
- HTTP/2
- WebSockets
- AWS Management Console

Untuk mentranskripsikan audio streaming di AWS Management Console, bicaralah ke mikrofon komputer Anda.

> **Tip**
>
> Untuk contoh kode SDK, lihat repositori AWS Sampel di. GitHub

Format audio yang didukung untuk transkripsi streaming adalah:

- FLAC
- Audio yang dikodekan OPUS dalam wadah Ogg
- PCM (hanya menandatangani format audio 16-bit little-endian, yang tidak termasuk WAV)

Format lossless (FLAC atau PCM) direkomendasikan.

> **Note**
>
> Transkripsi streaming tidak didukung dengan semua bahasa. Lihat kolom 'Input data' dalam tabel bahasa yang didukung untuk detailnya.

Untuk melihat ketersediaan Amazon Transcribe Wilayah untuk transkripsi streaming, lihat: Amazon Transcribe Titik Akhir dan Kuota.

Praktik terbaik

Rekomendasi berikut meningkatkan efisiensi transkripsi streaming:

- Jika memungkinkan, gunakan audio yang disandikan PCM.
- Pastikan streaming Anda sedekat mungkin dengan waktu nyata.
- Latensi tergantung pada ukuran potongan audio Anda. Jika Anda dapat menentukan ukuran potongan dengan jenis audio Anda (seperti dengan PCM), atur setiap potongan menjadi antara 50 ms dan 200 ms. Anda dapat menghitung ukuran potongan audio dengan rumus berikut:

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

- Gunakan ukuran potongan yang seragam.
- Pastikan Anda menentukan jumlah saluran audio dengan benar.
- Dengan audio PCM saluran tunggal, setiap sampel terdiri dari dua byte, sehingga setiap potongan harus terdiri dari jumlah byte genap.
- Dengan audio PCM saluran ganda, setiap sampel terdiri dari empat byte, sehingga setiap potongan harus kelipatan 4 byte.
- Ketika streaming audio Anda tidak berisi ucapan, encode dan kirim jumlah keheningan yang sama. Misalnya, diam untuk PCM adalah aliran nol byte.
- Pastikan Anda menentukan laju pengambilan sampel yang benar untuk audio Anda. Jika memungkinkan, rekam pada laju pengambilan sampel 16.000 Hz; ini memberikan kompromi terbaik antara kualitas dan volume data yang dikirim melalui jaringan. Perhatikan bahwa sebagian besar mikrofon kelas atas merekam pada 44.100 Hz atau 48.000 Hz.

Streaming dan hasil sebagian

Karena streaming bekerja secara real time, transkrip diproduksi dalam hasil sebagian. Amazon Transcribe memecah aliran audio yang masuk berdasarkan segmen ucapan alami, seperti perubahan speaker atau jeda dalam audio. Transkripsi dikembalikan ke aplikasi Anda dalam aliran peristiwa transkripsi, dengan setiap respons berisi lebih banyak ucapan yang ditranskripsi hingga seluruh segmen ditranskripsi.

Perkiraan ini ditunjukkan dalam blok kode berikut. Anda dapat melihat proses ini dalam tindakan dengan masuk ke AWS Management Console, memilih Transkripsi waktu nyata, dan berbicara ke mikrofon Anda. Tonton panel keluaran Transkripsi saat Anda berbicara.

Dalam contoh ini, setiap baris adalah hasil sebagian dari segmen audio.

```
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.
```

Hasil sebagian ini ada dalam output transkripsi Anda di dalam `Results`objek. Juga di blok objek ini adalah IsPartialbidang. Jika bidang ini benar, segmen transkripsi Anda belum lengkap. Anda dapat melihat perbedaan antara segmen yang tidak lengkap dan lengkap di bawah ini:

```
"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
```

Setiap kata dalam segmen lengkap memiliki skor kepercayaan terkait, yang merupakan nilai antara `0` dan`1`. Nilai yang lebih besar menunjukkan kemungkinan yang lebih besar bahwa kata tersebut ditranskripsi dengan benar.

> **Tip**
>
> Segmen `StartTime` dan `EndTime` audio dapat digunakan untuk menyinkronkan output transkripsi dengan dialog video.

Jika Anda menjalankan aplikasi yang membutuhkan latensi rendah, Anda mungkin ingin menggunakan stabilisasi hasil sebagian.

Stabilisasi hasil sebagian

Amazon Transcribe mulai mengembalikan hasil transkripsi segera setelah Anda mulai streaming audio Anda. Ini mengembalikan hasil sebagian ini secara bertahap sampai menghasilkan hasil akhir pada tingkat segmen ucapan alami. Segmen ucapan alami adalah ucapan berkelanjutan yang berisi jeda atau perubahan pembicara.

Amazon Transcribe terus mengeluarkan sebagian hasil sampai menghasilkan hasil transkripsi akhir untuk segmen bicara. Karena pengenalan suara dapat merevisi kata-kata karena memperoleh lebih banyak konteks, transkripsi streaming dapat sedikit berubah dengan setiap keluaran hasil parsi baru.

Proses ini memberi Anda dua opsi untuk setiap segmen pidato:

- Tunggu segmen jadi
- Gunakan hasil sebagian segmen

Stabilisasi hasil sebagian mengubah cara Amazon Transcribe menghasilkan hasil transkripsi akhir untuk setiap segmen lengkap. Saat diaktifkan, hanya beberapa kata terakhir dari hasil sebagian yang dapat berubah. Karena itu, akurasi transkripsi mungkin terpengaruh. Namun, transkrip Anda

dikembalikan lebih cepat daripada tanpa stabilisasi hasil sebagian. Pengurangan latensi ini mungkin bermanfaat saat membuat subtitling video atau membuat teks untuk streaming langsung.

Contoh berikut menunjukkan bagaimana aliran audio yang sama ditangani ketika stabilisasi hasil sebagian tidak diaktifkan dan kapan itu. Perhatikan bahwa Anda dapat mengatur tingkat stabilitas ke rendah, sedang, atau tinggi. Stabilitas rendah memberikan akurasi tertinggi. Stabilitas tinggi mentranskripsi lebih cepat, tetapi dengan akurasi yang sedikit lebih rendah.

“Transkrip”:	"EndTime":	"IsPartial":
Stabilisasi hasil sebagian tidak diaktifkan		
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. The Amazon is the largest rainforest on the planet. 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
Stabilisasi hasil sebagian diaktifkan (stabilitas tinggi)		
The	0.515	true

“Transkrip”:	"EndTime":	"IsPartial":
The The Amazon. The Amazon is The Amazon is the large The Amazon is the largest The Amazon is the largest rainfall. The Amazon is the largest rain forest. The Amazon is the largest rain forest on The Amazon is the largest rain forest on the planet. The Amazon is the largest rain forest on the planet. The Amazon is the largest rain forest on the planet. The Amazon is the largest rain forest on the planet.	1.015 1.515 2.015 2.515 3.015 3.515 4.015 4.515 5.015 5.515 6.015 6.335 6.335	true true true true true true true true true true true true false

Saat Anda mengaktifkan stabilisasi hasil sebagian, Amazon Transcribe gunakan `Stable` bidang untuk menunjukkan apakah suatu item stabil, di mana 'item' mengacu pada kata yang ditranskripsi atau tanda baca. Nilai untuk `Stable` adalah `true` atau`false`. Item yang ditandai sebagai `false` (tidak stabil) lebih mungkin berubah saat segmen Anda ditranskripsi. Sebaliknya, item yang ditandai sebagai `true` (stabil) tidak akan berubah.

Anda dapat memilih untuk membuat kata-kata yang tidak stabil sehingga teks Anda sejajar dengan ucapan. Bahkan jika teks sedikit berubah saat konteks ditambahkan, ini adalah pengalaman pengguna yang lebih baik daripada semburan teks berkala, yang mungkin atau mungkin tidak selaras dengan ucapan.

Anda juga dapat memilih untuk menampilkan kata-kata yang tidak stabil dalam format yang berbeda, seperti miring, untuk menunjukkan kepada pemirsa bahwa kata-kata ini dapat berubah. Menampilkan sebagian hasil membatasi jumlah teks yang ditampilkan pada waktu tertentu. Ini bisa menjadi penting ketika Anda berurusan dengan kendala ruang, seperti dengan teks video.

> Menyelam lebih dalam dengan Blog AWS Machine Learning
>
> Untuk mempelajari lebih lanjut tentang meningkatkan akurasi dengan transkripsi waktu nyata, lihat:
>
> - Tingkatkan pengalaman transkripsi streaming dengan stabilisasi hasil Amazon Transcribe sebagian
> - “Apa itu?” Meningkatkan akurasi subtitle untuk siaran langsung menggunakan Amazon Transcribe

Output contoh stabilisasi hasil sebagian

Contoh output berikut menunjukkan `Stable` flag untuk segmen yang tidak lengkap ()`"IsPartial": true`. Anda dapat melihat bahwa kata-kata "to" dan "Amazon" tidak stabil dan karenanya dapat berubah sebelum segmen diselesaikan.


```
"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
    }
]
}
```

Menyiapkan transkripsi streaming

Bagian ini meluas di bagian streaming utama. Ini dimaksudkan untuk memberikan informasi bagi pengguna yang ingin mengatur aliran mereka dengan HTTP/2 atau WebSockets secara langsung, bukan dengan SDK. AWS Informasi di bagian ini juga dapat digunakan untuk membangun SDK Anda sendiri.

> **Important**
>
> Kami sangat menyarankan menggunakan SDKs daripada menggunakan HTTP/2 dan WebSockets langsung. SDKs adalah metode paling sederhana dan paling dapat

diandalkan untuk menyalin aliran data. Untuk memulai streaming menggunakan AWS SDK, lihatMentranskripsikan dengan AWS SDKs.

Menyiapkan aliran HTTP/2

Komponen kunci untuk protokol HTTP/2 untuk streaming permintaan transkripsi adalah: Amazon Transcribe

- Bingkai header. Ini berisi header HTTP/2 untuk permintaan Anda, dan tanda tangan di header otorisasi yang Amazon Transcribe digunakan sebagai tanda tangan benih untuk menandatangani bingkai data.
- Satu atau beberapa frame pesan dalam pengkodean aliran acara yang berisi metadata dan byte audio mentah.
- Bingkai akhir. Ini adalah pesan yang ditandatangani dalam pengkodean aliran acara dengan badan kosong.

> **Note**
>
> Amazon Transcribe hanya mendukung satu aliran per sesi HTTP/2. Jika Anda mencoba menggunakan beberapa aliran, permintaan transkripsi Anda gagal.

1. Lampirkan kebijakan berikut ke IAM peran yang membuat permintaan. Lihat Menambahkan IAM kebijakan untuk informasi selengkapnya.

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

2. Untuk memulai sesi, kirim permintaan HTTP/2 ke. Amazon Transcribe

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

Operasi dan parameter tambahan tercantum dalam Referensi API; parameter yang umum untuk semua operasi AWS API tercantum di bagian Parameter Umum.

Amazon Transcribe mengirimkan respons berikut:

```
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. Buat acara audio yang berisi data audio Anda. Gabungkan header—dijelaskan dalam tabel berikut—dengan potongan byte audio dalam pesan yang disandikan peristiwa. Untuk membuat payload pesan acara, gunakan buffer dalam format raw-byte.

Nama header panjang byte	Nama header (string)	Jenis nilai header	Nilai panjang string byte	Nilai string (UTF-8)
13	:tipe konten	7	24	aplikasi/oktet-aliran
11	:tipe acara	7	10	AudioEvent

Nama header panjang byte	Nama header (string)	Jenis nilai header	Nilai panjang string byte	Nilai string (UTF-8)
13	:tipe pesan	7	5	kejadian

Data biner dalam permintaan contoh ini dikodekan base64. Dalam permintaan aktual, data adalah byte mentah.

```
:content-type: "application/vnd.amazon.eventstream"
:event-type: "AudioEvent"
:message-type: "event"
UklGRjzxPQBXQVZFZm10IBAAAAABAAEAgD4AAAB9AAACABAAZGF0YVTwPQAAAAAAAAAAAAAAAAD//wIA/
f8EAA==
```

4. Buat pesan audio yang berisi data audio Anda.

   a. Bingkai data pesan audio Anda berisi header pengkodean peristiwa yang menyertakan tanggal saat ini dan tanda tangan untuk potongan audio dan acara audio.

Nama header panjang byte	Nama header (string)	Jenis nilai header	Nilai panjang string byte	Nilai
16	:bongkahan tanda tangan	6	bervariasi	tanda tangan yang dihasilkan
5	:tanggal	8	8	timestamp

Data biner dalam permintaan ini dikodekan base64. Dalam permintaan aktual, data adalah byte mentah.

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

AAAA0gAAAIKVoRFcTTcjb250ZW50LXR5cGUHABhhcHBsaWNhdGlvbi9vY3RldC1zdHJlYW0LOmV2ZW50LXR5
cGUHAApBdWRpb0V2ZW50DTptZXNzYWdlLXR5cGUHAAVldmVudAxDb256ZW50LVR5cGUHABphcHBsaWNhdGlv
bi94LWFtei1qc29uLTEuMVJJRkY88T0AV0FWRWZtdCAQAAAAAQABAIA
+AAAAfQAAAgAQAGRhdGFU8D0AAAAA
AAAAAAAAAAAA//8CAP3/BAC7QLFf
```

b. Buat string untuk ditandatangani, seperti yang diuraikan dalam Buat string untuk menandatangani Signature Version 4. String Anda mengikuti format ini:

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

- DateTime: Tanggal dan waktu tanda tangan dibuat. Formatnya adalah YYYYMMDDTHHMMSSZ, di mana YYYYY=year, mm=month, dd=day, hh=hour, mm=minute, ss=seconds, dan 'T' dan 'Z' adalah karakter tetap. Untuk informasi selengkapnya, lihat Menangani Tanggal di Tanda Tangan Versi 4.
- Keypath: Lingkup tanda tangan dalam format`date/region/service/aws4_request`. Misalnya, `20220127/us-west-2/transcribe/aws4_request`.
- Hex: Fungsi yang mengkodekan input ke dalam representasi heksadesimal.
- PriorSignature: Tanda tangan untuk frame sebelumnya. Untuk frame data pertama, gunakan tanda tangan dari frame header.
- HexHash: Fungsi yang pertama membuat hash SHA-256 dari inputnya dan kemudian menggunakan fungsi Hex untuk menyandikan hash.
- nonSignatureHeaders: DateTime Header dikodekan sebagai string.
- payload: Buffer byte yang berisi data peristiwa audio.

c. Dapatkan kunci penandatanganan dari kunci akses AWS rahasia Anda dan gunakan untuk menandatangani. `stringToSign` Untuk tingkat perlindungan yang lebih besar, kunci turunan khusus untuk tanggal, layanan, dan Wilayah AWS. Untuk informasi selengkapnya, lihat Menghitung tanda tangan untuk Versi AWS Tanda Tangan 4.

   Pastikan Anda menerapkan `GetSignatureKey` fungsi untuk mendapatkan kunci penandatanganan Anda. Jika Anda belum mendapatkan kunci penandatanganan, lihat Contoh cara mendapatkan kunci penandatanganan untuk Signature Version 4.

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

- HMACSHA256: Fungsi yang membuat tanda tangan menggunakan fungsi hash SHA-256.
- derivedSigningKey: Kunci penandatanganan Signature Version 4.
- stringToSign: String yang Anda hitung untuk bingkai data.

Setelah menghitung tanda tangan untuk bingkai data, buat buffer byte yang berisi tanggal, tanda tangan, dan muatan peristiwa audio. Kirim array byte Amazon Transcribe untuk transkripsi.

5. Untuk menunjukkan aliran audio selesai, kirim bingkai akhir (bingkai data kosong) yang hanya berisi tanggal dan tanda tangan. Anda membangun bingkai akhir ini dengan cara yang sama seperti Anda membangun bingkai data.

   Amazon Transcribe merespons dengan aliran peristiwa transkripsi, dikirim ke aplikasi Anda. Respons ini adalah aliran acara yang dikodekan. Ini berisi pendahuluan standar dan header berikut.

Nama header panjang byte	Nama header (string)	Jenis nilai header	Nilai panjang string byte	Nilai string (UTF-8)
13	:tipe konten	7	16	aplikasi/json
11	:tipe acara	7	15	TranscriptEvent
13	:tipe pesan	7	5	kejadian

   Peristiwa dikirim dalam format raw-byte. Dalam contoh ini, byte dikodekan base64.

```
AAAAUwAAAEP1RHpYBTpkYXRlCAAAAWiXUkMLEDpjaHVuay1zaWduYXR1cmUGACCt6Zy+uymwEK2SrLp/
zVBI
5eGn83jdBwCaRUBJA+eaDafqjqI=
```

   Untuk melihat hasil transkripsi, dekode byte mentah menggunakan pengkodean aliran peristiwa.

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

```
:event-type: "TranscriptEvent"
:message-type: "event"

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

6. Untuk mengakhiri streaming Anda, kirim acara audio kosong ke Amazon Transcribe. Buat acara audio persis seperti yang lain, kecuali dengan muatan kosong. Tanda tangani acara dan sertakan tanda tangan di `:chunk-signature` header, sebagai berikut:

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

Menangani kesalahan streaming HTTP/2

Jika terjadi kesalahan saat memproses aliran media Anda, Amazon Transcribe kirimkan respons pengecualian. Responsnya adalah aliran acara yang dikodekan.

Respons berisi pendahuluan standar dan header berikut:

Nama header panjang byte	Nama header (string)	Jenis nilai header	Nilai panjang string byte	Nilai string (UTF-8)
13	:tipe konten	7	16	aplikasi/json
11	:tipe acara	7	19	BadRequestException
13	:tipe pesan	7	9	pengecualian

Ketika respon pengecualian diterjemahkan, itu berisi informasi berikut:

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

```
:event-type: "BadRequestException"
:message-type: "exception"

Exception message
```

Menyiapkan WebSocket aliran

Komponen kunci untuk WebSocketprotokol untuk streaming permintaan transkripsi Amazon Transcribe adalah:

- Permintaan upgrade. Ini berisi parameter kueri untuk permintaan Anda, dan tanda tangan yang Amazon Transcribe digunakan sebagai tanda tangan benih untuk menandatangani bingkai data.
- Satu atau beberapa frame pesan dalam pengkodean aliran acara yang berisi metadata dan byte audio mentah.
- Bingkai akhir. Ini adalah pesan yang ditandatangani dalam pengkodean aliran acara dengan badan kosong.

> **Note**
>
> Amazon Transcribe hanya mendukung satu aliran per WebSocket sesi. Jika Anda mencoba menggunakan beberapa aliran, permintaan transkripsi Anda gagal.

1. Lampirkan kebijakan berikut ke IAM peran yang membuat permintaan. Lihat Menambahkan IAM kebijakan untuk informasi selengkapnya.

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

2. Untuk memulai sesi, buat URL presigned dalam format berikut. Jeda baris telah ditambahkan untuk keterbacaan.

```
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**
>
> Nilai maksimum untuk `X-Amz-Expires` adalah 300 (5 menit).

Operasi dan parameter tambahan tercantum dalam Referensi API; parameter yang umum untuk semua operasi AWS API tercantum di bagian Parameter Umum.

Untuk membuat URL permintaan Anda dan membuat tanda tangan Versi Tanda Tangan 4, lihat langkah-langkah berikut. Contohnya dalam pseudocode.

a. Membuat permintaan kanonik. Permintaan kanonik adalah string yang menyertakan informasi dari permintaan Anda dalam format standar. Ini memastikan bahwa ketika AWS menerima permintaan, itu dapat menghitung tanda tangan yang sama yang Anda buat untuk URL Anda. Untuk informasi selengkapnya, lihat Membuat Permintaan Kanonik untuk Tanda Tangan Versi 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. Buat URI kanonik, yang merupakan bagian dari URI antara domain dan string kueri.

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

c. Buat header kanonik dan header yang ditandatangani. Perhatikan trailing \n di header kanonik.

- Tambahkan nama header huruf kecil diikuti dengan titik dua (:).
- Tambahkan daftar nilai yang dipisahkan koma untuk header tersebut. Jangan mengurutkan nilai dalam header yang memiliki beberapa nilai.
- Tambahkan baris baru (\n).

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

d. Cocokkan algoritma dengan algoritma hashing. Gunakan SHA-256.

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

e. Buat cakupan kredensi, yang mencakup kunci turunan untuk tanggal, Wilayah AWS, dan layanan. Misalnya, *20220127*/*us-west-2*/transcribe/aws4_request.

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

f. Buat string kueri kanonik. Nilai string kueri harus dikodekan URI dan diurutkan berdasarkan nama.

- Urutkan nama parameter dengan titik kode karakter dalam urutan naik. Parameter dengan nama duplikat harus diurutkan berdasarkan nilai. Misalnya, nama parameter yang dimulai dengan huruf besar F mendahului nama parameter yang dimulai dengan huruf kecil b.

- Jangan mengkodekan URI salah satu karakter tanpa syarat yang ditentukan RFC 3986: A-Z, a-z, 0-9, tanda hubung (-), garis bawah (_), titik (.), dan tilde (~).
- Persen-encode semua karakter lain dengan %XY, di mana X dan Y adalah karakter heksadesimal (0-9 dan huruf besar A-F). Misalnya, karakter spasi harus dikodekan sebagai% 20 (jangan sertakan '+', seperti yang dilakukan beberapa skema pengkodean); karakter UTF-8 yang diperluas harus dalam bentuk %XY%ZA%BC.
- Enkode ganda karakter yang sama dengan (=) dalam nilai parameter.

```
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. Buat hash dari payload. Untuk GET permintaan, payload adalah string kosong.

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

h. Gabungkan elemen-elemen berikut untuk membuat permintaan kanonik.

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

3. Buat string untuk ditandatangani, yang berisi informasi meta tentang permintaan Anda. Anda menggunakan string untuk masuk ke langkah berikutnya ketika Anda menghitung tanda tangan permintaan. Untuk informasi selengkapnya, lihat Membuat String untuk Tanda Tangan Versi 4.

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

4. Hitung tanda tangannya. Untuk melakukan ini, dapatkan kunci penandatanganan dari kunci akses AWS rahasia Anda. Untuk tingkat perlindungan yang lebih besar, kunci turunan khusus untuk tanggal, layanan, dan Wilayah AWS. Gunakan kunci turunan ini untuk menandatangani permintaan. Untuk informasi selengkapnya, lihat Menghitung Tanda Tangan untuk AWS Tanda Tangan Versi 4.

   Pastikan Anda menerapkan `GetSignatureKey` fungsi untuk mendapatkan kunci penandatanganan Anda. Jika Anda belum mendapatkan kunci penandatanganan, lihat Contoh cara mendapatkan kunci penandatanganan untuk Signature Version 4.

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

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

   Fungsi ini `HMAC(key, data)` merupakan fungsi HMAC- SHA256 yang mengembalikan hasil dalam format biner.

5. Tambahkan informasi penandatanganan ke permintaan dan buat URL permintaan.

   Setelah Anda menghitung tanda tangan, tambahkan ke string kueri. Untuk informasi selengkapnya, lihat Menambahkan Tanda Tangan ke Permintaan.

   Pertama, tambahkan informasi otentikasi ke string kueri.

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

   Kedua, buat URL untuk permintaan tersebut.

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

   Gunakan URL permintaan dengan WebSocket pustaka Anda untuk membuat permintaan Amazon Transcribe.

6. Permintaan untuk Amazon Transcribe harus menyertakan header berikut. Biasanya header ini dikelola oleh pustaka WebSocket klien Anda.

   ```
   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. Saat Amazon Transcribe menerima WebSocket permintaan Anda, itu merespons dengan respons WebSocket peningkatan. Biasanya WebSocket pustaka Anda mengelola respons ini dan menyiapkan soket untuk komunikasi dengannya Amazon Transcribe.

   Berikut ini adalah tanggapan dari Amazon Transcribe. Jeda baris telah ditambahkan untuk keterbacaan.

```
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. Buat permintaan WebSocket streaming Anda.

   Setelah WebSocket koneksi dibuat, klien dapat mulai mengirim urutan bingkai audio, masing-masing dikodekan menggunakan pengkodean aliran acara.

   Setiap frame data berisi tiga header yang dikombinasikan dengan potongan byte audio mentah; tabel berikut menjelaskan header ini.

Nama header panjang byte	Nama header (string)	Jenis nilai header	Nilai panjang string byte	Nilai string (UTF-8)
13	:tipe konten	7	24	aplikasi/oktet-aliran
11	:tipe acara	7	10	AudioEvent
13	:tipe pesan	7	5	kejadian

9. Untuk mengakhiri aliran data, kirim potongan audio kosong dalam pesan yang disandikan aliran acara.

   Respons berisi aliran peristiwa yang dikodekan byte mentah dalam payload. Ini berisi pendahuluan standar dan header berikut.

Nama header panjang byte	Nama header (string)	Jenis nilai header	Nilai panjang string byte	Nilai string (UTF-8)
13	:tipe konten	7	16	aplikasi/json
11	:tipe acara	7	15	TranscriptEvent
13	:tipe pesan	7	5	kejadian

   Saat Anda memecahkan kode respons biner, Anda berakhir dengan struktur JSON yang berisi hasil transkripsi.

Menangani kesalahan WebSocket streaming

Jika pengecualian terjadi saat memproses permintaan Anda, Amazon Transcribe merespons dengan WebSocket bingkai terminal yang berisi respons yang dikodekan aliran peristiwa. Respons ini berisi header yang dijelaskan dalam tabel berikut; isi respons berisi pesan kesalahan deskriptif. Setelah mengirim respons pengecualian, Amazon Transcribe kirim bingkai dekat.

Nama header panjang byte	Nama header (string)	Jenis nilai header	Nilai panjang string byte	Nilai string (UTF-8)
13	:tipe konten	7	16	aplikasi/json
15	:jenis pengecualian	7	bervariasi	bervariasi, lihat di bawah
13	:tipe pesan	7	9	pengecualian

`exception-type`Header berisi salah satu nilai berikut:

- **`BadRequestException`**: Ada kesalahan klien saat streaming dibuat, atau terjadi kesalahan saat streaming data. Pastikan klien Anda siap menerima data dan coba permintaan Anda lagi.
- **`InternalFailureException`**: Amazon Transcribe memiliki masalah selama jabat tangan dengan klien. Coba permintaan Anda lagi.
- **`LimitExceededException`**: Klien melebihi batas aliran bersamaan. Untuk informasi selengkapnya, lihat Amazon Transcribe Batasan-batasan. Kurangi jumlah stream yang Anda transkripsikan.
- **`UnrecognizedClientException`**: Permintaan WebSocket pemutakhiran ditandatangani dengan kunci akses atau kunci rahasia yang salah. Pastikan Anda membuat kunci akses dengan benar dan coba permintaan Anda lagi.

Amazon Transcribe juga dapat mengembalikan salah satu kesalahan layanan umum. Untuk daftar, lihat Kesalahan Umum.

Pengkodean aliran acara

Amazon Transcribe menggunakan format yang disebut pengkodean aliran peristiwa untuk transkripsi streaming.

Event stream encoding menyediakan komunikasi dua arah antara klien dan server. Bingkai data yang dikirim ke layanan Amazon Transcribe streaming dikodekan dalam format ini. Respons dari Amazon Transcribe juga menggunakan pengkodean ini.

Setiap pesan terdiri dari dua bagian: pendahuluan dan data. Pendahuluan terdiri dari:

1. Total panjang byte pesan
2. Panjang byte gabungan dari semua header

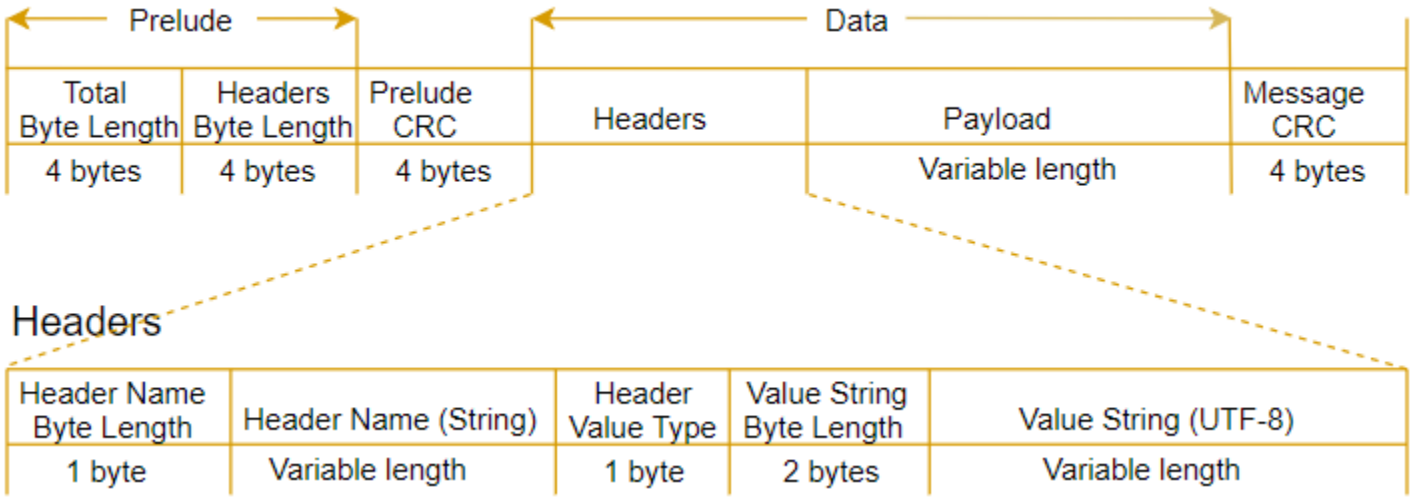
Bagian data terdiri dari:

1. Header
2. Muatan

Setiap bagian diakhiri dengan checksum pemeriksaan redundansi siklik (CRC) 4-byte big-endian integer. Pesan CRC checksum adalah untuk bagian pendahuluan dan bagian data. Amazon Transcribe menggunakan CRC32 (sering disebut sebagai GZIP CRC32) untuk menghitung keduanya CRCs. Untuk informasi selengkapnya CRC32, lihat Spesifikasi format file GZIP versi 4.3.

Total overhead pesan, termasuk pendahuluan dan kedua checksum, adalah 16 byte.

Diagram berikut menunjukkan komponen yang membentuk pesan dan header. Ada beberapa header per pesan.



Setiap pesan berisi komponen-komponen berikut:

- Prelude: Terdiri dari dua bidang 4-byte, dengan total tetap 8 byte.
  - 4 byte pertama: Panjang byte integer endian besar dari seluruh pesan, termasuk bidang panjang 4-byte ini.
  - 4 byte kedua: Panjang byte integer endian besar dari bagian 'header' pesan, tidak termasuk bidang panjang 'header' itu sendiri.

- Prelude CRC: Checksum CRC 4-byte untuk bagian awal pesan, tidak termasuk CRC itu sendiri. Pendahuluan memiliki CRC terpisah dari pesan CRC. Itu memastikan bahwa Amazon Transcribe dapat mendeteksi informasi panjang byte yang rusak dengan segera tanpa menyebabkan kesalahan, seperti buffer overruns.
- Header: Metadata yang menganotasi pesan; misalnya, jenis pesan dan jenis konten. Pesan memiliki beberapa header, yang merupakan pasangan key:value, di mana kuncinya adalah string UTF-8. Header dapat muncul dalam urutan apa pun di bagian 'header' pesan, dan setiap header hanya dapat muncul sekali.
- Payload: Konten audio yang akan ditranskripsikan.
- Pesan CRC: Checksum CRC 4-byte dari awal pesan hingga awal checksum. Artinya, semua yang ada dalam pesan kecuali CRC itu sendiri.

Bingkai header adalah bingkai otorisasi untuk transkripsi streaming. Amazon Transcribe menggunakan nilai header otorisasi sebagai benih untuk menghasilkan rantai header otorisasi untuk frame data dalam permintaan.

Setiap header berisi komponen-komponen berikut; ada beberapa header per frame.

- Nama header byte-length: Panjang byte dari nama header.
- Nama header: Nama header yang menunjukkan jenis header. Untuk nilai yang valid, lihat deskripsi bingkai berikut.
- Jenis nilai header: Angka yang menunjukkan nilai header. Daftar berikut menunjukkan nilai yang mungkin untuk header dan apa yang mereka tunjukkan.
  - 0— BENAR
  - 1— SALAH
  - 2— BYTE
  - 3— PENDEK
  - 4— BILANGAN BULAT
  - 5— PANJANG
  - 6— ARRAY BYTE
  - 7— STRING
  - 8— STEMPEL WAKTU
  - 9— UUID
- Nilai panjang string byte: Panjang byte dari string nilai header.

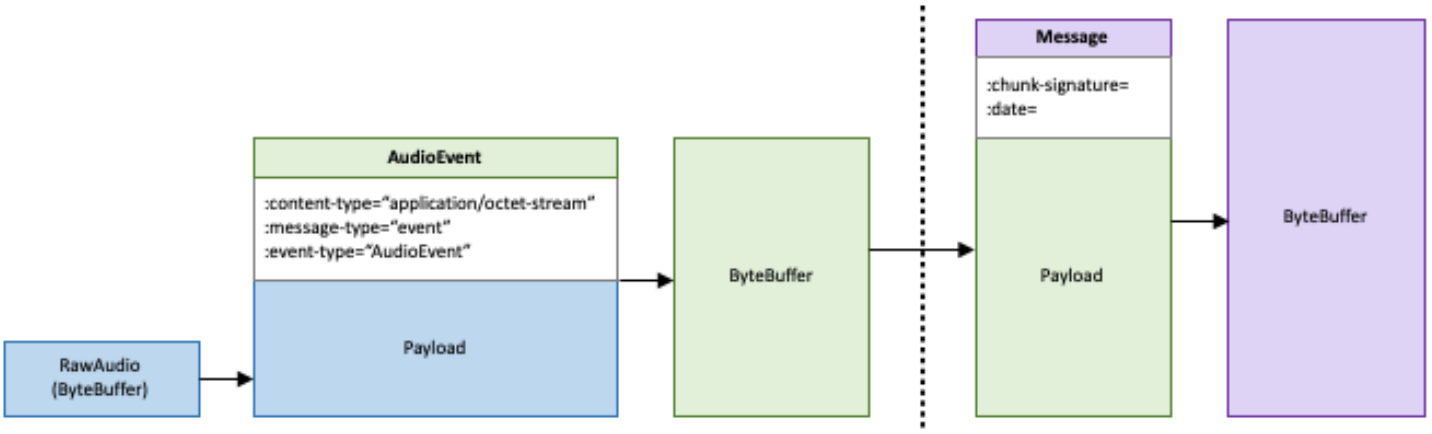
- Nilai header: Nilai string header. Nilai yang valid untuk bidang ini tergantung pada jenis header. Lihat Menyiapkan aliran HTTP/2 atau Menyiapkan WebSocket aliran untuk informasi lebih lanjut.

Bingkai data

Setiap permintaan streaming berisi satu atau lebih bingkai data. Ada dua langkah untuk membuat bingkai data:

1. Gabungkan data audio mentah dengan metadata untuk membuat muatan permintaan Anda.
2. Gabungkan payload dengan tanda tangan untuk membentuk pesan acara yang dikirim ke Amazon Transcribe.

Diagram berikut menunjukkan bagaimana inii bekerja.



Antrian pekerjaan

Dengan menggunakan antrian pekerjaan, Anda dapat mengirimkan lebih banyak permintaan pekerjaan transkripsi daripada yang dapat diproses secara bersamaan. Tanpa antrian pekerjaan, setelah Anda mencapai kuota permintaan bersamaan yang diizinkan, Anda harus menunggu hingga satu atau beberapa permintaan selesai sebelum mengirimkan permintaan baru.

Antrian pekerjaan bersifat opsional untuk pekerjaan transkripsi dan permintaan pekerjaan analitik pasca-panggilan.

Jika Anda mengaktifkan antrian pekerjaan, Amazon Transcribe buat antrian yang berisi semua permintaan yang melebihi batas Anda. Segera setelah permintaan selesai, permintaan baru ditarik dari antrian Anda dan diproses. Permintaan antrian diproses dalam urutan FIFO (masuk pertama, keluar pertama).

Anda dapat menambahkan hingga 10.000 pekerjaan ke antrian Anda. Jika Anda melebihi batas ini, Anda mendapatkan `LimitExceededConcurrentJobException` kesalahan. Untuk mempertahankan kinerja optimal, Amazon Transcribe hanya gunakan hingga 90 persen dari kuota Anda (rasio bandwidth 0,9) untuk memproses pekerjaan antrian. Perhatikan bahwa ini adalah nilai default yang dapat ditingkatkan berdasarkan permintaan.

> **Tip**
>
> Anda dapat menemukan daftar batas default dan kuota untuk Amazon Transcribe sumber daya di Referensi AWS Umum. Beberapa default ini dapat ditingkatkan berdasarkan permintaan.

Jika Anda mengaktifkan antrian pekerjaan tetapi tidak melebihi kuota untuk permintaan bersamaan, semua permintaan akan diproses secara bersamaan.

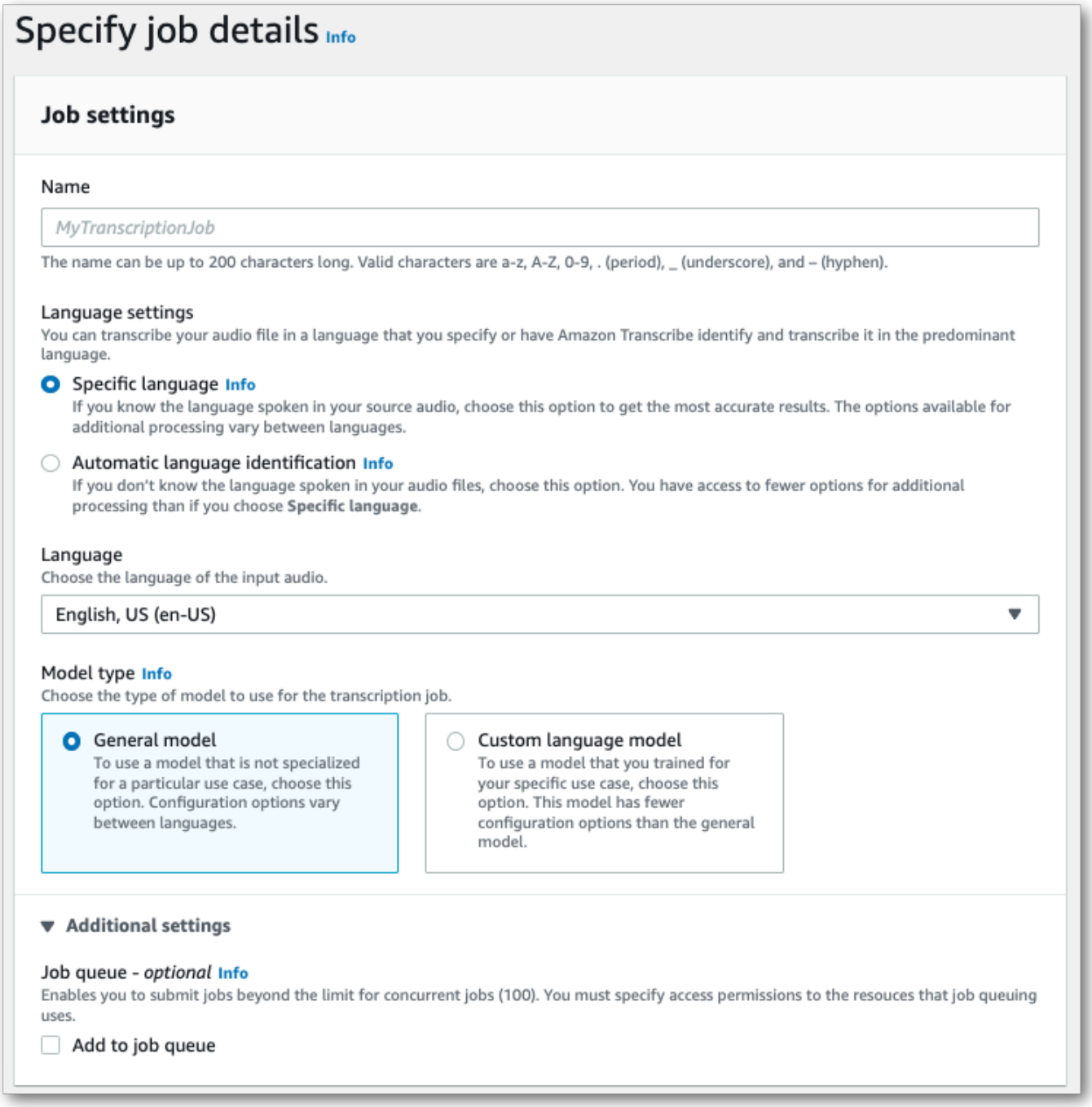
Mengaktifkan antrian pekerjaan

Anda dapat mengaktifkan antrian pekerjaan menggunakan AWS Management Console, AWS CLI, atau AWS SDKs; lihat contoh berikut; lihat contoh berikut ini:

AWS Management Console

1. Masuk ke AWS Management Console.

2. Di panel navigasi, pilih Pekerjaan transkripsi, lalu pilih Buat pekerjaan (kanan atas). Ini membuka halaman Tentukan detail pekerjaan.
3. Di kotak Job Settings, ada panel Pengaturan tambahan. Jika Anda memperluas panel ini, Anda dapat memilih kotak Tambahkan ke antrean pekerjaan untuk mengaktifkan antrian pekerjaan.



4. Isi kolom lain yang ingin Anda sertakan di halaman Tentukan detail pekerjaan, lalu pilih Berikutnya. Ini membawa Anda ke halaman Konfigurasi pekerjaan - opsional.

5. Pilih Buat pekerjaan untuk menjalankan pekerjaan transkripsi Anda.

AWS CLI

Contoh ini menggunakan start-transcription-jobperintah dan `job-execution-settings` parameter dengan `AllowDeferredExecution` sub-parameter. Perhatikan bahwa ketika Anda memasukkan `AllowDeferredExecution` dalam permintaan Anda, Anda juga harus menyertakan`DataAccessRoleArn`.

Untuk informasi selengkapnya, silakan lihat `StartTranscriptionJob` dan `JobExecutionSettings`.

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

Berikut contoh lain menggunakan start-transcription-jobperintah, dan badan permintaan yang memungkinkan antrian.

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

File my-first-queueing-request.json berisi badan permintaan berikut.

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

```
  "JobExecutionSettings": {
        "AllowDeferredExecution": true,
        "DataAccessRoleArn": "arn:aws:iam::111122223333:role/ExampleRole"
  }
}
```

AWS SDK untuk Python (Boto3)

Contoh ini menggunakan AWS SDK untuk Python (Boto3) untuk mengaktifkan antrian pekerjaan menggunakan `AllowDeferredExecution` argumen untuk metode start_transcription_job. Perhatikan bahwa ketika Anda memasukkan `AllowDeferredExecution` dalam permintaan Anda, Anda juga harus menyertakan`DataAccessRoleArn`. Untuk informasi selengkapnya, silakan lihat `StartTranscriptionJob` dan `JobExecutionSettings`.

Untuk contoh tambahan menggunakan AWS SDKs, termasuk contoh khusus fitur, skenario, dan lintas layanan, lihat bagian ini. Contoh kode untuk Amazon Transcribe menggunakan AWS SDKs

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

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

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

Anda dapat melihat kemajuan pekerjaan antrian melalui AWS Management Console atau dengan mengirimkan permintaan. `GetTranscriptionJob` Ketika pekerjaan antri, itu adalah. `Status` `QUEUED` Status berubah menjadi `IN_PROGRESS` setelah pekerjaan Anda mulai diproses, kemudian berubah menjadi `COMPLETED` atau `FAILED` saat pemrosesan selesai.

Penandaan pada sumber daya

Tag adalah label metadata kustom yang dapat Anda tambahkan ke sumber daya agar lebih mudah mengidentifikasi, mengatur, dan menemukan dalam pencarian. Tag terdiri dari dua bagian individual: Kunci tag dan nilai tag. Ini disebut sebagai pasangan key:value.

Kunci tag biasanya mewakili kategori yang lebih besar, sedangkan nilai tag mewakili subset dari kategori tersebut. Misalnya Anda dapat memiliki tag key=color dan tag value=blue, yang akan menghasilkan pasangan key:value. `Color:Blue` Perhatikan bahwa Anda dapat mengatur nilai tag ke string kosong, tetapi Anda tidak dapat mengatur nilai tag ke null. Mengabaikan nilai tag sama dengan menggunakan sebuah string kosong.

> **Tip**
>
> AWS Manajemen Penagihan dan Biaya dapat menggunakan tag untuk memisahkan tagihan Anda ke dalam kategori dinamis. Misalnya, jika Anda menambahkan tag untuk mewakili departemen yang berbeda dalam perusahaan Anda, seperti `Department:Sales` atau`Department:Legal`, AWS dapat memberi Anda distribusi biaya per departemen.

Di Amazon Transcribe, Anda dapat menandai sumber daya berikut:

- Lowongan kerja transkripsi
- Pekerjaan transkripsi medis
- Pekerjaan transkripsi panggilan analitik panggilan panggilan
- Kosakata khusus
- Kosakata medis khusus
- Filter kosakata khusus
- Kategori analitik panggilan
- Model bahasa khusus

Tombol tag dapat memiliki panjang hingga 128 karakter dan nilai tag dapat mencapai 256 karakter; keduanya peka huruf besar/kecil. Amazon Transcribe mendukung hingga 50 tag per sumber daya. Untuk sumber daya tertentu, setiap kunci tag harus unik dengan hanya satu nilai. Perhatikan bahwa tag Anda tidak dapat dimulai `aws:` karena AWS menyimpan awalan ini untuk tag yang dihasilkan

sistem. Anda tidak dapat menambahkan, memodifikasi, atau menghapus `aws:*` tag, dan mereka tidak dihitung terhadap tags-per-resource batas Anda.

> **Operasi API khusus untuk penandaan sumber daya**
>
> `ListTagsForResource`, `TagResource`, `UntagResource`
> Untuk menggunakan penandaan APIs, Anda harus menyertakan Nama Sumber Daya Amazon (ARN) dengan permintaan Anda. ARNs memiliki formatnya`arn:partition:service:region:account-id:resource-type/resource-id`. Misalnya, ARN yang terkait dengan pekerjaan transkripsi mungkin terlihat seperti:. `arn:aws:transcribe:us-west-2:111122223333:transcription-job/my-transcription-job-name`

Untuk mempelajari lebih lanjut tentang penandaan, termasuk praktik terbaik, lihat Menandai sumber daya AWS.

Kontrol akses berbasis tag

Anda dapat menggunakan tag untuk mengontrol akses di dalam Anda Akun AWS. Untuk kontrol akses berbasis tag, Anda memberikan informasi tag dalam elemen kondisi IAM kebijakan. Anda kemudian dapat menggunakan tag dan kunci kondisi tag terkait untuk mengontrol akses ke:

- Sumber Daya: Kontrol akses ke Amazon Transcribe sumber daya Anda berdasarkan tag yang telah Anda tetapkan ke sumber daya tersebut.
  - Gunakan tombol `aws:ResourceTag/key-name` kondisi untuk menentukan pasangan kunci tag: nilai mana yang harus dilampirkan ke sumber daya.
- Permintaan: Kontrol tag mana yang dapat diteruskan dalam permintaan.
  - Gunakan tombol `aws:RequestTag/key-name` kondisi untuk menentukan tag mana yang dapat ditambahkan, dimodifikasi, atau dihapus dari IAM pengguna atau peran.
- Proses otorisasi: Kontrol akses berbasis tag untuk bagian mana pun dari proses otorisasi Anda.
  - Gunakan tombol `aws:TagKeys/` kondisi untuk mengontrol apakah kunci tag tertentu dapat digunakan pada sumber daya, permintaan, atau oleh prinsipal. Dalam hal ini, nilai kunci tidak masalah.

Untuk contoh kebijakan kontrol akses berbasis tag, lihatMelihat pekerjaan transkripsi berdasarkan tag.

Untuk informasi lebih rinci tentang kontrol akses berbasis tag, lihat Mengontrol akses ke AWS sumber daya menggunakan tag.

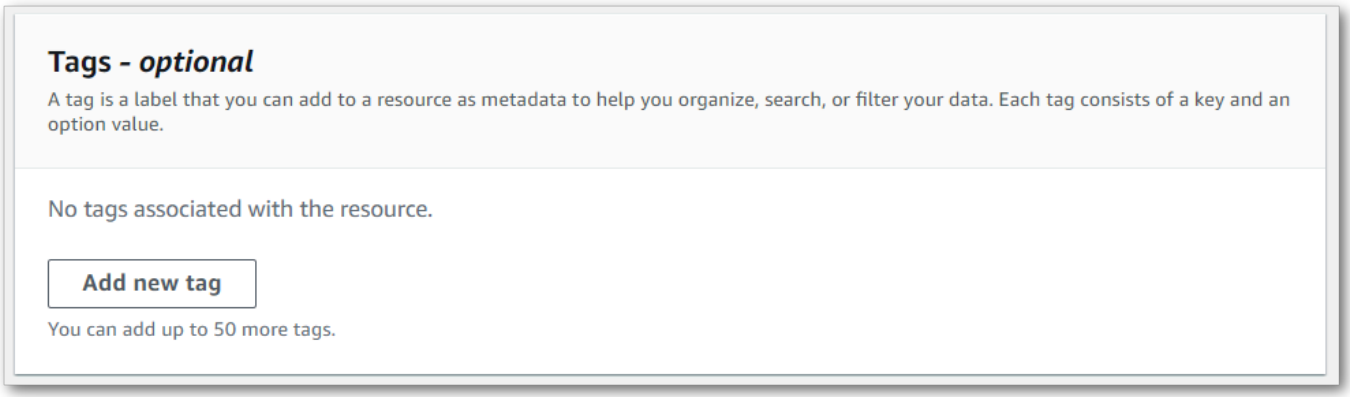
Menambahkan tag ke Amazon Transcribe sumber daya Anda

Anda dapat menambahkan tag sebelum atau setelah Anda menjalankan Amazon Transcribe pekerjaan Anda. Menggunakan Create* dan Start* yang ada APIs, Anda dapat menambahkan tag tambahan dengan permintaan transkripsi Anda.

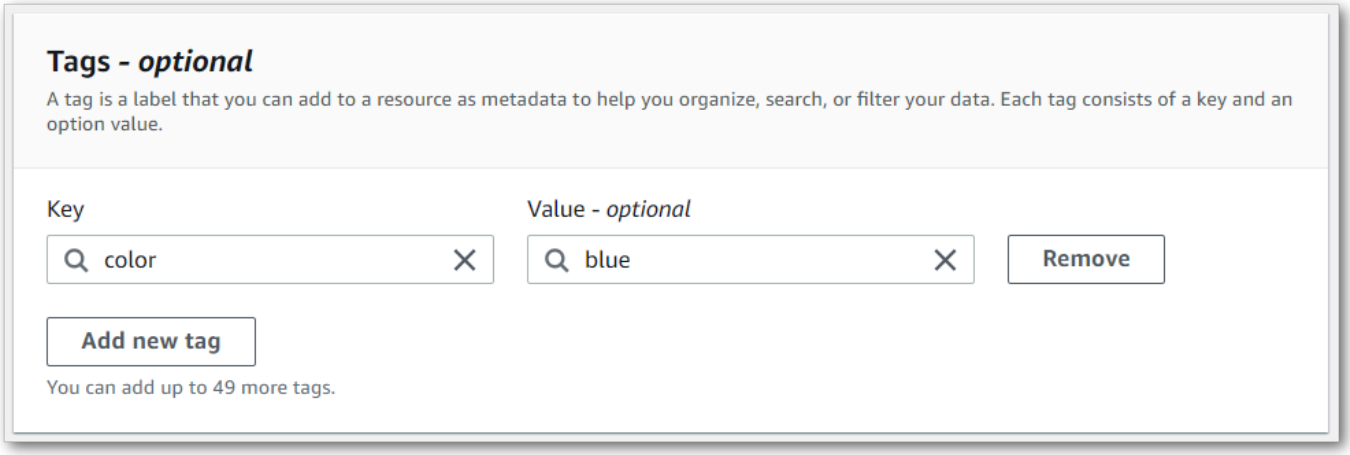
Anda dapat menambahkan, memodifikasi, atau menghapus tag menggunakan AWS Management ConsoleAWS CLI,, atau AWS SDKs; lihat contoh berikut:

AWS Management Console

1. Masuk ke AWS Management Console.
2. Di panel navigasi, pilih Pekerjaan transkripsi, lalu pilih Buat pekerjaan (kanan atas). Ini membuka halaman Tentukan detail pekerjaan.
3. Gulir ke bagian bawah halaman Tentukan detail pekerjaan untuk menemukan Tag - kotak opsional dan pilih Tambahkan tag baru.



4. Masukkan informasi untuk bidang Kunci dan, secara opsional, bidang Nilai.



5. Isi kolom lain yang ingin Anda sertakan di halaman Tentukan detail pekerjaan, lalu pilih Berikutnya. Ini membawa Anda ke halaman Konfigurasi pekerjaan - opsional.

   Pilih Buat pekerjaan untuk menjalankan pekerjaan transkripsi Anda.

6. Anda dapat melihat tag yang terkait dengan pekerjaan transkripsi dengan menavigasi ke halaman pekerjaan Transkripsi, memilih pekerjaan transkripsi, dan menggulir ke bagian bawah halaman informasi pekerjaan itu. Jika Anda ingin mengedit tag Anda, Anda dapat melakukannya dengan memilih Kelola tag.

AWS CLI

Contoh ini menggunakan start-transcription-jobperintah dan `Tags` parameter. Untuk informasi selengkapnya, silakan lihat `StartTranscriptionJob` dan `Tag`.

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

```
--tags Key=color,Value=blue Key=shape,Value=square
```

Berikut contoh lain menggunakan start-transcription-jobperintah, dan badan permintaan yang menambahkan tag ke pekerjaan itu.

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

File my-first-tagging-job.json berisi badan permintaan berikut.

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

AWS SDK untuk Python (Boto3)

Contoh berikut menggunakan AWS SDK untuk Python (Boto3) untuk menambahkan tag dengan menggunakan `Tags` argumen untuk metode start_transcription_job. Untuk informasi selengkapnya, silakan lihat `StartTranscriptionJob` dan `Tag`.

Untuk contoh tambahan menggunakan AWS SDKs, termasuk contoh khusus fitur, skenario, dan lintas layanan, lihat bagian ini. Contoh kode untuk Amazon Transcribe menggunakan AWS SDKs

```
from __future__ import print_function
```

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

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

Mempartisi speaker (diarisasi)

Dengan diarisasi speaker, Anda dapat membedakan antara speaker yang berbeda dalam output transkripsi Anda. Amazon Transcribe dapat membedakan antara maksimal 30 speaker unik dan memberi label teks dari setiap pembicara unik dengan nilai unik (`spk_0`melalui`spk_9`).

Selain bagian transkrip standar (`transcripts`dan`items`), permintaan dengan partisi speaker diaktifkan termasuk bagian. `speaker_labels` Bagian ini dikelompokkan berdasarkan pembicara dan berisi informasi tentang setiap ucapan, termasuk label speaker dan stempel waktu.

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

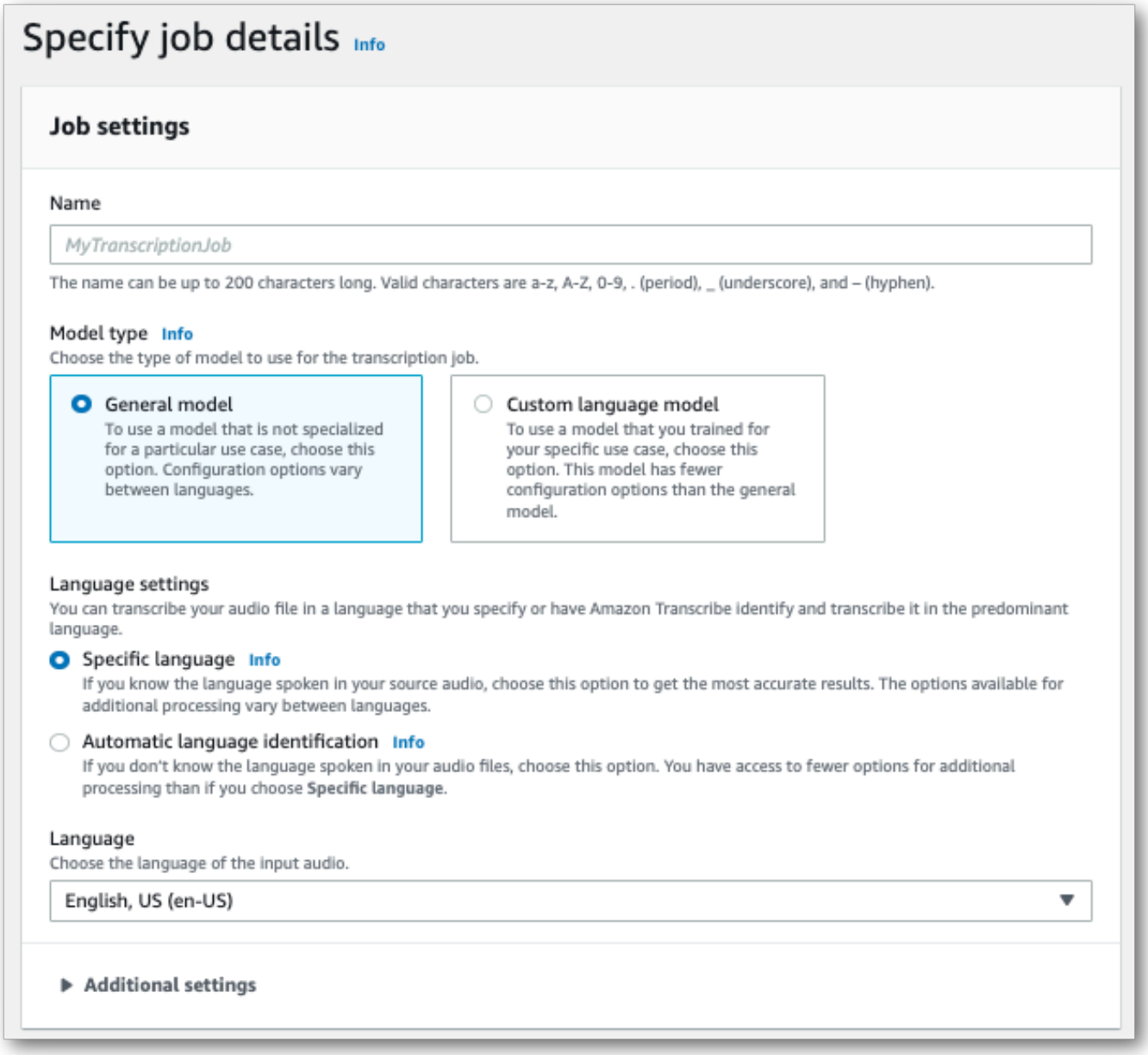
Untuk melihat contoh transkrip lengkap dengan partisi pembicara (untuk dua pembicara), lihat. Contoh keluaran diarisasi (batch)

Mempartisi speaker dalam transkripsi batch

Untuk mempartisi speaker dalam transkripsi batch, lihat contoh berikut:

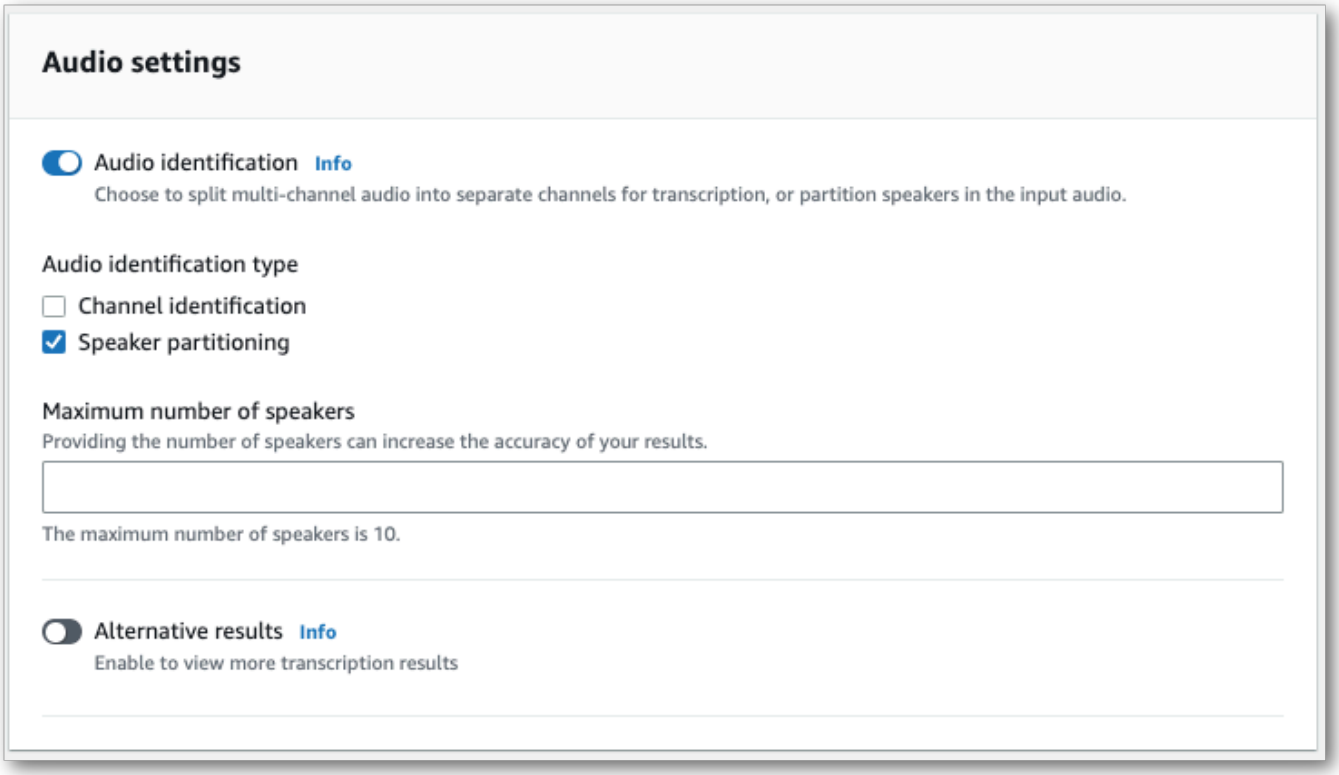
AWS Management Console

1. Masuk ke AWS Management Console.
2. Di panel navigasi, pilih Pekerjaan transkripsi, lalu pilih Buat pekerjaan (kanan atas). Ini membuka halaman Tentukan detail pekerjaan.



3. Isi kolom yang ingin Anda sertakan di halaman Tentukan detail pekerjaan, lalu pilih Berikutnya. Ini membawa Anda ke halaman Konfigurasi pekerjaan - opsional.

   Untuk mengaktifkan partisi speaker, di Pengaturan audio, pilih Identifikasi audio. Kemudian pilih Partisi speaker dan tentukan jumlah speaker.



4. Pilih Buat pekerjaan untuk menjalankan pekerjaan transkripsi Anda.

AWS CLI

Contoh ini menggunakan start-transcription-job. Untuk informasi selengkapnya, lihat StartTranscriptionJob.

```
aws transcribe start-transcription-job \
--region us-west-2 \
--transcription-job-name my-first-transcription-job \
--media MediaFileUri=s3://amzn-s3-demo-bucket/my-input-files/my-media-file.flac \
--output-bucket-name amzn-s3-demo-bucket \
--output-key my-output-files/ \
--language-code en-US \
--settings ShowSpeakerLabels=true,MaxSpeakerLabels=3
```

Berikut contoh lain menggunakan start-transcription-jobperintah, dan badan permintaan yang memungkinkan pembicara partisi dengan pekerjaan itu.

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

File my-first-transcription-job.json berisi badan permintaan berikut.

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

AWS SDK untuk Python (Boto3)

Contoh ini menggunakan AWS SDK untuk Python (Boto3) untuk mengidentifikasi saluran menggunakan metode start_transcription_job. Untuk informasi selengkapnya, lihat StartTranscriptionJob.

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

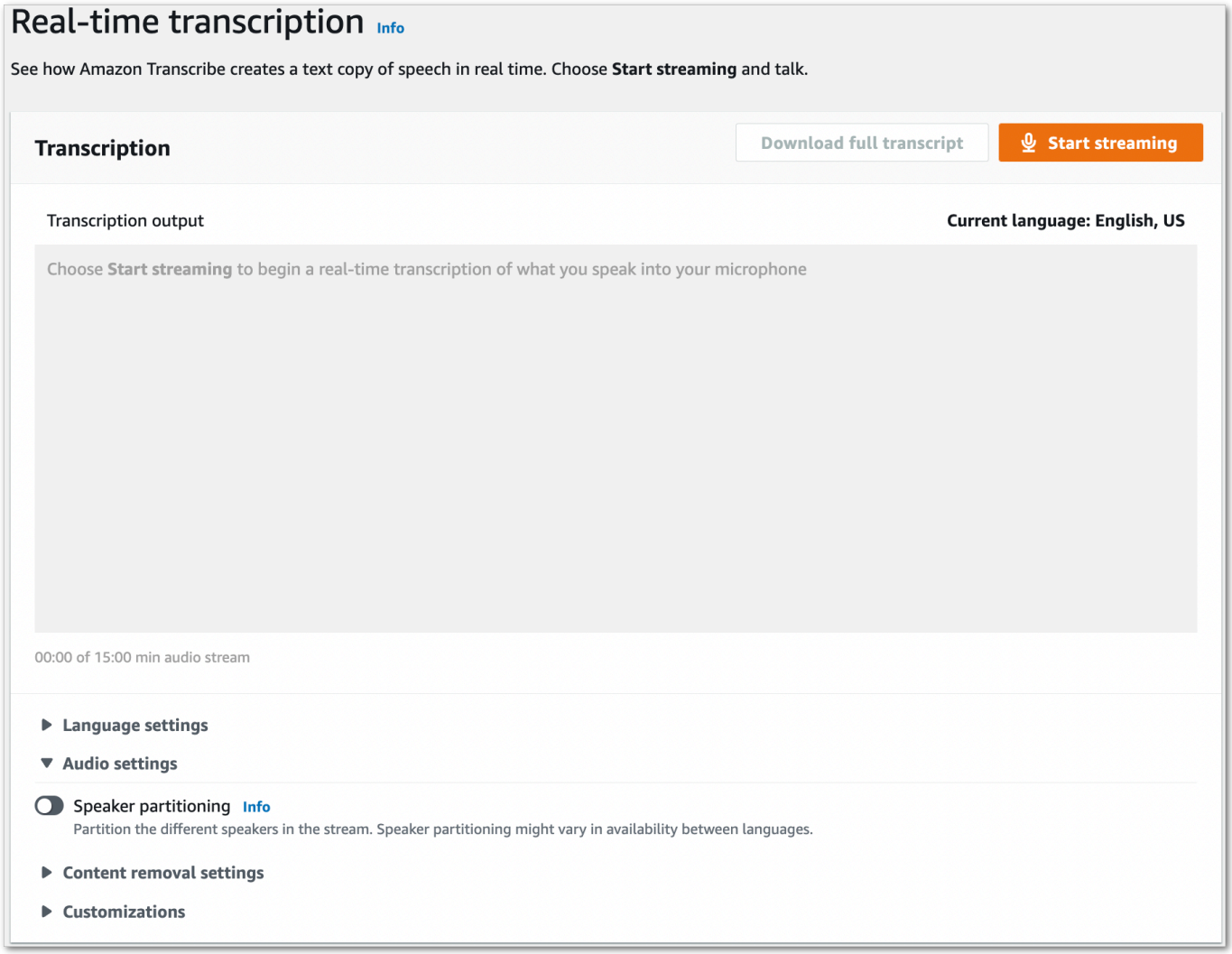
```
    Settings = {
        'ShowSpeakerLabels': True,
        'MaxSpeakerLabels': 3
    }
)

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

Mempartisi speaker dalam transkripsi streaming

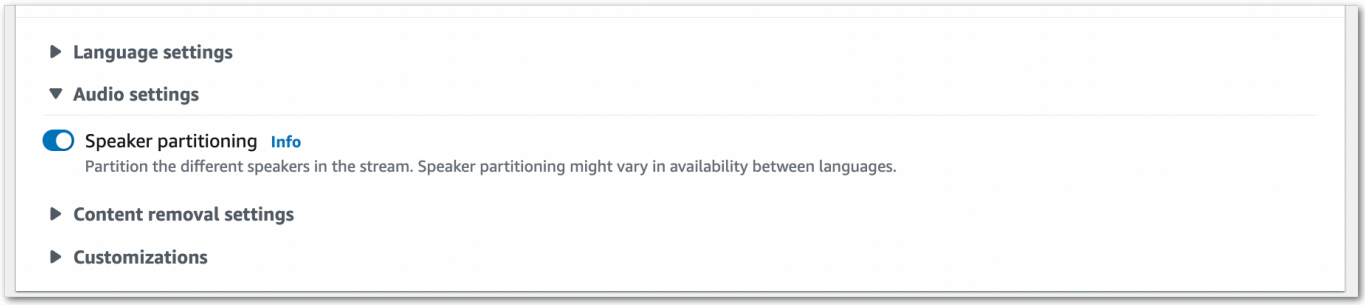
Untuk mempartisi speaker dalam transkripsi streaming, lihat contoh berikut:

Transkripsi streaming

1. Masuk ke AWS Management Console.
2. Di panel navigasi, pilih Transkripsi waktu nyata. Gulir ke bawah ke pengaturan Audio dan perluas bidang ini jika diminimalkan.



3. Beralih pada partisi Speaker.



4. Anda sekarang siap untuk mentranskripsikan aliran Anda. Pilih Mulai streaming dan mulai berbicara. Untuk mengakhiri dikte Anda, pilih Hentikan streaming.

Aliran HTTP/2

Contoh ini membuat permintaan HTTP/2 yang mempartisi speaker dalam output transkripsi Anda. Untuk informasi lebih lanjut tentang menggunakan streaming HTTP/2 dengan Amazon Transcribe, lihat. Menyiapkan aliran HTTP/2 Untuk detail selengkapnya tentang parameter dan header khusus untuk Amazon Transcribe, lihat 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
```

Definisi parameter dapat ditemukan di Referensi API; parameter yang umum untuk semua operasi AWS API tercantum di bagian Parameter Umum.

WebSocket aliran

Contoh ini membuat URL presigned yang memisahkan speaker dalam output transkripsi Anda. Jeda baris telah ditambahkan untuk keterbacaan. Untuk informasi selengkapnya tentang penggunaan WebSocket stream dengan Amazon Transcribe, lihatMenyiapkan WebSocket aliran. Untuk detail lebih lanjut tentang parameter, lihat `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
```

Definisi parameter dapat ditemukan di Referensi API; parameter yang umum untuk semua operasi AWS API tercantum di bagian Parameter Umum.

Contoh keluaran diarisasi (batch)

Berikut adalah contoh keluaran untuk transkripsi batch dengan diarisasi diaktifkan.

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

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

```
                        "end_time": "9.24"
                    }
                ]
            }
        ]
    },
    "items": [
        {
            "id": 0,
            "start_time": "4.87",
            "speaker_label": "spk_0",
            "end_time": "5.02",
            "alternatives": [
                {
                    "confidence": "1.0",
                    "content": "I've"
                }
            ],
            "type": "pronunciation"
        },
        {
            "id": 1,
            "start_time": "5.02",
            "speaker_label": "spk_0",
            "end_time": "5.17",
            "alternatives": [
                {
                    "confidence": "1.0",
                    "content": "been"
                }
            ],
            "type": "pronunciation"
        },
        {
            "id": 2,
            "start_time": "5.17",
            "speaker_label": "spk_0",
            "end_time": "5.29",
            "alternatives": [
                {
                    "confidence": "1.0",
                    "content": "on"
                }
            ],
```

```
            "type": "pronunciation"
        },
        {
            "id": 3,
            "start_time": "5.29",
            "speaker_label": "spk_0",
            "end_time": "5.64",
            "alternatives": [
                {
                    "confidence": "1.0",
                    "content": "hold"
                }
            ],
            "type": "pronunciation"
        },
        {
            "id": 4,
            "start_time": "5.64",
            "speaker_label": "spk_0",
            "end_time": "5.84",
            "alternatives": [
                {
                    "confidence": "1.0",
                    "content": "for"
                }
            ],
            "type": "pronunciation"
        },
        {
            "id": 5,
            "start_time": "6.11",
            "speaker_label": "spk_0",
            "end_time": "6.26",
            "alternatives": [
                {
                    "confidence": "1.0",
                    "content": "an"
                }
            ],
            "type": "pronunciation"
        },
        {
            "id": 6,
            "start_time": "6.26",
```

```
                "speaker_label": "spk_0",
                "end_time": "6.88",
                "alternatives": [
                    {
                        "confidence": "1.0",
                        "content": "hour"
                    }
                ],
                "type": "pronunciation"
            },
            {
                "id": 7,
                "speaker_label": "spk_0",
                "alternatives": [
                    {
                        "confidence": "0.0",
                        "content": "."
                    }
                ],
                "type": "punctuation"
            },
            {
                "id": 8,
                "start_time": "8.49",
                "speaker_label": "spk_1",
                "end_time": "8.88",
                "alternatives": [
                    {
                        "confidence": "1.0",
                        "content": "Sorry"
                    }
                ],
                "type": "pronunciation"
            },
            {
                "id": 9,
                "start_time": "8.88",
                "speaker_label": "spk_1",
                "end_time": "9.05",
                "alternatives": [
                    {
                        "confidence": "0.902",
                        "content": "about"
                    }
```

```
                ],
                "type": "pronunciation"
            },
            {
                "id": 10,
                "start_time": "9.05",
                "speaker_label": "spk_1",
                "end_time": "9.24",
                "alternatives": [
                    {
                        "confidence": "1.0",
                        "content": "that"
                    }
                ],
                "type": "pronunciation"
            },
            {
                "id": 11,
                "speaker_label": "spk_1",
                "alternatives": [
                    {
                        "confidence": "0.0",
                        "content": "."
                    }
                ],
                "type": "punctuation"
            }
        ],
        "audio_segments": [
            {
                "id": 0,
                "transcript": "I've been on hold for an hour.",
                "start_time": "4.87",
                "end_time": "6.88",
                "speaker_label": "spk_0",
                "items": [
                    0,
                    1,
                    2,
                    3,
                    4,
                    5,
                    6,
                    7
```

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

Mentranskripsikan audio multi-saluran

Jika audio Anda memiliki dua saluran, Anda dapat menggunakan identifikasi saluran untuk mentranskripsikan ucapan dari setiap saluran secara terpisah. Amazon Transcribe Saat ini tidak mendukung audio dengan lebih dari dua saluran.

Dalam transkrip Anda, saluran diberi label `ch_0` dan`ch_1`.

Selain bagian transkrip standar (`transcriptsdanitems`), permintaan dengan identifikasi saluran diaktifkan termasuk `channel_labels` bagian. Bagian ini berisi setiap ucapan atau tanda baca, dikelompokkan berdasarkan saluran, dan label saluran terkait, cap waktu, dan skor kepercayaan.

```
"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
    },
```

Perhatikan bahwa jika seseorang di satu saluran berbicara pada waktu yang sama dengan orang di saluran terpisah, stempel waktu untuk setiap saluran tumpang tindih saat individu berbicara satu sama lain.

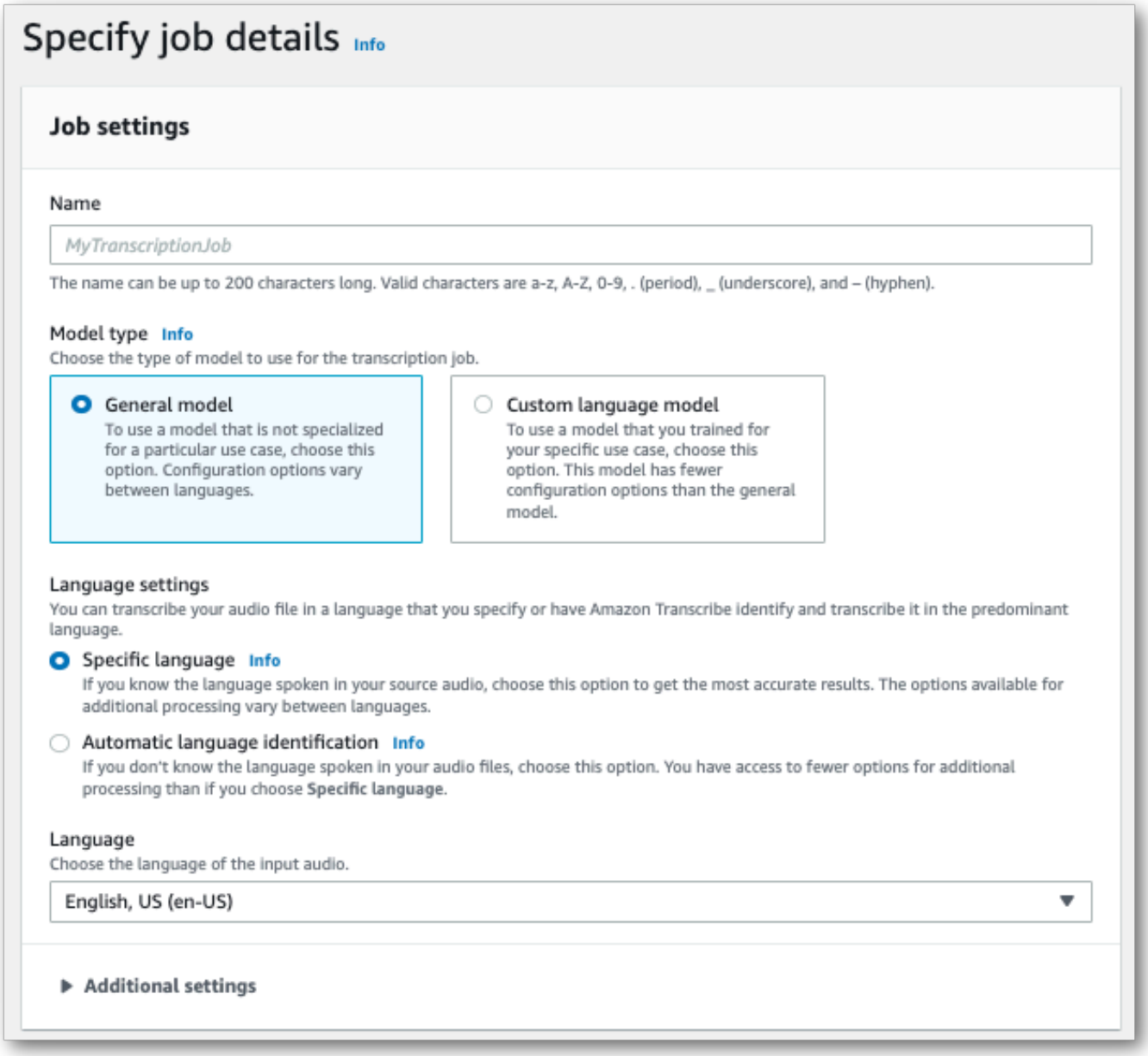
Untuk melihat contoh transkrip lengkap dengan identifikasi saluran, lihatContoh keluaran identifikasi saluran (batch).

Menggunakan identifikasi saluran dalam transkripsi batch

Untuk mengidentifikasi saluran dalam transkripsi batch, Anda dapat menggunakan AWS Management Console, AWS CLI, atau AWS SDKs; lihat contoh berikut:

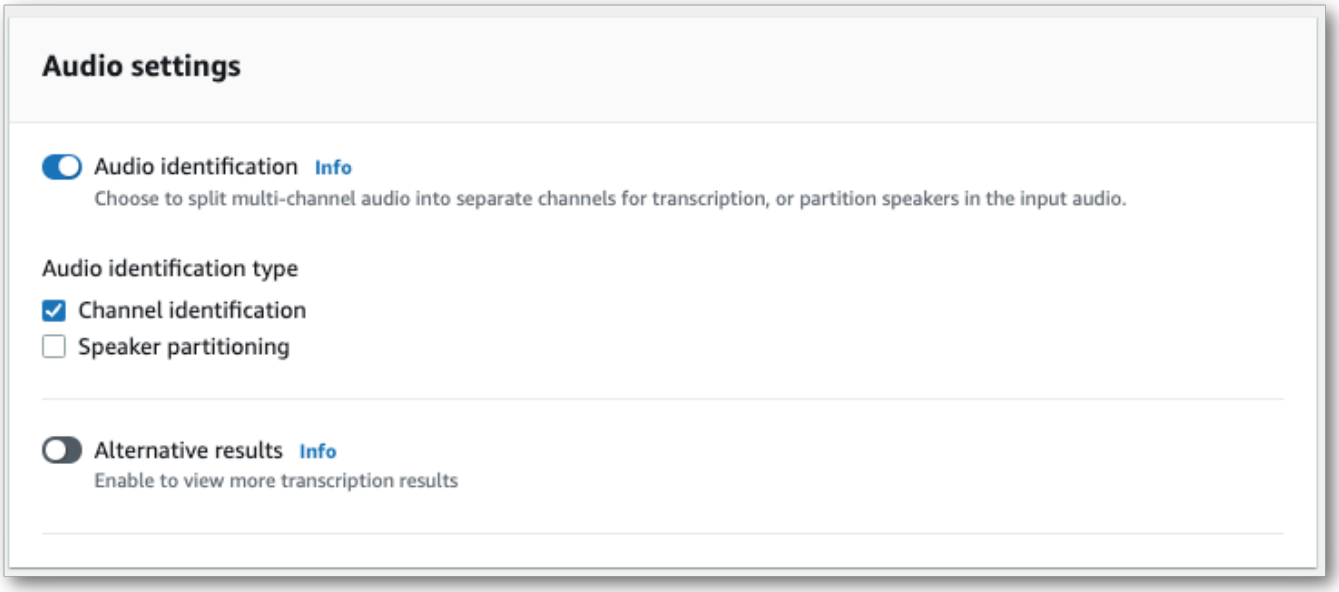
AWS Management Console

1. Masuk ke AWS Management Console.
2. Di panel navigasi, pilih Pekerjaan transkripsi, lalu pilih Buat pekerjaan (kanan atas). Ini membuka halaman Tentukan detail pekerjaan.



3. Isi kolom yang ingin Anda sertakan di halaman Tentukan detail pekerjaan, lalu pilih Berikutnya. Ini membawa Anda ke halaman Konfigurasi pekerjaan - opsional.

   Di panel Pengaturan audio, pilih Identifikasi saluran (di bawah judul 'Jenis identifikasi audio').



4. Pilih Buat pekerjaan untuk menjalankan pekerjaan transkripsi Anda.

AWS CLI

Contoh ini menggunakan start-transcription-job. Untuk informasi selengkapnya, lihat `StartTranscriptionJob`.

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

Berikut contoh lain menggunakan start-transcription-jobperintah, dan badan permintaan yang memungkinkan identifikasi saluran dengan pekerjaan itu.

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

File my-first-transcription-job.json berisi badan permintaan berikut.

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

AWS SDK untuk Python (Boto3)

Contoh ini menggunakan AWS SDK untuk Python (Boto3) untuk mengidentifikasi saluran menggunakan metode start_transcription_job. Untuk informasi selengkapnya, lihat StartTranscriptionJob.

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

while True:
    status = transcribe.get_transcription_job(TranscriptionJobName = job_name)
    if status['TranscriptionJob']['TranscriptionJobStatus'] in ['COMPLETED', 'FAILED']:
        break
```

```
    print("Not ready yet...")
    time.sleep(5)
print(status)
```

Menggunakan identifikasi saluran dalam transkripsi streaming

Untuk mengidentifikasi saluran dalam transkripsi streaming, Anda dapat menggunakan HTTP/2 atau WebSockets; lihat contoh berikut:

Aliran HTTP/2

Contoh ini membuat permintaan HTTP/2 yang memisahkan saluran dalam keluaran transkripsi Anda. Untuk informasi lebih lanjut tentang menggunakan streaming HTTP/2 dengan Amazon Transcribe, lihat. Menyiapkan aliran HTTP/2 Untuk detail selengkapnya tentang parameter dan header khusus untuk Amazon Transcribe, lihat 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
```

Definisi parameter dapat ditemukan di Referensi API; parameter yang umum untuk semua operasi AWS API tercantum di bagian Parameter Umum.

WebSocket aliran

Contoh ini membuat URL presigned yang memisahkan saluran dalam output transkripsi Anda. Jeda baris telah ditambahkan untuk keterbacaan. Untuk informasi selengkapnya tentang penggunaan WebSocket stream dengan Amazon Transcribe, lihatMenyiapkan WebSocket aliran. Untuk detail lebih lanjut tentang parameter, lihat `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
```

Definisi parameter dapat ditemukan di Referensi API; parameter yang umum untuk semua operasi AWS API tercantum di bagian Parameter Umum.

Contoh keluaran identifikasi saluran (batch)

Berikut adalah contoh keluaran untuk transkripsi batch dengan identifikasi saluran diaktifkan.

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

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

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

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

```
                    {
                        "channel_label": "ch_1",
                        "alternatives": [
                            {
                                "confidence": "0.0",
                                "content": "."
                            }
                        ],
                        "type": "punctuation"
                    }
                ]
            }
        ],
        "number_of_channels": 2
    },
    "items": [
        {
            "id": 0,
            "channel_label": "ch_0",
            "start_time": "4.86",
            "end_time": "5.01",
            "alternatives": [
                {
                    "confidence": "1.0",
                    "content": "I've"
                }
            ],
            "type": "pronunciation"
        },
        {
            "id": 1,
            "channel_label": "ch_0",
            "start_time": "5.01",
            "end_time": "5.16",
            "alternatives": [
                {
                    "confidence": "1.0",
                    "content": "been"
                }
            ],
            "type": "pronunciation"
        },
        {
            "id": 2,
```

```
                "channel_label": "ch_0",
                "start_time": "5.16",
                "end_time": "5.28",
                "alternatives": [
                    {
                        "confidence": "1.0",
                        "content": "on"
                    }
                ],
                "type": "pronunciation"
            },
            {
                "id": 3,
                "channel_label": "ch_0",
                "start_time": "5.28",
                "end_time": "5.62",
                "alternatives": [
                    {
                        "confidence": "1.0",
                        "content": "hold"
                    }
                ],
                "type": "pronunciation"
            },
            {
                "id": 4,
                "channel_label": "ch_0",
                "start_time": "5.62",
                "end_time": "5.83",
                "alternatives": [
                    {
                        "confidence": "1.0",
                        "content": "for"
                    }
                ],
                "type": "pronunciation"
            },
            {
                "id": 5,
                "channel_label": "ch_0",
                "start_time": "6.1",
                "end_time": "6.25",
                "alternatives": [
                    {
```

```
                "confidence": "1.0",
                "content": "an"
            }
        ],
        "type": "pronunciation"
    },
    {
        "id": 6,
        "channel_label": "ch_0",
        "start_time": "6.25",
        "end_time": "6.87",
        "alternatives": [
            {
                "confidence": "1.0",
                "content": "hour"
            }
        ],
        "type": "pronunciation"
    },
    {
        "id": 7,
        "channel_label": "ch_0",
        "alternatives": [
            {
                "confidence": "0.0",
                "content": "."
            }
        ],
        "type": "punctuation"
    },
    {
        "id": 8,
        "channel_label": "ch_1",
        "start_time": "8.5",
        "end_time": "8.89",
        "alternatives": [
            {
                "confidence": "1.0",
                "content": "Sorry"
            }
        ],
        "type": "pronunciation"
    },
    {
```

```
                "id": 9,
                "channel_label": "ch_1",
                "start_time": "8.89",
                "end_time": "9.06",
                "alternatives": [
                    {
                        "confidence": "0.9176",
                        "content": "about"
                    }
                ],
                "type": "pronunciation"
            },
            {
                "id": 10,
                "channel_label": "ch_1",
                "start_time": "9.06",
                "end_time": "9.25",
                "alternatives": [
                    {
                        "confidence": "1.0",
                        "content": "that"
                    }
                ],
                "type": "pronunciation"
            },
            {
                "id": 11,
                "channel_label": "ch_1",
                "alternatives": [
                    {
                        "confidence": "0.0",
                        "content": "."
                    }
                ],
                "type": "punctuation"
            }
        ],
        "audio_segments": [
            {
                "id": 0,
                "transcript": "I've been on hold for an hour.",
                "start_time": "4.86",
                "end_time": "6.87",
                "channel_label": "ch_0",
```

```
                "items": [
                    0,
                    1,
                    2,
                    3,
                    4,
                    5,
                    6,
                    7
                ]
            },
            {
                "id": 1,
                "transcript": "Sorry about that.",
                "start_time": "8.5",
                "end_time": "9.25",
                "channel_label": "ch_1",
                "items": [
                    8,
                    9,
                    10,
                    11
                ]
            }
        ]
    },
    "status": "COMPLETED"
}
```

Mengidentifikasi bahasa dominan di media Anda

Amazon Transcribe dapat secara otomatis mengidentifikasi bahasa yang digunakan di media Anda tanpa Anda harus menentukan kode bahasa.

Identifikasi bahasa Batch dapat mengidentifikasi bahasa dominan yang digunakan dalam file media Anda atau, jika media Anda berisi beberapa bahasa, itu dapat mengidentifikasi semua bahasa yang digunakan. Untuk meningkatkan akurasi identifikasi bahasa, Anda dapat secara opsional memberikan daftar dua atau lebih bahasa yang menurut Anda mungkin ada di media Anda.

Identifikasi bahasa streaming dapat mengidentifikasi satu bahasa per saluran (maksimal dua saluran didukung) atau, jika aliran Anda berisi beberapa bahasa, itu dapat mengidentifikasi semua bahasa yang digunakan. Permintaan streaming harus memiliki minimal dua opsi bahasa tambahan yang disertakan dalam permintaan Anda. Menyediakan opsi bahasa memungkinkan identifikasi bahasa yang lebih cepat. Semakin cepat Amazon Transcribe dapat mengidentifikasi bahasa, semakin sedikit perubahan kehilangan data dalam beberapa detik pertama aliran Anda.

> **Important**
>
> Transkripsi batch dan streaming mendukung berbagai bahasa. Lihat kolom masukan data dalam tabel bahasa yang didukung untuk detailnya. Perhatikan bahwa Swedia dan Vietnam saat ini tidak didukung dengan identifikasi bahasa.

Untuk mempelajari tentang pemantauan dan peristiwa dengan identifikasi bahasa, lihatAcara identifikasi bahasa.

Identifikasi bahasa dengan pekerjaan transkripsi batch

Gunakan identifikasi bahasa batch untuk secara otomatis mengidentifikasi bahasa, atau bahasa, dalam file media Anda.

Jika media Anda hanya berisi satu bahasa, Anda dapat mengaktifkan identifikasi bahasa tunggal, yang mengidentifikasi bahasa dominan yang digunakan dalam file media Anda dan membuat transkrip Anda hanya menggunakan bahasa ini.

Jika media Anda berisi lebih dari satu bahasa, Anda dapat mengaktifkan identifikasi multi-bahasa, yang mengidentifikasi semua bahasa yang digunakan dalam file media Anda dan membuat transkrip

Anda menggunakan setiap bahasa yang diidentifikasi. Perhatikan bahwa transkrip multi-bahasa diproduksi. Anda dapat menggunakan layanan lain, seperti Amazon Translate, untuk menerjemahkan transkrip Anda.

Lihat tabel bahasa yang didukung untuk daftar lengkap bahasa yang didukung dan kode bahasa terkait.

Untuk hasil terbaik, pastikan file media Anda berisi setidaknya 30 detik pidato.

Untuk contoh penggunaan dengan SDK AWS Management Console, AWS CLI, dan AWS Python, lihat. Menggunakan identifikasi bahasa dengan transkripsi batch

Mengidentifikasi bahasa dalam audio multi-bahasa

Identifikasi multi-bahasa ditujukan untuk file media multi-bahasa, dan memberi Anda transkrip yang mencerminkan semua bahasa yang didukung yang digunakan di media Anda. Ini berarti bahwa jika penutur mengubah bahasa di tengah percakapan, atau jika setiap peserta berbicara bahasa yang berbeda, output transkripsi Anda mendeteksi dan mentranskripsikan setiap bahasa dengan benar. Misalnya, jika media Anda berisi pembicara bilingual yang bergantian antara bahasa Inggris AS (`en-US`) dan Hindi (`hi-IN`), identifikasi multi-bahasa dapat mengidentifikasi dan mentranskripsikan bahasa Inggris AS lisan sebagai dan bahasa Hindi yang diucapkan sebagai`en-US`. `hi-IN`

Ini berbeda dari identifikasi bahasa tunggal, di mana hanya satu bahasa dominan yang digunakan untuk membuat transkrip. Dalam hal ini, bahasa lisan apa pun yang bukan bahasa dominan ditranskripsikan secara tidak benar.

> **Note**
>
> Redaksi dan model bahasa kustom saat ini tidak didukung dengan identifikasi multi-bahasa.

> **Note**
>
> Bahasa-bahasa berikut saat ini didukung dengan identifikasi multi-bahasa: En-AB, en-AU, en-GB, en-IE, en-in, en-NZ, en-US, en-WL, en-ZA, es-ES, es-US, fr-CA, fr-FR, zh-CN, zh-TW, Pt-BR, Pt-pt, de-ch, de-de, af-za, ar-Ae, da-dK, He-il, Hi-il, Dalam, Id-ID, Fa-ir, IT-it, Ja-jp, Ko-kr, MS-saya, Nl-nl, Ru-ru, Ta-in, TE-in, TH-th, Tr-tr

Transkrip multi-bahasa memberikan ringkasan bahasa yang terdeteksi dan total waktu setiap bahasa digunakan di media Anda. Inilah contohnya:

```
"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
            }
        ]
}
```

Meningkatkan akurasi identifikasi bahasa

Dengan identifikasi bahasa, Anda memiliki opsi untuk memasukkan daftar bahasa yang menurut Anda mungkin ada di media Anda. Menyertakan opsi bahasa (`LanguageOptions`) membatasi Amazon Transcribe penggunaan hanya bahasa yang Anda tentukan saat mencocokkan audio Anda dengan bahasa yang benar, yang dapat mempercepat identifikasi bahasa dan meningkatkan akurasi yang terkait dengan penetapan dialek bahasa yang benar.

Jika Anda memilih untuk memasukkan kode bahasa, Anda harus menyertakan setidaknya dua. Tidak ada batasan jumlah kode bahasa yang dapat Anda sertakan, tetapi sebaiknya gunakan antara dua dan lima untuk efisiensi dan akurasi yang optimal.

> **Note**
>
> Jika Anda menyertakan kode bahasa dengan permintaan Anda dan tidak ada kode bahasa yang Anda berikan yang cocok dengan bahasa, atau bahasa, yang diidentifikasi dalam audio Anda, Amazon Transcribe memilih kecocokan bahasa terdekat dari kode bahasa yang Anda tentukan. Kemudian menghasilkan transkrip dalam bahasa itu. Misalnya, jika media Anda dalam bahasa Inggris AS (`en-US`) dan Anda memberikan Amazon Transcribe kode bahasa,, dan `zh-CN` `fr-FRde-DE`, Amazon Transcribe kemungkinan akan mencocokkan media Anda dengan bahasa Jerman (`de-DE`) dan menghasilkan transkripsi berbahasa Jerman. Ketidakcocokan kode bahasa dan bahasa lisan dapat mengakibatkan transkrip yang tidak akurat, jadi sebaiknya berhati-hati saat menyertakan kode bahasa.

Menggabungkan identifikasi bahasa dengan Amazon Transcribe fitur lain

Anda dapat menggunakan identifikasi bahasa batch dalam kombinasi dengan Amazon Transcribe fitur lainnya. Jika menggabungkan identifikasi bahasa dengan fitur lain, Anda terbatas pada bahasa yang didukung dengan fitur tersebut. Misalnya, jika menggunakan identifikasi bahasa dengan redaksi konten, Anda terbatas pada bahasa Inggris AS (`en-US`) atau Spanyol AS (`es-US`), karena ini hanya bahasa yang tersedia untuk redaksi. Lihat Bahasa yang didukung dan fitur khusus bahasa untuk informasi lebih lanjut.

> **Important**
>
> Jika Anda menggunakan identifikasi bahasa otomatis dengan redaksi konten diaktifkan dan audio Anda berisi bahasa selain bahasa Inggris AS (`en-US`) atau Spanyol AS (`es-US`), hanya konten Inggris AS atau Spanyol AS yang disunting dalam transkrip Anda. Bahasa lain tidak dapat disunting dan tidak ada peringatan atau kegagalan pekerjaan.

Model bahasa kustom, kosakata kustom, dan filter kosakata kustom

Jika Anda ingin menambahkan satu atau lebih model bahasa kustom, kosakata kustom, atau filter kosakata kustom ke permintaan identifikasi bahasa Anda, Anda harus menyertakan parameter. `LanguageIdSettings` Anda kemudian dapat menentukan kode bahasa dengan model bahasa kustom yang sesuai, kosakata kustom, dan filter kosakata kustom. Perhatikan bahwa identifikasi multi-bahasa tidak mendukung model bahasa khusus.

Disarankan agar Anda menyertakan `LanguageOptions` saat menggunakan `LanguageIdSettings`untuk memastikan bahwa dialek bahasa yang benar diidentifikasi. Misalnya, jika Anda menentukan kosakata `en-US` khusus, tetapi Amazon Transcribe menentukan bahwa bahasa yang digunakan di media Anda`en-AU`, kosakata khusus Anda tidak diterapkan pada transkripsi Anda. Jika Anda menyertakan `LanguageOptions` dan menentukan `en-US` sebagai satu-satunya dialek bahasa Inggris, kosakata khusus Anda diterapkan pada transkripsi Anda.

Untuk contoh `LanguageIdSettings`dalam permintaan, lihat Opsi 2 di panel AWS CLIdan AWS SDKstarik-turun di bagianMenggunakan identifikasi bahasa dengan transkripsi batch.

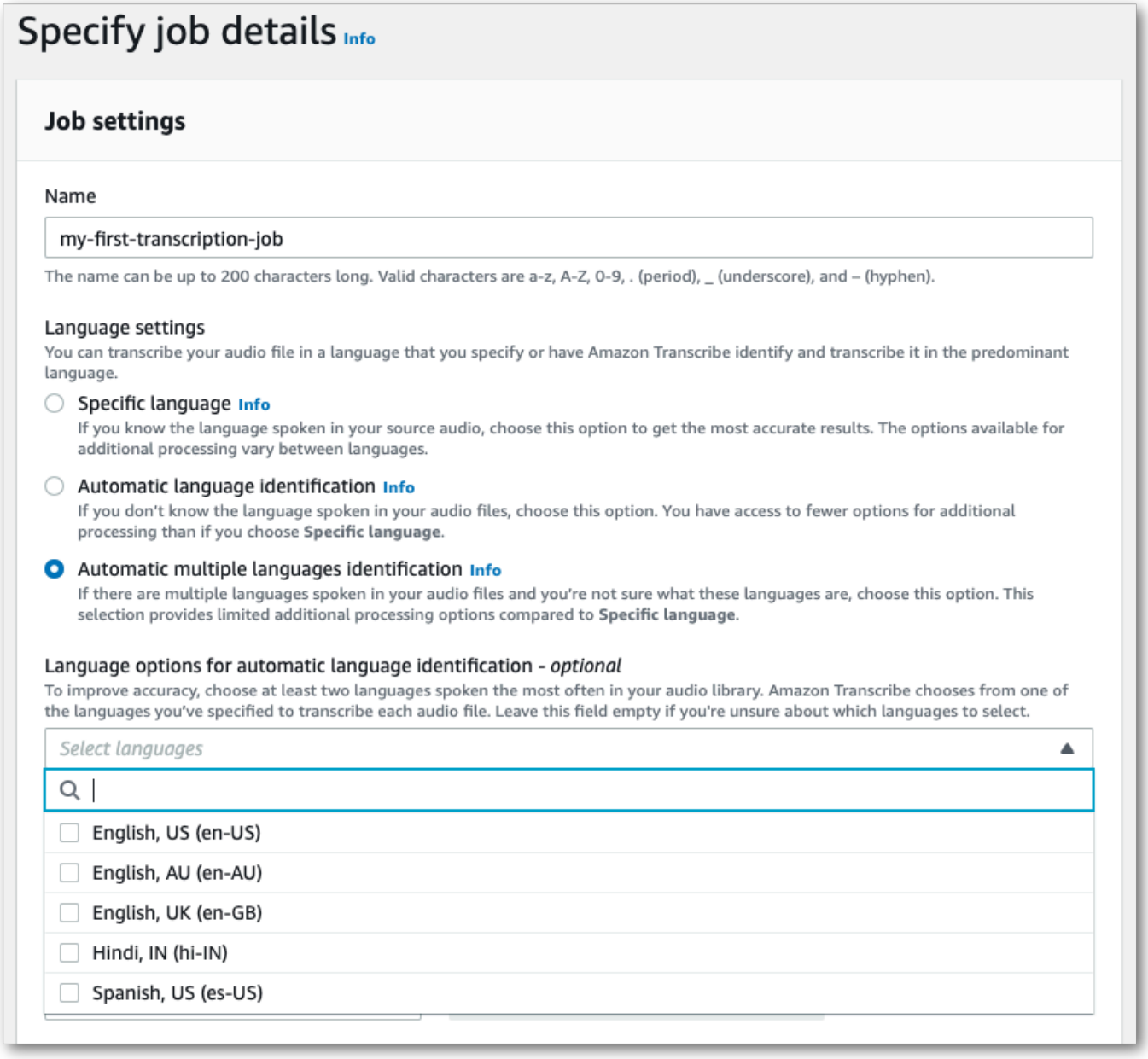
Menggunakan identifikasi bahasa dengan transkripsi batch

Anda dapat menggunakan identifikasi bahasa otomatis dalam pekerjaan transkripsi batch menggunakan AWS Management Console, AWS CLI, atau AWS SDKs; lihat contoh berikut:

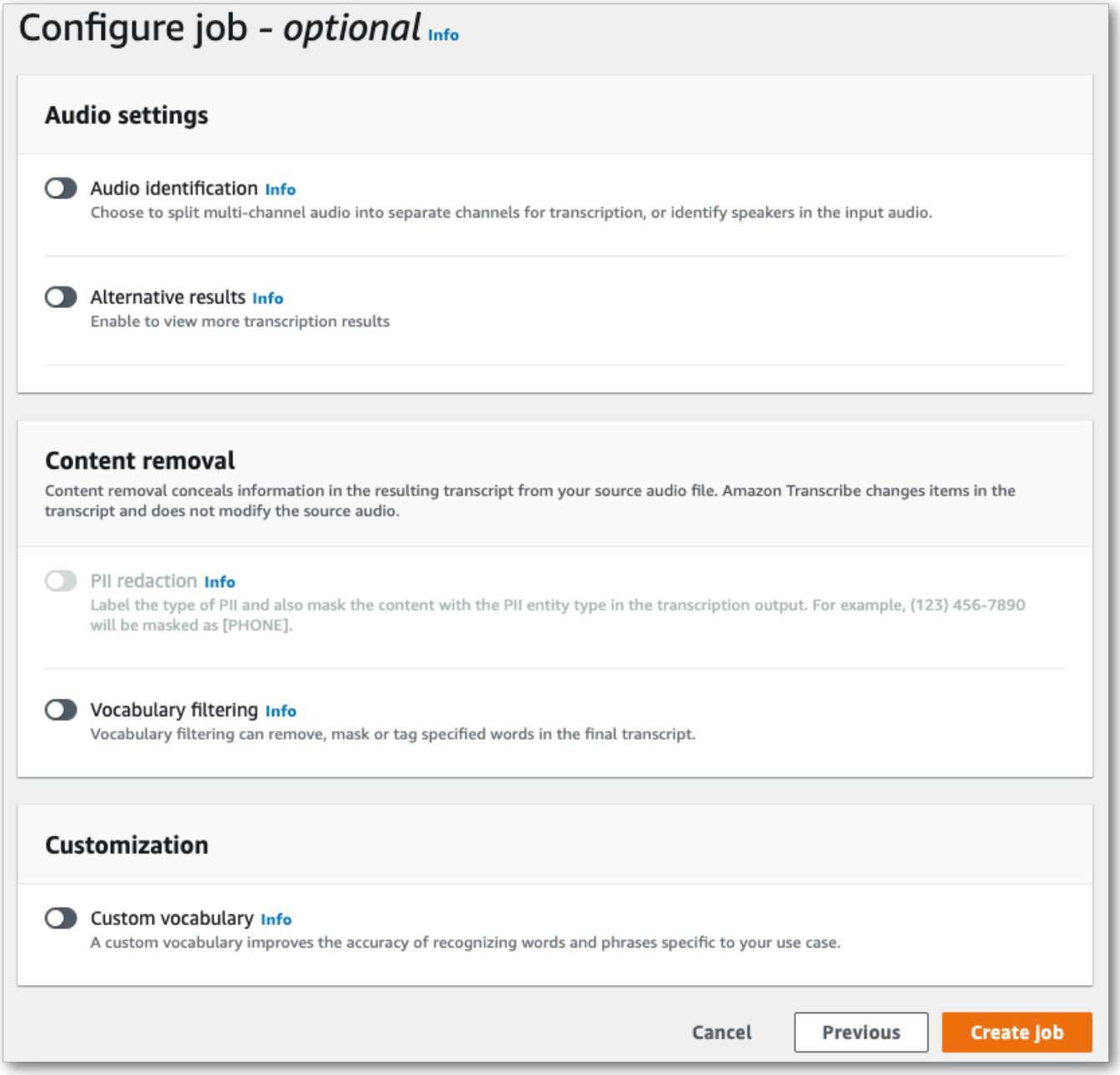
AWS Management Console

1. Masuk ke AWS Management Console.
2. Di panel navigasi, pilih Pekerjaan transkripsi, lalu pilih Buat pekerjaan (kanan atas). Ini membuka halaman Tentukan detail pekerjaan.
3. Di panel Pengaturan pekerjaan, temukan bagian Pengaturan bahasa dan pilih Identifikasi bahasa otomatis atau Identifikasi beberapa bahasa otomatis.

   Anda memiliki opsi untuk memilih beberapa opsi bahasa (dari kotak tarik-turun Pilih bahasa) jika Anda tahu bahasa mana yang ada di file audio Anda. Menyediakan pilihan bahasa dapat meningkatkan akurasi, tetapi tidak diperlukan.



4. Isi kolom lain yang ingin Anda sertakan di halaman Tentukan detail pekerjaan, lalu pilih Berikutnya. Ini membawa Anda ke halaman Konfigurasi pekerjaan - opsional.



5. Pilih Buat pekerjaan untuk menjalankan pekerjaan transkripsi Anda.

AWS CLI

Contoh ini menggunakan start-transcription-jobperintah dan `IdentifyLanguage` parameter. Untuk informasi selengkapnya, lihat `StartTranscriptionJob` dan `LanguageIdSettings`.

Opsi 1: Tanpa `language-id-settings` parameter. Gunakan opsi ini jika Anda tidak menyertakan model bahasa khusus, kosakata khusus, atau filter kosakata khusus dalam permintaan Anda. `language-options`bersifat opsional, tetapi direkomendasikan.

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

Opsi 2: Dengan `language-id-settings` parameter. Gunakan opsi ini jika Anda menyertakan model bahasa khusus, kosakata khusus, atau filter kosakata khusus dalam permintaan Anda.

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

Berikut contoh lain menggunakan start-transcription-jobperintah, dan badan permintaan yang mengidentifikasi bahasa.

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

File my-first-language-id-job.json berisi badan permintaan berikut.

Opsi 1: Tanpa `LanguageIdSettings` parameter. Gunakan opsi ini jika Anda tidak menyertakan model bahasa khusus, kosakata khusus, atau filter kosakata khusus dalam permintaan Anda. `LanguageOptions`bersifat opsional, tetapi direkomendasikan.

```
{
  "TranscriptionJobName": "my-first-transcription-job",
  "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
   },
  "OutputBucketName": "amzn-s3-demo-bucket",
  "OutputKey": "my-output-files/",
  "IdentifyLanguage": true,  (or "IdentifyMultipleLanguages": true),
  "LanguageOptions": [
        "en-US", "hi-IN"
  ]
}
```

Opsi 2: Dengan `LanguageIdSettings` parameter. Gunakan opsi ini jika Anda menyertakan model bahasa khusus, kosakata khusus, atau filter kosakata khusus dalam permintaan Anda.

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

AWS SDK untuk Python (Boto3)

Contoh ini menggunakan AWS SDK untuk Python (Boto3) untuk mengidentifikasi bahasa file Anda menggunakan `IdentifyLanguage` argumen untuk metode start_transcription_job. Untuk informasi selengkapnya, lihat `StartTranscriptionJob` dan `LanguageIdSettings`.

Untuk contoh tambahan menggunakan AWS SDKs, termasuk contoh khusus fitur, skenario, dan lintas layanan, lihat bagian ini. Contoh kode untuk Amazon Transcribe menggunakan AWS SDKs

Opsi 1: Tanpa `LanguageIdSettings` parameter. Gunakan opsi ini jika Anda tidak menyertakan model bahasa khusus, kosakata khusus, atau filter kosakata khusus dalam permintaan Anda. `LanguageOptions`bersifat opsional, tetapi direkomendasikan.

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

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

Opsi 2: Dengan `LanguageIdSettings` parameter. Gunakan opsi ini jika Anda menyertakan model bahasa khusus, kosakata khusus, atau filter kosakata khusus dalam permintaan Anda.

```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe')
job_name = "my-first-transcription-job"
job_uri = "s3://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
transcribe.start_transcription_job(
    TranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'amzn-s3-demo-bucket',
    OutputKey = 'my-output-files/',
    MediaFormat='flac',
    IdentifyLanguage=True,  (or IdentifyMultipleLanguages=True)
    LanguageOptions = [
        'en-US', 'hi-IN'
    ],
    LanguageIdSettings={
        'en-US': {
            'VocabularyName': 'my-en-US-vocabulary',
            'VocabularyFilterName': 'my-en-US-vocabulary-filter',
            'LanguageModelName': 'my-en-US-language-model'
        },
        'hi-IN': {
            'VocabularyName': 'my-hi-IN-vocabulary',
            'VocabularyFilterName': 'my-hi-IN-vocabulary-filter'
        }
    }
)

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

Identifikasi bahasa dengan transkripsi streaming

Identifikasi bahasa streaming dapat mengidentifikasi bahasa dominan yang digunakan dalam aliran media Anda. Amazon Transcribe membutuhkan minimal satu detik pidato untuk mengidentifikasi bahasa.

Jika aliran Anda hanya berisi satu bahasa, Anda dapat mengaktifkan identifikasi bahasa tunggal, yang mengidentifikasi bahasa dominan yang digunakan dalam file media Anda dan membuat transkrip Anda hanya menggunakan bahasa ini.

Jika streaming Anda berisi lebih dari satu bahasa, Anda dapat mengaktifkan identifikasi multi-bahasa, yang mengidentifikasi semua bahasa yang digunakan dalam aliran Anda dan membuat transkrip menggunakan setiap bahasa yang diidentifikasi. Perhatikan bahwa transkrip multi-bahasa diproduksi. Anda dapat menggunakan layanan lain, seperti Amazon Transcribe, untuk menerjemahkan transkrip Anda.

Untuk menggunakan identifikasi bahasa streaming, Anda harus memberikan setidaknya dua kode bahasa, dan Anda hanya dapat memilih satu dialek bahasa per bahasa per aliran. Ini berarti Anda tidak dapat memilih `en-US` dan `en-AU` sebagai opsi bahasa untuk transkripsi yang sama.

Anda juga memiliki opsi untuk memilih bahasa pilihan dari kumpulan kode bahasa yang Anda berikan. Menambahkan bahasa pilihan dapat mempercepat proses identifikasi bahasa, yang berguna untuk klip audio pendek.

> **Important**
>
> Jika tidak ada kode bahasa yang Anda berikan cocok dengan bahasa, atau bahasa, yang diidentifikasi dalam audio Anda, Amazon Transcribe pilih kecocokan bahasa terdekat dari kode bahasa yang Anda tentukan. Kemudian menghasilkan transkrip dalam bahasa itu. Misalnya, jika media Anda dalam bahasa Inggris AS (`en-US`) dan Anda memberikan Amazon Transcribe kode bahasa,, dan `zh-CN` `fr-FRde-DE`, Amazon Transcribe kemungkinan akan mencocokkan media Anda dengan bahasa Jerman (`de-DE`) dan menghasilkan transkripsi berbahasa Jerman. Ketidakcocokan kode bahasa dan bahasa lisan dapat mengakibatkan transkrip yang tidak akurat, jadi sebaiknya berhati-hati saat menyertakan kode bahasa.

Jika media Anda berisi dua saluran, Amazon Transcribe dapat mengidentifikasi bahasa dominan yang digunakan di setiap saluran. Dalam hal ini, atur `ChannelIdentification`parameter ke `true` dan setiap saluran ditranskripsi secara terpisah. Perhatikan bahwa default untuk parameter ini

adalah`false`. Jika Anda tidak mengubahnya, hanya saluran pertama yang ditranskripsi dan hanya satu bahasa yang diidentifikasi.

Identifikasi bahasa streaming tidak dapat digabungkan dengan model bahasa khusus atau redaksi. Jika menggabungkan identifikasi bahasa dengan fitur lain, Anda terbatas pada bahasa yang didukung dengan fitur tersebut, dan juga dengan transkripsi streaming. Lihat bahasa yang didukung.

> **Note**
>
> PCM dan FLAC adalah satu-satunya format audio yang didukung untuk identifikasi bahasa streaming. Untuk identifikasi multi-bahasa, hanya PCM yang didukung.

Mengidentifikasi bahasa dalam audio multi-bahasa

Identifikasi multi-bahasa ditujukan untuk aliran multi-bahasa, dan memberi Anda transkrip yang mencerminkan semua bahasa yang didukung yang digunakan dalam aliran Anda. Ini berarti bahwa jika penutur mengubah bahasa di tengah percakapan, atau jika setiap peserta berbicara bahasa yang berbeda, output transkripsi Anda mendeteksi dan mentranskripsikan setiap bahasa dengan benar.

Misalnya, jika streaming Anda berisi pembicara bilingual yang bergantian antara bahasa Inggris AS (`en-US`) dan Hindi (`hi-IN`), identifikasi multi-bahasa dapat mengidentifikasi dan mentranskripsikan bahasa Inggris AS lisan sebagai dan bahasa Hindi yang diucapkan sebagai`en-US`. `hi-IN` Ini berbeda dari identifikasi bahasa tunggal, di mana hanya satu bahasa dominan yang digunakan untuk membuat transkrip. Dalam hal ini, bahasa lisan apa pun yang bukan bahasa dominan ditranskripsikan secara tidak benar.

> **Note**
>
> Redaksi dan model bahasa kustom saat ini tidak didukung dengan identifikasi multi-bahasa.

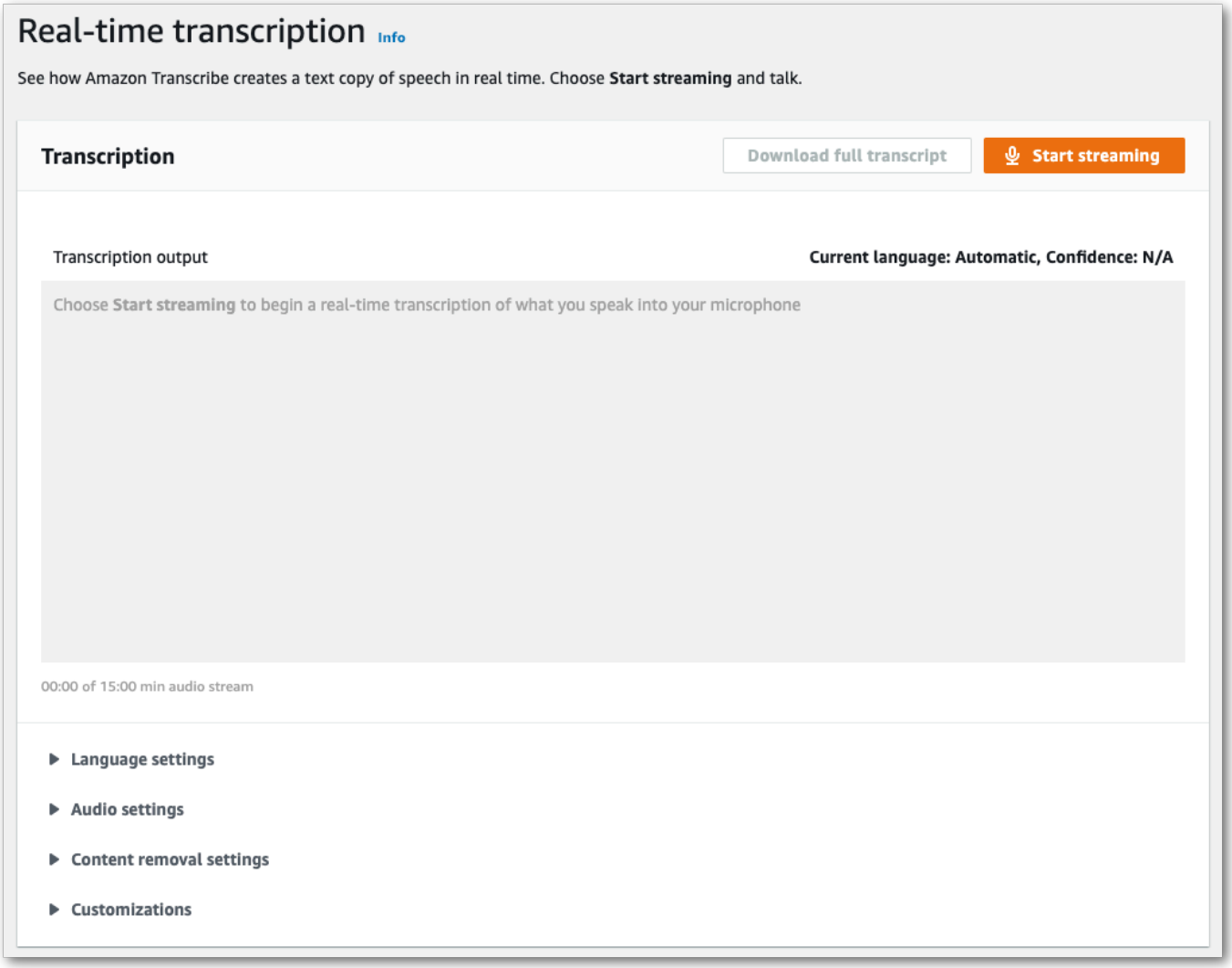
Menggunakan identifikasi bahasa dengan media streaming

Anda dapat menggunakan identifikasi bahasa otomatis dalam transkripsi streaming menggunakan AWS Management Console, HTTP/2, atau WebSockets; lihat contoh berikut:

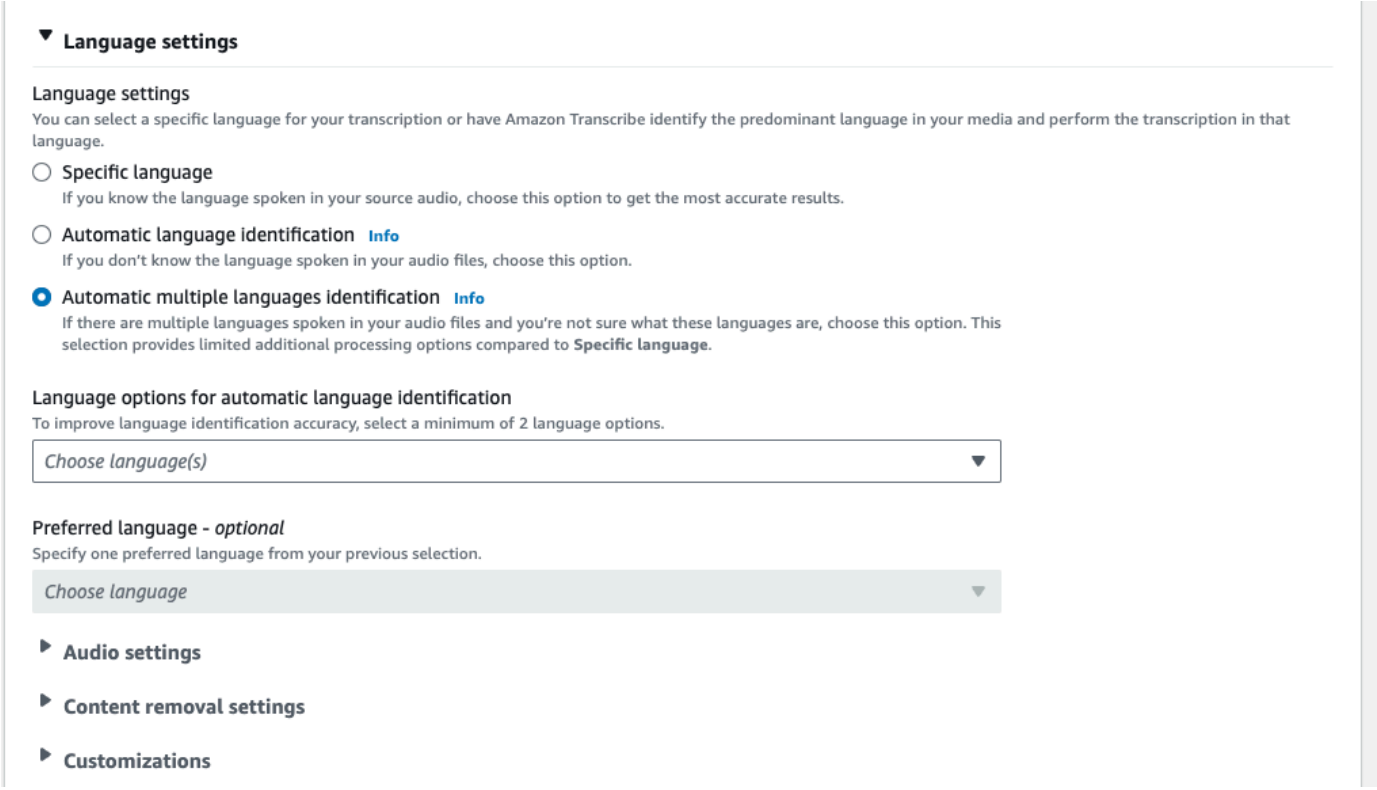
AWS Management Console

1. Masuk ke AWS Management Console.

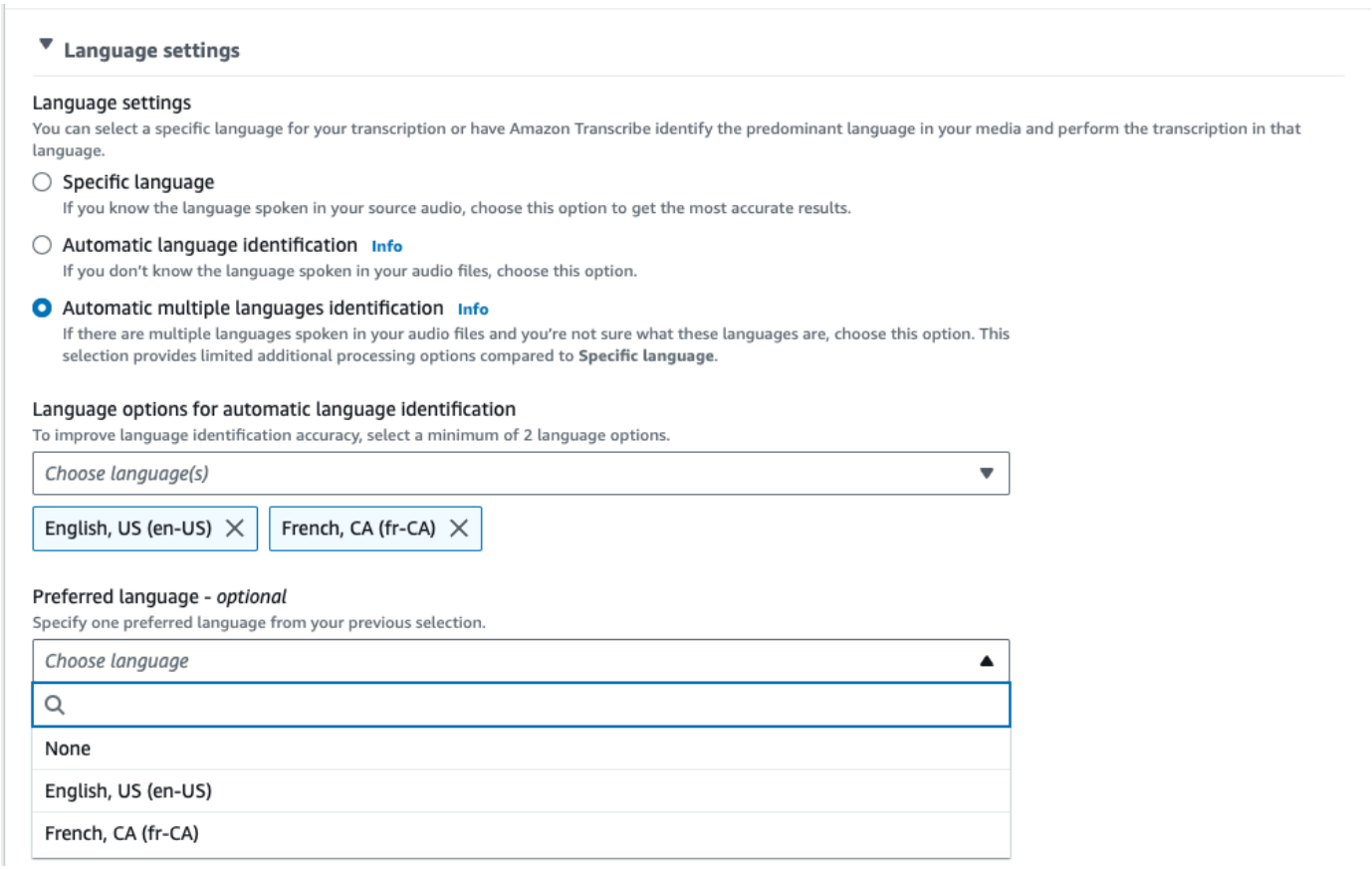
2. Di panel navigasi, pilih Transkripsi waktu nyata. Gulir ke bawah ke pengaturan Bahasa dan perluas bidang ini jika diminimalkan.



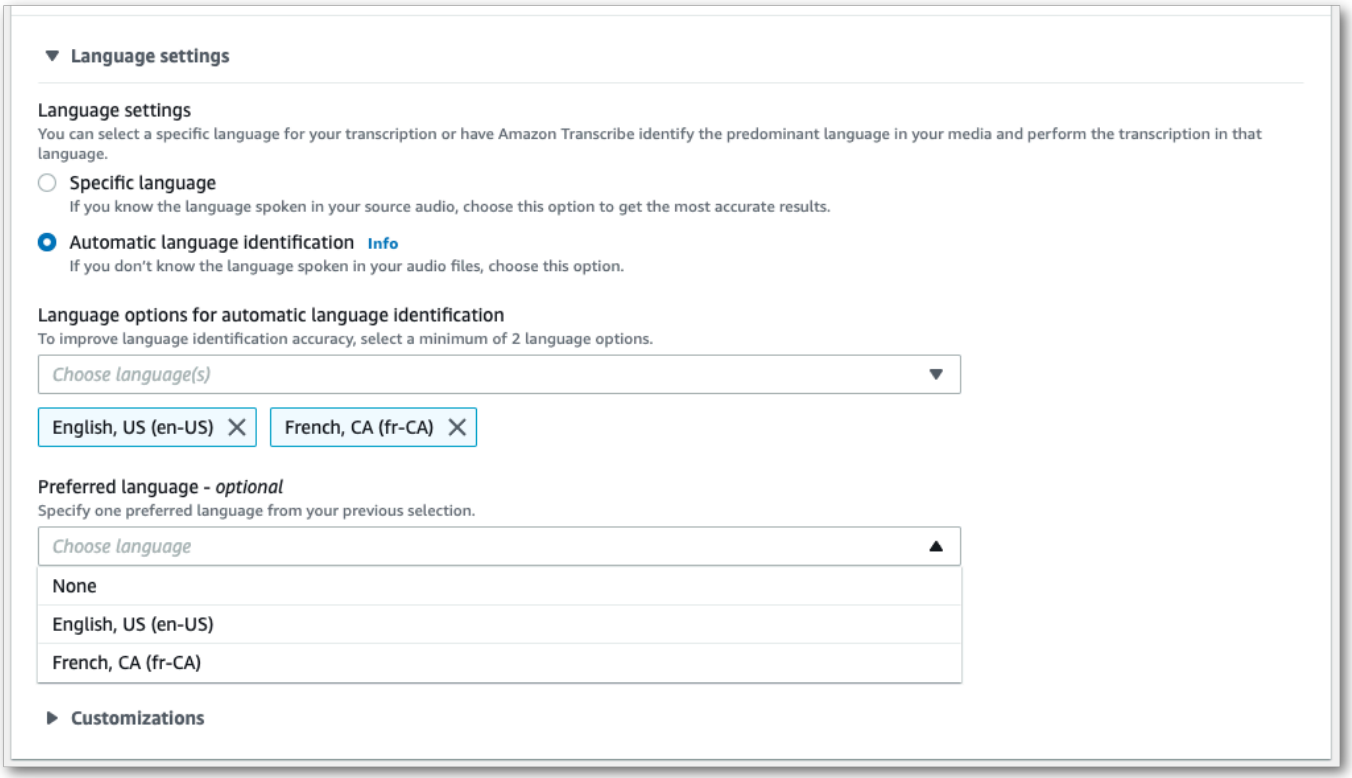
3. Pilih Identifikasi bahasa otomatis atau Identifikasi beberapa bahasa otomatis.



4. Berikan minimal dua kode bahasa untuk transkripsi Anda. Perhatikan bahwa Anda hanya dapat menyediakan satu dialek per bahasa. Misalnya, Anda tidak dapat memilih keduanya `en-US` dan `fr-CA` sebagai opsi bahasa untuk transkripsi yang sama.



5. (Opsional) Dari subset bahasa yang Anda pilih pada langkah sebelumnya, Anda dapat memilih bahasa yang disukai untuk transkrip Anda.



6. Anda sekarang siap untuk mentranskripsikan aliran Anda. Pilih Mulai streaming dan mulai berbicara. Untuk mengakhiri dikte Anda, pilih Hentikan streaming.

Aliran HTTP/2

Contoh ini membuat permintaan HTTP/2 dengan identifikasi bahasa diaktifkan. Untuk informasi lebih lanjut tentang menggunakan streaming HTTP/2 dengan Amazon Transcribe, lihat. Menyiapkan aliran HTTP/2 Untuk detail selengkapnya tentang parameter dan header khusus untuk Amazon Transcribe, lihat `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
```

Contoh ini membuat permintaan HTTP/2 dengan identifikasi beberapa bahasa diaktifkan. Untuk informasi lebih lanjut tentang menggunakan streaming HTTP/2 dengan Amazon Transcribe, lihat. Menyiapkan aliran HTTP/2 Untuk detail selengkapnya tentang parameter dan header khusus untuk Amazon Transcribe, lihat `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
```

Jika Anda menggunakan `identify-language` atau `identify-multiple-languages` dalam permintaan Anda, Anda juga harus menyertakan`language-options`. Anda tidak dapat menggunakan keduanya `language-code` dan `identify-language` dalam permintaan yang sama.

Definisi parameter dapat ditemukan di Referensi API; parameter yang umum untuk semua operasi AWS API tercantum di bagian Parameter Umum.

WebSocket aliran

Contoh ini membuat URL presigned yang menggunakan identifikasi bahasa dalam WebSocket aliran. Jeda baris telah ditambahkan untuk keterbacaan. Untuk informasi selengkapnya tentang penggunaan WebSocket stream dengan Amazon Transcribe, lihatMenyiapkan WebSocket aliran. Untuk detail lebih lanjut tentang parameter, lihat `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
```

Contoh ini membuat URL presigned yang menggunakan identifikasi beberapa bahasa dalam WebSocket aliran. Jeda baris telah ditambahkan untuk keterbacaan. Untuk informasi selengkapnya tentang penggunaan WebSocket stream dengan Amazon Transcribe, lihatMenyiapkan WebSocket aliran. Untuk detail lebih lanjut tentang parameter, lihat `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
```

Jika Anda menggunakan `identify-language` atau `identify-multiple-languages` dalam permintaan Anda, Anda juga harus menyertakan`language-options`. Anda tidak dapat menggunakan keduanya `language-code` dan `identify-language` dalam permintaan yang sama.

Definisi parameter dapat ditemukan di Referensi API; parameter yang umum untuk semua operasi AWS API tercantum di bagian Parameter Umum.

Transkripsi alternatif

Saat Amazon Transcribe mentranskripsikan audio, itu membuat versi berbeda dari transkrip yang sama dan memberikan skor kepercayaan untuk setiap versi. Dalam transkripsi tipikal, Anda hanya mendapatkan versi dengan skor kepercayaan tertinggi.

Jika Anda mengaktifkan transkripsi alternatif, Amazon Transcribe kembalikan versi lain dari transkrip Anda yang memiliki tingkat kepercayaan lebih rendah. Anda dapat memilih untuk mengembalikan hingga 10 transkripsi alternatif. Jika Anda menentukan jumlah alternatif yang lebih besar daripada yang Amazon Transcribe mengidentifikasi, hanya jumlah sebenarnya dari alternatif yang dikembalikan.

Semua alternatif terletak di file keluaran transkripsi yang sama dan disajikan pada tingkat segmen. Segmen adalah jeda alami dalam pidato, seperti perubahan speaker atau jeda dalam audio.

Transkripsi alternatif hanya tersedia untuk transkripsi batch.

Output transkripsi Anda disusun sebagai berikut. Elips (`...`) dalam contoh kode menunjukkan di mana konten telah dihapus untuk singkatnya.

1. Transkripsi akhir lengkap untuk segmen tertentu.

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

2. Skor kepercayaan untuk setiap kata di bagian sebelumnya. `transcript`

```
"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. Transkripsi alternatif Anda terletak di `segments` bagian output transkripsi Anda. Alternatif untuk setiap segmen diurutkan berdasarkan skor kepercayaan turun.

```
"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. Status di akhir keluaran transkripsi Anda.

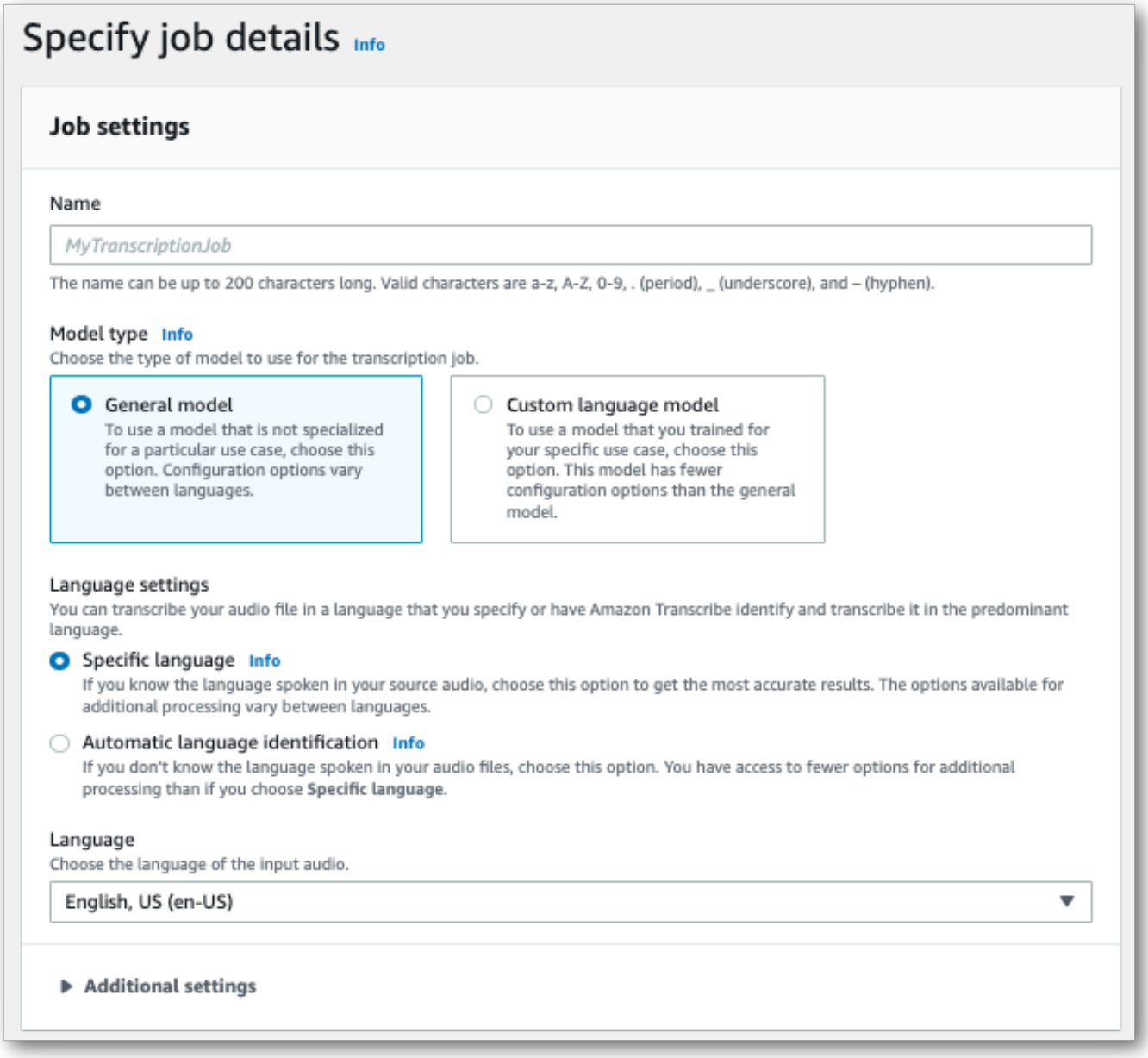
```
"status": "COMPLETED"
}
```

Meminta transkripsi alternatif

Anda dapat meminta transkripsi alternatif menggunakan AWS Management Console, AWS CLI, atau AWS SDKs; lihat contoh berikut:

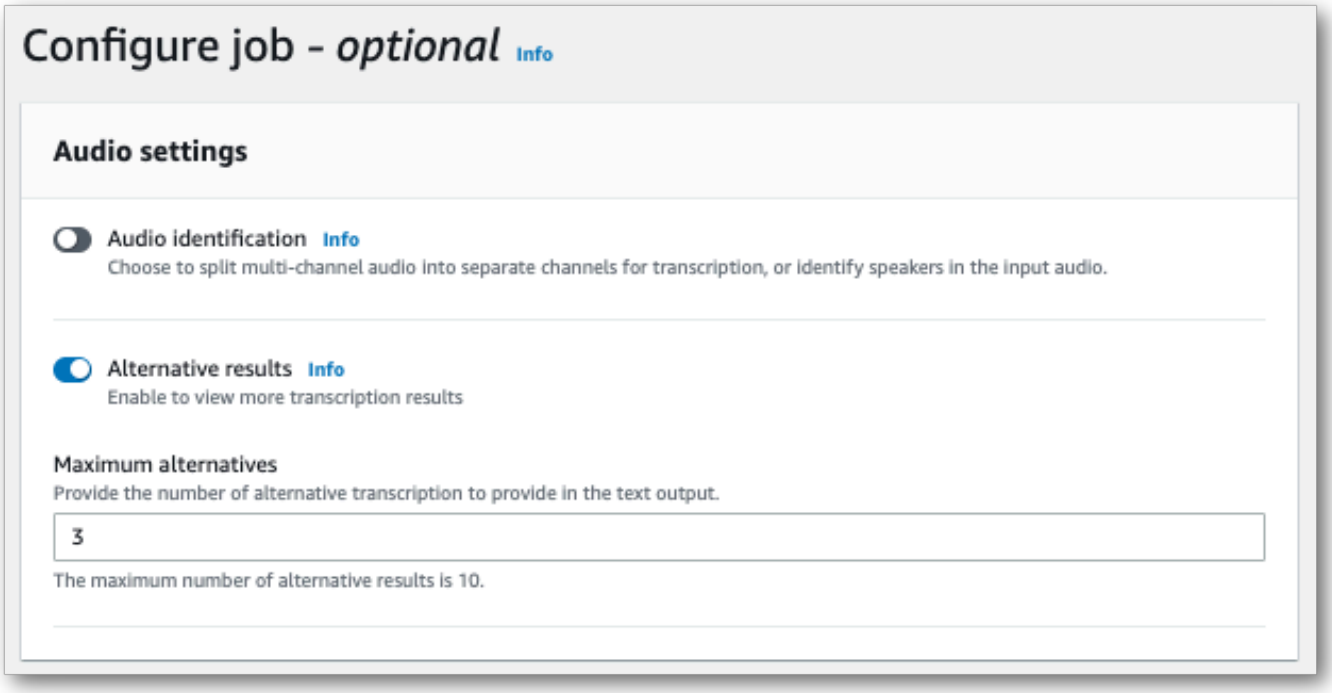
AWS Management Console

1. Masuk ke AWS Management Console.
2. Di panel navigasi, pilih Pekerjaan transkripsi, lalu pilih Buat pekerjaan (kanan atas). Ini membuka halaman Tentukan detail pekerjaan.



3. Isi kolom yang ingin Anda sertakan di halaman Tentukan detail pekerjaan, lalu pilih Berikutnya. Ini membawa Anda ke halaman Konfigurasi pekerjaan - opsional.

   Pilih Hasil alternatif dan tentukan jumlah maksimum hasil transkripsi alternatif yang Anda inginkan dalam transkrip Anda.



4. Pilih Buat pekerjaan untuk menjalankan pekerjaan transkripsi Anda.

AWS CLI

Contoh ini menggunakan start-transcription-jobperintah dan `ShowAlternatives` parameter. Untuk informasi selengkapnya, silakan lihat `StartTranscriptionJob` dan `ShowAlternatives`.

Perhatikan bahwa jika Anda memasukkan `ShowAlternatives=true` dalam permintaan Anda, Anda juga harus menyertakan`MaxAlternatives`.

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

Berikut contoh lain menggunakan start-transcription-jobperintah, dan badan permintaan yang menyertakan transkripsi alternatif.

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

```
--region us-west-2 \
--cli-input-json file://filepath/my-first-alt-transcription-job.json
```

File my-first-alt-transcription-job.json berisi badan permintaan berikut.

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

AWS SDK untuk Python (Boto3)

Contoh berikut menggunakan AWS SDK untuk Python (Boto3) untuk meminta transkripsi alternatif dengan menggunakan `ShowAlternatives` argumen untuk metode start_transcription_job. Untuk informasi selengkapnya, silakan lihat `StartTranscriptionJob` dan `ShowAlternatives`.

Untuk contoh tambahan menggunakan AWS SDKs, termasuk contoh khusus fitur, skenario, dan lintas layanan, lihat bagian ini. Contoh kode untuk Amazon Transcribe menggunakan AWS SDKs

Perhatikan bahwa jika Anda memasukkan `'ShowAlternatives':True` dalam permintaan Anda, Anda juga harus menyertakan`MaxAlternatives`.

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

```
    OutputBucketName = 'amzn-s3-demo-bucket',
    OutputKey = 'my-output-files/',
    LanguageCode = 'en-US',
    Settings = {
        'ShowAlternatives':True,
        'MaxAlternatives':4
    }
)

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

Meningkatkan akurasi transkripsi dengan kosakata khusus dan model bahasa khusus

Jika media Anda berisi istilah khusus domain atau non-standar, seperti nama merek, akronim, kata teknis, dan jargon, Amazon Transcribe mungkin tidak menangkap istilah ini dengan benar dalam keluaran transkripsi Anda.

Untuk memperbaiki ketidakakuratan transkripsi dan menyesuaikan output Anda untuk kasus penggunaan spesifik Anda, Anda dapat membuat dan. Kosakata khusus Model bahasa khusus

- Kosakata khususdirancang untuk menyetel dan meningkatkan pengenalan dan pemformatan kata-kata tertentu dalam semua konteks. Ini melibatkan penyediaan Amazon Transcribe kata-kata dan, opsional, pengucapan dan bentuk tampilan.

  Jika Amazon Transcribe tidak merender istilah tertentu dengan benar dalam transkrip Anda, Anda dapat membuat file kosakata khusus yang memberi tahu Amazon Transcribe bagaimana Anda ingin istilah ini ditampilkan. Pendekatan khusus kata ini paling tepat untuk mengoreksi istilah seperti nama merek dan akronim.
- Model bahasa khususdirancang untuk menangkap konteks yang terkait dengan istilah. Ini melibatkan penyediaan Amazon Transcribe dengan volume besar data teks khusus domain.

  Jika Amazon Transcribe tidak merender istilah teknis dengan benar atau menggunakan homofon yang salah dalam transkrip Anda, Anda dapat membuat model bahasa khusus yang mengajarkan Amazon Transcribe bahasa khusus domain Anda. Misalnya, model bahasa khusus dapat mempelajari kapan harus menggunakan 'floe' (ice floe) versus 'flow' (aliran linier).

  Pendekatan sadar konteks ini paling tepat untuk menyalin volume besar pidato spesifik domain. Model bahasa khusus dapat menghasilkan peningkatan akurasi yang signifikan dibandingkan kosakata khusus saja. Saat menggunakan transkripsi batch, Anda dapat menyertakan model bahasa khusus dan kosakata khusus dalam permintaan Anda.

> **Tip**
>
> Untuk mencapai akurasi transkripsi tertinggi, gunakan kosakata khusus bersama dengan model bahasa khusus Anda.

Untuk demo video tentang cara membuat kosakata khusus menggunakan AWS Management Console, lihat Menggunakan kosakata khusus.

Untuk demo video tentang cara membuat dan menggunakan model bahasa khusus, lihat Menggunakan Model Bahasa Kustom (CLM) untuk meningkatkan akurasi transkripsi.

Selami lebih dalam dengan Blog AWS Machine Learning

Kosakata khusus:

- Transkripsi langsung dari balapan F1 menggunakan Amazon Transcribe

Model bahasa khusus:

- Membangun model bahasa khusus untuk meningkatkan kinerja speech-to-text Amazon Transcribe
- Tingkatkan akurasi transkripsi kuliah kelas dengan model bahasa khusus untuk Amazon Transcribe

Kosakata khusus

Gunakan kosakata khusus untuk meningkatkan akurasi transkripsi untuk satu atau lebih kata tertentu. Ini umumnya istilah khusus domain, seperti nama merek dan akronim, kata benda yang tepat, dan kata-kata yang tidak ditampilkan dengan benar. Amazon Transcribe

Kosakata khusus dapat digunakan dengan semua bahasa yang didukung. Perhatikan bahwa hanya karakter yang tercantum dalam kumpulan karakter bahasa Anda yang dapat digunakan dalam kosakata khusus.

Important

Anda bertanggung jawab atas integritas data Anda sendiri saat Anda menggunakannya Amazon Transcribe. Jangan memasukkan informasi rahasia, informasi pribadi (PII), atau informasi kesehatan yang dilindungi (PHI) ke dalam kosakata khusus.

Pertimbangan saat membuat kosakata khusus:

- Anda dapat memiliki hingga 100 file kosakata khusus per Akun AWS
- Batas ukuran untuk setiap file kosakata kustom adalah 50 Kb
- Jika menggunakan API untuk membuat kosakata kustom Anda, file kosakata Anda harus dalam format teks (*.txt). Jika menggunakan AWS Management Console, file kosakata Anda dapat dalam format teks (*.txt) atau format nilai dipisahkan koma (*.csv).
- Setiap entri dalam kosakata kustom tidak boleh melebihi 256 karakter
- Untuk menggunakan kosakata khusus, itu harus dibuat Wilayah AWS sama dengan transkripsi Anda.

> **Tip**
>
> Anda dapat menguji kosakata khusus Anda menggunakan. AWS Management Console Setelah kosakata kustom Anda siap digunakan, masuk ke, pilih Transkripsi waktu nyata AWS Management Console, gulir ke Kustomisasi, aktifkan kosakata khusus, dan pilih kosakata khusus Anda dari daftar tarik-turun. Kemudian pilih mulai streaming. Ucapkan beberapa kata dalam kosakata khusus Anda ke mikrofon Anda untuk melihat apakah kata-kata tersebut ditampilkan dengan benar.

Tabel kosakata khusus versus daftar

> **Important**
>
> Kosakata kustom dalam format daftar tidak digunakan lagi. Jika Anda membuat kosakata kustom baru, gunakan format tabel.

Tabel memberi Anda lebih banyak pilihan untuk — dan lebih banyak kontrol atas — input dan output kata dalam kosakata khusus Anda. Dengan tabel, Anda harus menentukan beberapa kategori (Phrase and DisplayAs), memungkinkan Anda untuk menyempurnakan output Anda.

Daftar tidak memiliki opsi tambahan, jadi Anda hanya dapat mengetikkan entri seperti yang Anda inginkan untuk muncul di transkrip Anda, menggantikan semua spasi dengan tanda hubung.

The AWS Management Console, AWS CLI, dan AWS SDKs semua menggunakan tabel kosakata kustom dengan cara yang sama; daftar digunakan secara berbeda untuk setiap metode dan dengan

demikian mungkin memerlukan pemformatan tambahan untuk keberhasilan penggunaan antar metode.

Untuk informasi selengkapnya, silakan lihat Membuat kosakata khusus menggunakan tabel dan Membuat kosakata khusus menggunakan daftar.

Untuk menyelam lebih dalam dan mempelajari cara menggunakan Amazon Augmented AI dengan kosakata khusus, lihat Mulai membuat ulasan manusia bersama dengan Amazon Transcribe

> Operasi API khusus untuk kosakata khusus
>
> `CreateVocabulary`, `DeleteVocabulary`, `GetVocabulary`, `ListVocabularies`, `UpdateVocabulary`

Membuat kosakata khusus menggunakan tabel

Menggunakan format tabel adalah cara yang lebih disukai untuk membuat kosakata kustom Anda. Tabel kosakata harus terdiri dari empat kolom (Phrase, SoundsLike, IPA, and DisplayAs), yang dapat dimasukkan dalam urutan apa pun:

Frase	SoundsLike	IPA	DisplayAs
Wajib. Setiap baris dalam tabel Anda harus berisi entri di kolom ini. Jangan gunakan spasi di kolom ini. Jika entri Anda berisi beberapa kata, pisahkan setiap kata dengan tanda hubung (-). Misalnya, **`Andorra-la-Vella`** atau **`Los-Angeles`** .	`SoundsLike` tidak lagi didukung untuk Kosakata Kustom. Silakan biarkan kolom kosong. Nilai apa pun di kolom ini akan diabaikan. Kami akan menghapus dukungan untuk kolom ini di masa mendatang.	`IPA`tidak lagi didukung untuk Kosakata Kustom. Silakan biarkan kolom kosong. Nilai apa pun di kolom ini akan diabaikan. Kami akan menghapus dukungan untuk kolom ini di masa mendatang.	Tidak wajib. Baris di kolom ini dapat dibiarkan kosong. Anda dapat menggunakan spasi di kolom ini. Mendefinisikan bagaimana Anda ingin entri Anda terlihat dalam output transkripsi Anda. Misalnya, **`Andorra-la-Vella`** di `Phrase` kolom ada

Frase	SoundsLike	IPA	DisplayAs
Untuk akronim, setiap huruf yang diucapkan harus dipisahkan oleh titik. Periode trailing juga perlu diucapkan . Jika akronim Anda jamak, Anda harus menggunakan tanda hubung antara akronim dan 's'. Misalnya, 'CLI' adalah **`C.L.I.`** (tidak**`C.L.I`**) dan 'ABCs' adalah **`A.B.C.-s`** (tidak**`A.B.C-s`**). Jika frasa Anda terdiri dari kata dan akronim, kedua komponen ini harus dipisahkan oleh tanda hubung. Misalnya, 'DynamoDB ' adalah. **`Dynamo-D. B.`** Jangan sertakan digit di kolom ini; angka harus dijabarkan. Misalnya, 'VX02Q' adalah. **`V.X.-zero -two-Q.`**			**`Andorra la Vella`** di `DisplayAs` kolom. Jika baris di kolom ini kosong, Amazon Transcribe gunakan isi `Phrase` kolom untuk menentukan output. Anda dapat menyertakan digit (`0-9`) di kolom ini.

Hal-hal yang perlu diperhatikan saat membuat tabel Anda:

- Tabel Anda harus berisi keempat header kolom (Phrase, SoundsLike, IPA, and DisplayAs). `Phrase`Kolom harus berisi entri pada setiap baris. Kemampuan untuk memberikan input pengucapan melalui `IPA` dan `SoundsLike` tidak lagi didukung dan Anda dapat membiarkan kolom kosong. Nilai apa pun di kolom ini akan diabaikan.
- Setiap kolom harus TAB atau koma (,) digambarkan; ini berlaku untuk setiap baris dalam file kosakata kustom Anda. Jika baris berisi kolom kosong, Anda masih harus menyertakan delineator (TAB atau koma) untuk setiap kolom.
- Spasi hanya diperbolehkan di dalam `DisplayAs` kolom `IPA` dan. Jangan gunakan spasi untuk memisahkan kolom.
- `IPA`dan `SoundsLike` tidak lagi didukung untuk Kosakata Kustom. Silakan biarkan kolom kosong. Nilai apa pun di kolom ini akan diabaikan. Kami akan menghapus dukungan untuk kolom ini di masa mendatang.
- `DisplayAs`Kolom mendukung simbol dan karakter khusus (misalnya, C ++). Semua kolom lain mendukung karakter yang tercantum di halaman set karakter bahasa Anda.
- Jika Anda ingin memasukkan angka di `Phrase` kolom, Anda harus mengejanya. Digit (`0-9`) hanya didukung di `DisplayAs` kolom.
- Anda harus menyimpan tabel Anda sebagai file plaintext (*.txt) dalam format. `LF` Jika Anda menggunakan format lain, sepertiCRLF, kosakata kustom Anda tidak dapat diproses.
- Anda harus mengunggah file kosakata khusus Anda ke dalam Amazon S3 ember dan memprosesnya `CreateVocabulary`sebelum Anda dapat memasukkannya ke dalam permintaan transkripsi. Lihat Membuat tabel kosakata khusus untuk instruksi.

> **Note**
>
> Masukkan akronim, atau kata lain yang hurufnya harus diucapkan satu per satu, sebagai huruf tunggal yang dipisahkan oleh titik (**`A.B.C.`**). Untuk memasukkan bentuk jamak akronim, seperti ", pisahkan ABCs 's' dari akronim dengan tanda hubung (). **`A.B.C.-s`** Anda dapat menggunakan huruf besar atau kecil untuk mendefinisikan akronim. Akronim tidak didukung dalam semua bahasa; lihat. Bahasa yang didukung dan fitur khusus bahasa

Berikut adalah contoh tabel kosakata kustom (di mana **`[TAB]`** mewakili karakter tab):

```
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
```

Untuk kejernihan visual, berikut adalah tabel yang sama dengan kolom sejajar. Jangan menambahkan spasi antar kolom dalam tabel kosakata kustom Anda; tabel Anda akan terlihat tidak sejajar seperti contoh sebelumnya.

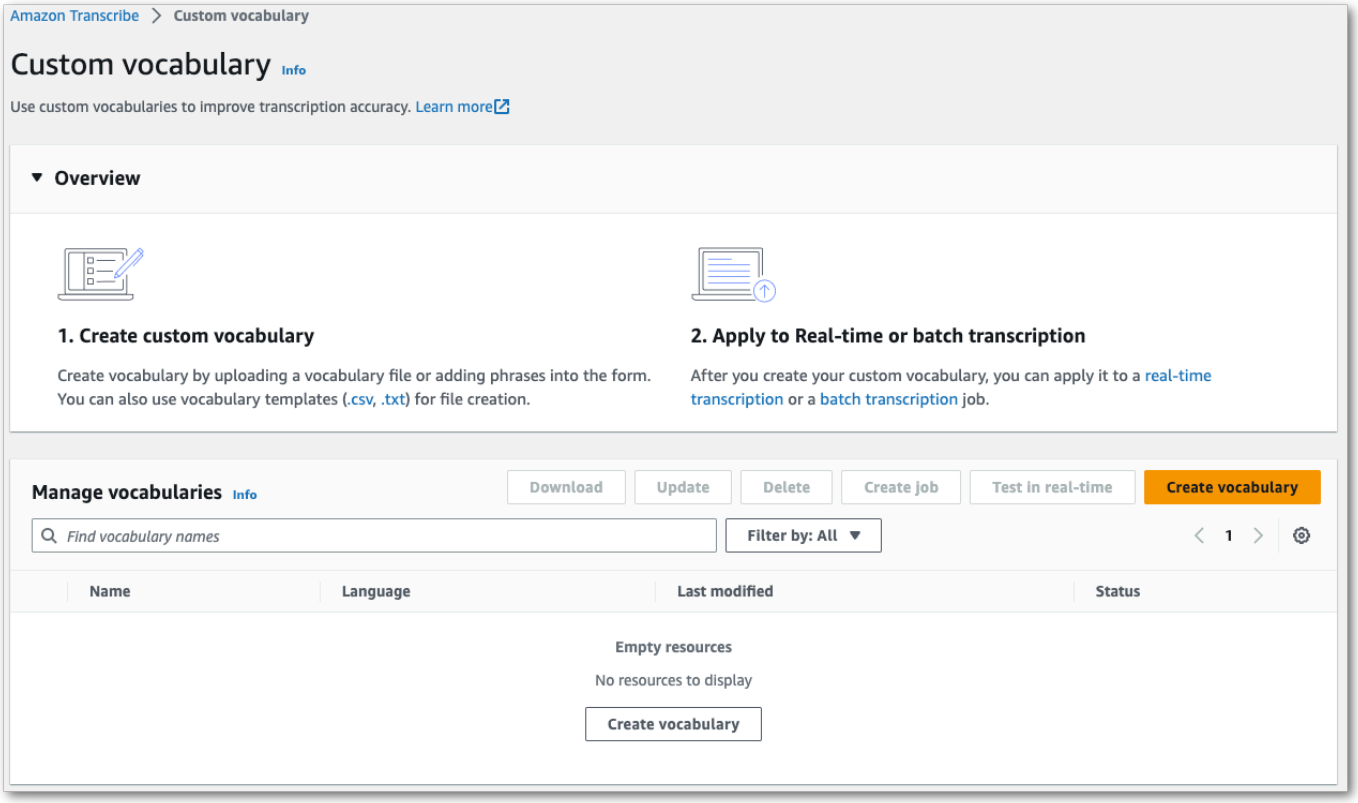
```
Phrase          [TAB]SoundsLike         [TAB]IPA                [TAB]DisplayAs
Los-Angeles     [TAB]                   [TAB]                   [TAB]Los Angeles
Eva-Maria       [TAB]                   [TAB]                   [TAB]
A.B.C.-s        [TAB]                   [TAB]                   [TAB]ABCs
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
```

Membuat tabel kosakata khusus

Untuk memproses tabel kosakata khusus untuk digunakan Amazon Transcribe, lihat contoh berikut:

AWS Management Console

1. Masuk ke AWS Management Console.
2. Di panel navigasi, pilih Kosakata khusus. Ini membuka halaman kosakata kustom di mana Anda dapat melihat kosakata yang ada atau membuat yang baru.
3. Pilih Buat kosakata.

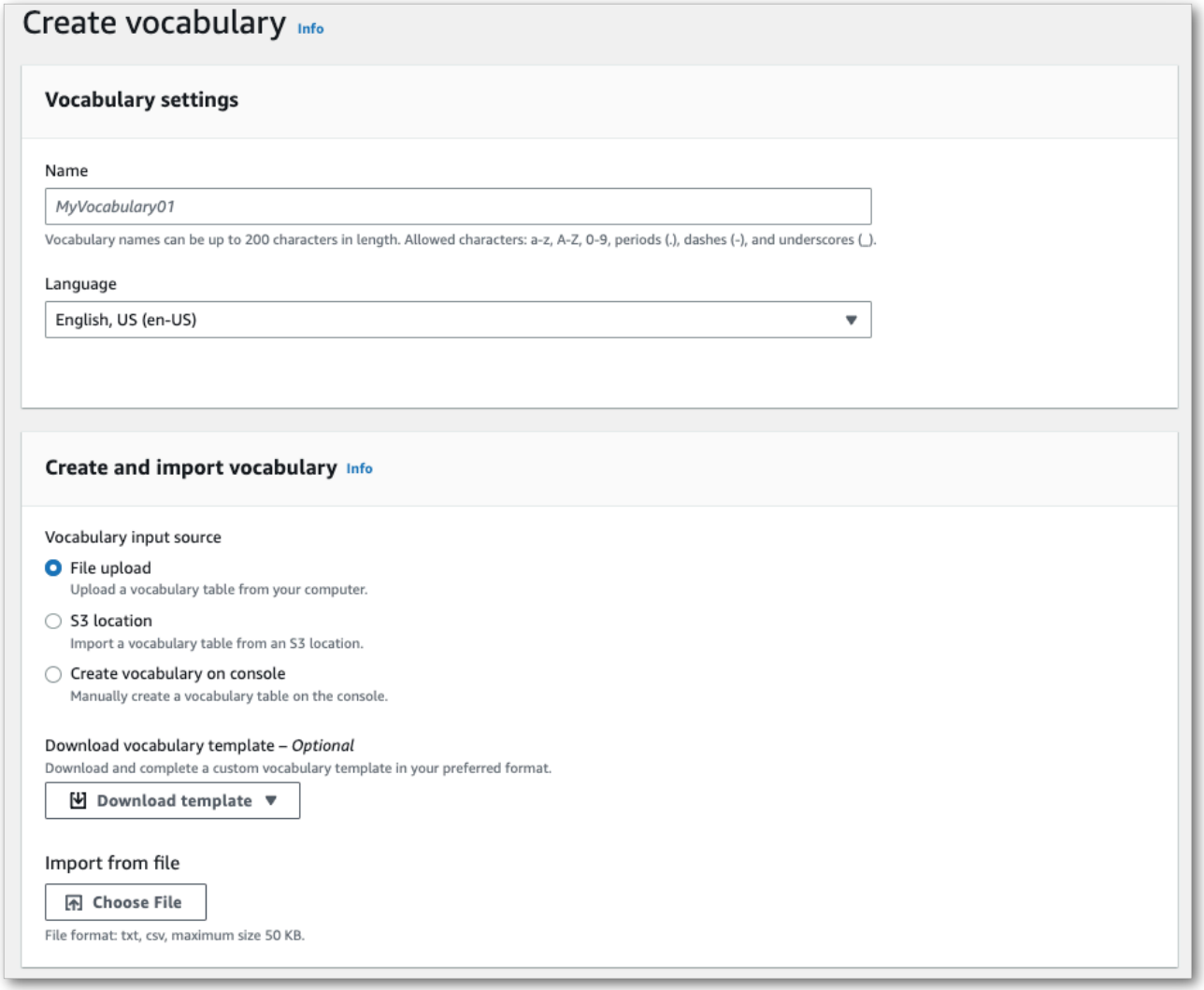


Ini membawa Anda ke halaman Buat kosakata. Masukkan nama untuk kosakata kustom baru Anda.

Di sini, Anda memiliki tiga opsi:

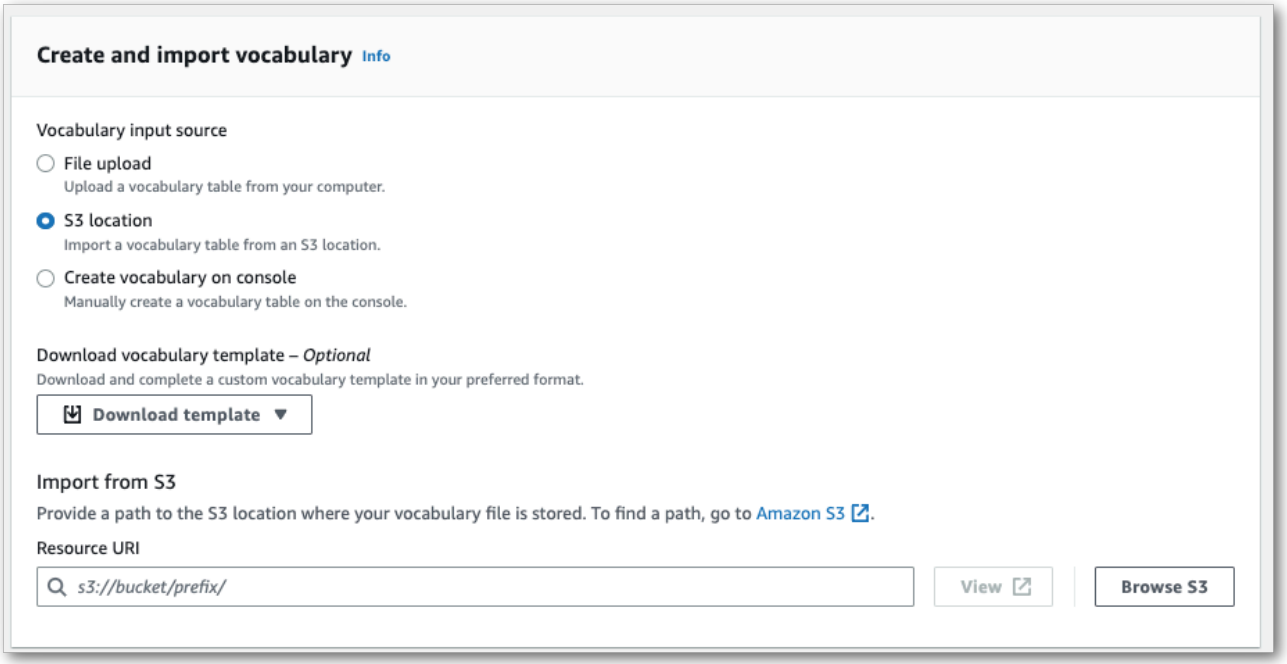
a. Unggah file txt atau csv dari komputer Anda.

   Anda dapat membuat kosakata khusus dari awal atau mengunduh templat untuk membantu Anda memulai. Kosakata Anda kemudian diisi otomatis di panel Lihat dan edit kosakata.



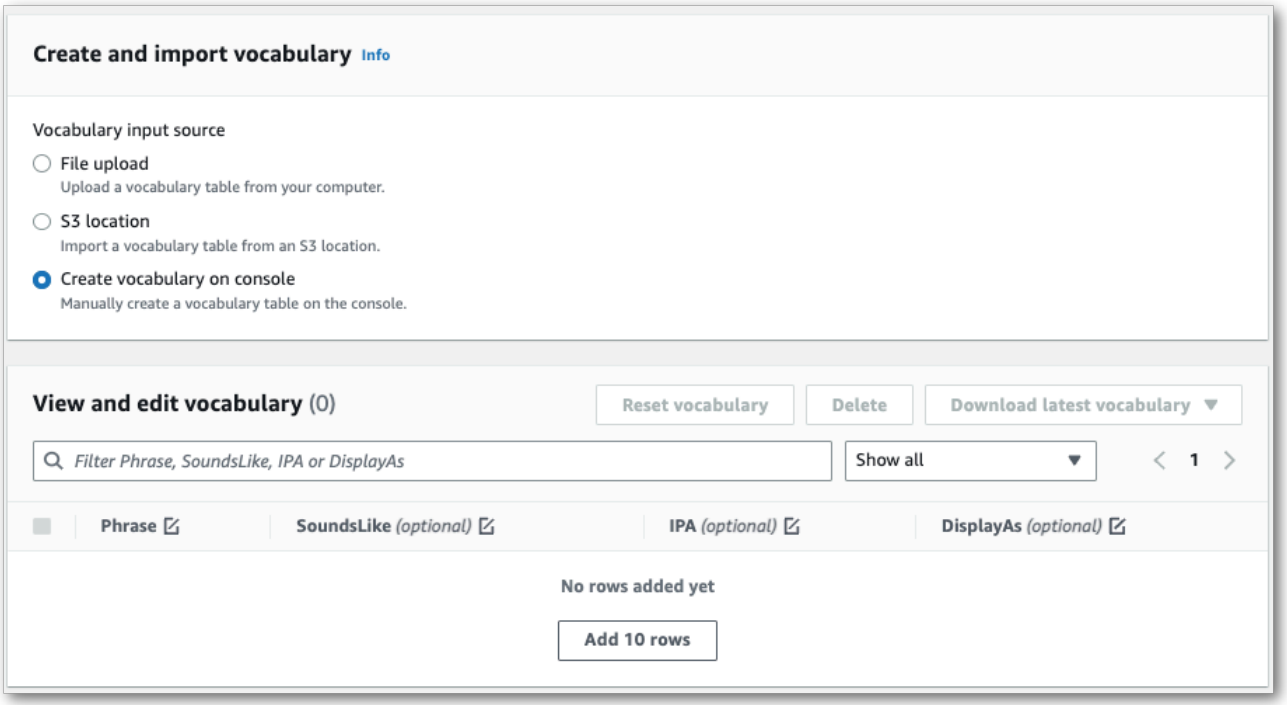
b. Impor file txt atau csv dari suatu lokasi. Amazon S3

Anda dapat membuat kosakata khusus dari awal atau mengunduh templat untuk membantu Anda memulai. Unggah file kosakata Anda yang sudah selesai ke Amazon S3 bucket dan tentukan URI dalam permintaan Anda. Kosakata Anda kemudian diisi otomatis di panel Lihat dan edit kosakata.

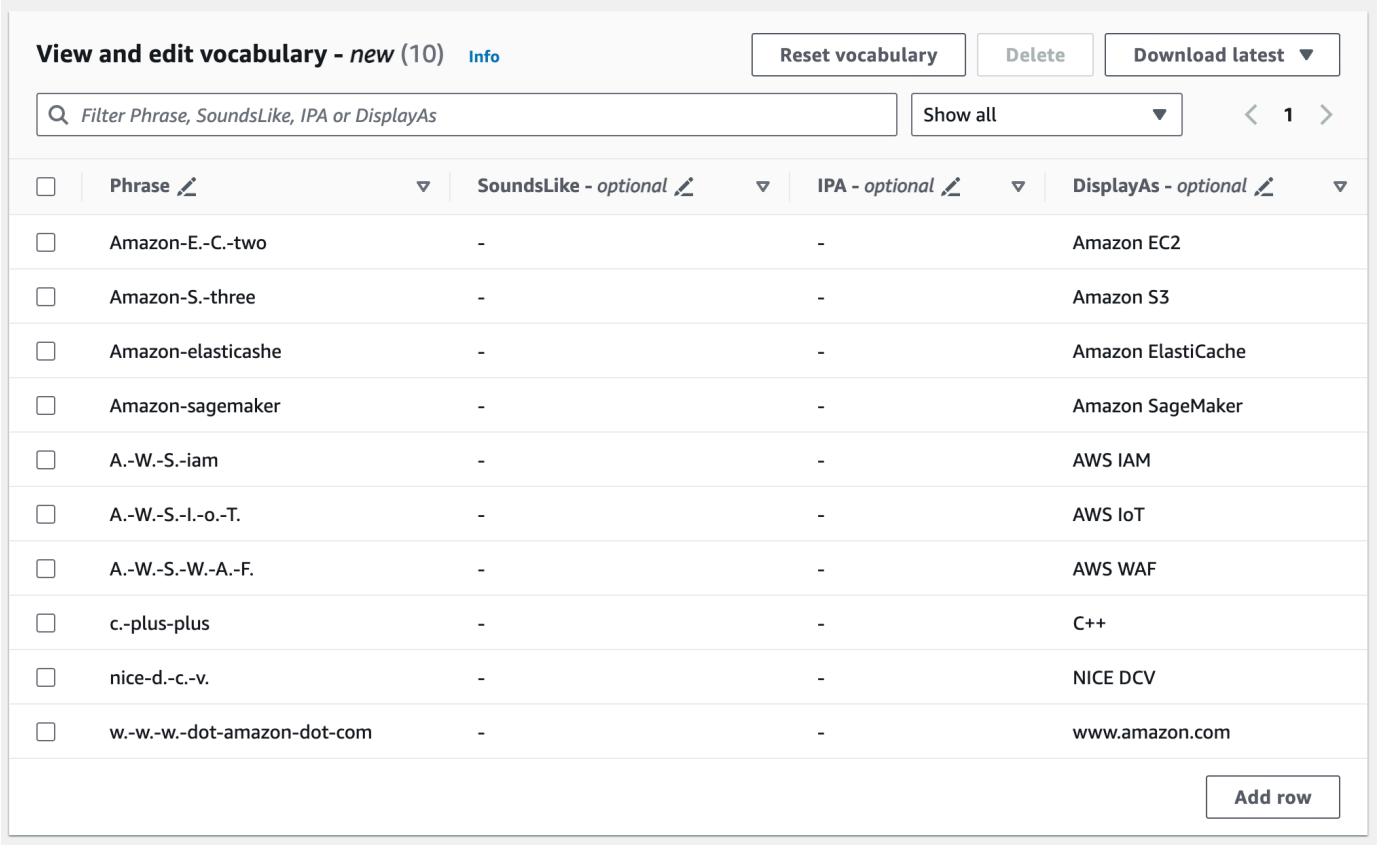


c. Buat kosakata Anda secara manual di konsol.

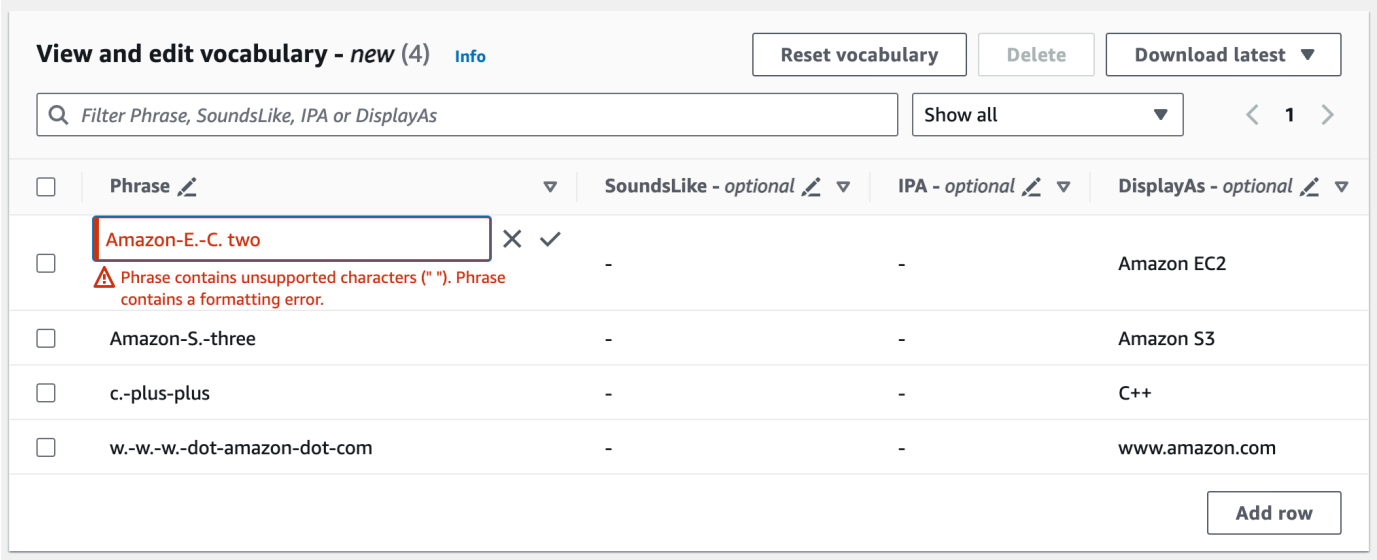
Gulir ke panel Lihat dan edit kosakata dan pilih Tambahkan 10 baris. Anda sekarang dapat memasukkan istilah secara manual.



4. Anda dapat mengedit kosakata Anda Lihat dan mengedit panel kosakata. Untuk membuat perubahan, klik pada entri yang ingin Anda modifikasi.

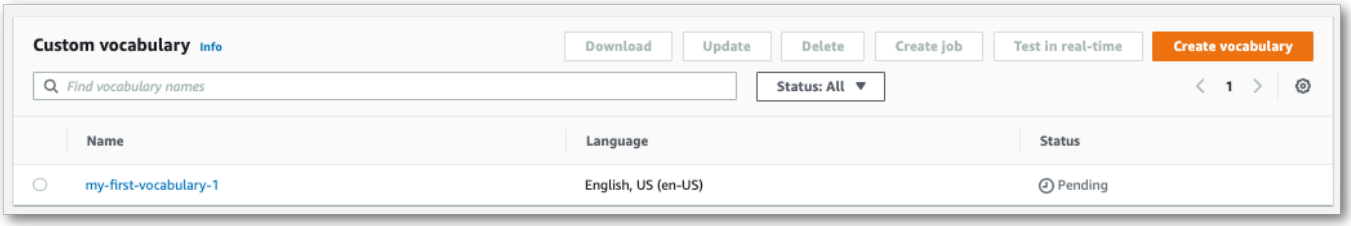


Jika Anda membuat kesalahan, Anda mendapatkan pesan kesalahan terperinci sehingga Anda dapat memperbaiki masalah apa pun sebelum memproses kosakata Anda. Perhatikan bahwa jika Anda tidak memperbaiki semua kesalahan sebelum memilih Buat kosakata, permintaan kosakata Anda gagal.

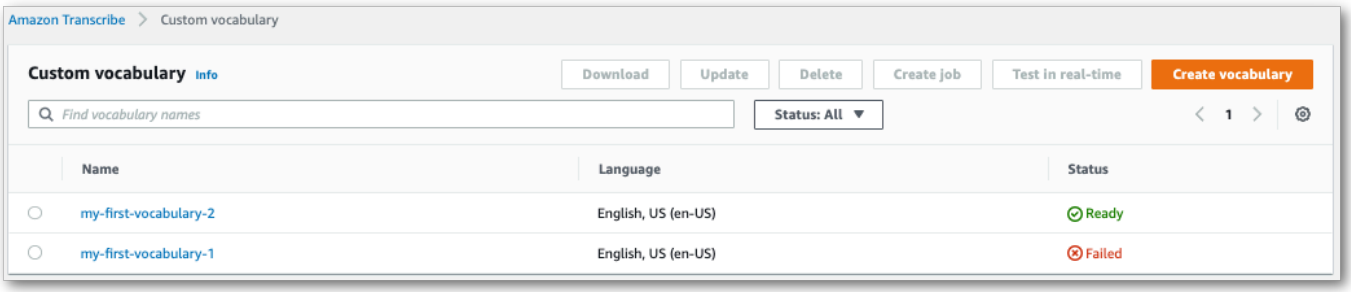


Pilih tanda centang (✓) untuk menyimpan perubahan Anda atau 'X' untuk membuang perubahan Anda.

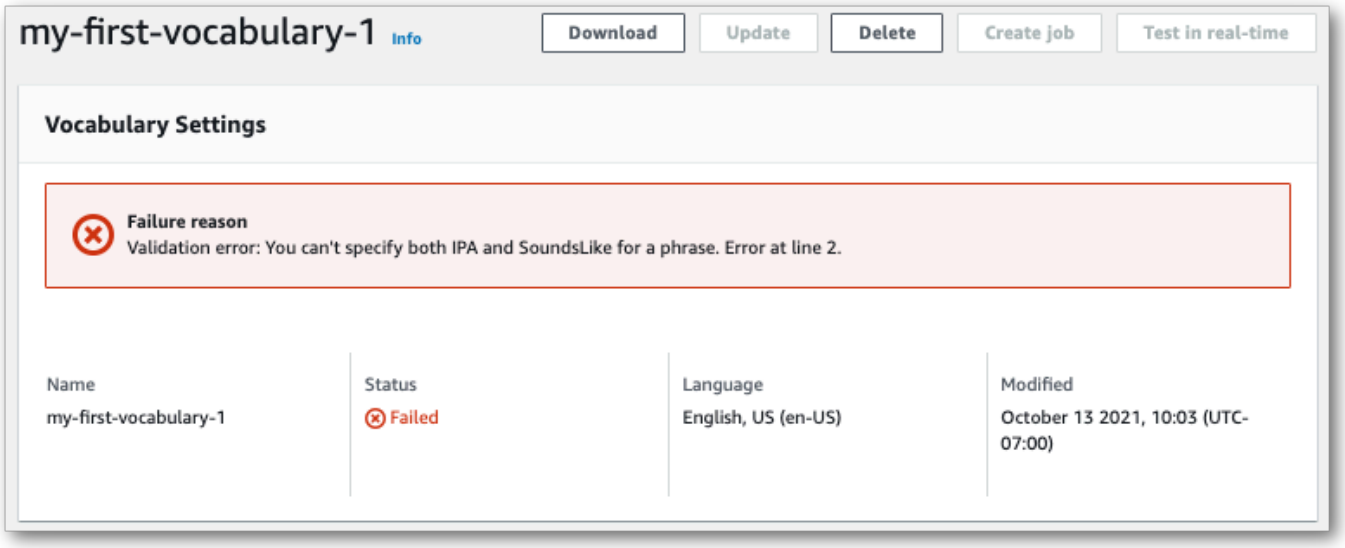
5. Secara opsional, tambahkan tag ke kosakata kustom Anda. Setelah semua bidang selesai dan puas dengan kosakata Anda, pilih Buat kosakata di bagian bawah halaman. Ini membawa Anda kembali ke halaman kosakata Kustom di mana Anda dapat melihat status kosakata kustom Anda. Ketika status berubah dari 'Tertunda' menjadi 'Siap' kosakata kustom Anda dapat digunakan dengan transkripsi.



6. Jika status berubah menjadi 'Gagal', pilih nama kosakata khusus Anda untuk membuka halaman informasinya.



Ada spanduk alasan kegagalan di bagian atas halaman ini yang memberikan informasi tentang mengapa kosakata kustom Anda gagal. Perbaiki kesalahan dalam file teks Anda dan coba lagi.



AWS CLI

Contoh ini menggunakan perintah create-vocabulary dengan file kosakata yang diformat tabel. Untuk informasi selengkapnya, lihat `CreateVocabulary`.

Untuk menggunakan kosakata kustom yang ada dalam pekerjaan transkripsi, atur `VocabularyName` di `Settings`bidang saat Anda memanggil `StartTranscriptionJob`operasi atau, dari AWS Management Console, pilih kosakata khusus dari daftar tarik-turun.

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

Berikut contoh lain menggunakan perintah create-vocabulary, dan badan permintaan yang membuat kosakata kustom Anda.

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

File my-first-vocab-table.json berisi badan permintaan berikut.

```
{
  "VocabularyName": "my-first-vocabulary",
```

```
  "VocabularyFileUri": "s3://amzn-s3-demo-bucket/my-vocabularies/my-vocabulary-
table.txt",
  "LanguageCode": "en-US"
}
```

Setelah `VocabularyState` berubah dari `PENDING` ke`READY`, kosakata khusus Anda siap digunakan dengan transkripsi. Untuk melihat status kosakata kustom Anda saat ini, jalankan:

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

AWS SDK untuk Python (Boto3)

Contoh ini menggunakan AWS SDK untuk Python (Boto3) untuk membuat kosakata kustom dari tabel menggunakan metode create_vocabulary. Untuk informasi selengkapnya, lihat `CreateVocabulary`.

Untuk menggunakan kosakata kustom yang ada dalam pekerjaan transkripsi, atur `VocabularyName` di `Settings`bidang saat Anda memanggil `StartTranscriptionJob`operasi atau, dari AWS Management Console, pilih kosakata khusus dari daftar tarik-turun.

Untuk contoh tambahan menggunakan AWS SDKs, termasuk contoh khusus fitur, skenario, dan lintas layanan, lihat bagian ini. Contoh kode untuk Amazon Transcribe menggunakan AWS SDKs

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

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

> **Note**
>
> Jika Anda membuat Amazon S3 bucket baru untuk file kosakata kustom Anda, pastikan IAM peran yang membuat `CreateVocabulary`permintaan memiliki izin untuk mengakses bucket ini. Jika peran tidak memiliki izin yang benar, permintaan Anda gagal. Anda dapat secara opsional menentukan IAM peran dalam permintaan Anda dengan menyertakan `DataAccessRoleArn` parameter. Untuk informasi selengkapnya tentang IAM peran dan kebijakan di Amazon Transcribe, lihatAmazon Transcribe contoh kebijakan berbasis identitas.

Membuat kosakata khusus menggunakan daftar

> **Important**
>
> Kosakata kustom dalam format daftar tidak digunakan lagi, jadi jika Anda membuat kosakata kustom baru, kami sangat menyarankan menggunakan format tabel.

Anda dapat membuat kosakata kustom dari daftar menggunakan AWS Management Console, AWS CLI, atau. AWS SDKs

- AWS Management Console: Anda harus membuat dan mengunggah file teks yang berisi kosakata khusus Anda. Anda dapat menggunakan entri yang dipisahkan baris atau dipisahkan koma. Perhatikan bahwa daftar Anda harus disimpan sebagai file teks (*.txt) dalam format. `LF` Jika Anda menggunakan format lain, seperti`CRLF`, kosakata khusus Anda tidak diterima oleh. Amazon Transcribe
- AWS CLIdan AWS SDKs: Anda harus menyertakan kosakata kustom Anda sebagai entri yang dipisahkan koma dalam panggilan API Anda menggunakan bendera. `Phrases`

Jika entri berisi beberapa kata, Anda harus memberi tanda hubung setiap kata. Misalnya, Anda memasukkan 'Los Angeles' sebagai **`Los-Angeles`** dan 'Andorra la Vella' sebagai. **`Andorra-la-Vella`**

Berikut adalah contoh dari dua format daftar yang valid. Lihat Membuat daftar kosakata khusus contoh khusus metode.

- Entri yang dipisahkan koma:

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

- Entri yang dipisahkan baris:

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

> **Important**
>
> Anda hanya dapat menggunakan karakter yang didukung untuk bahasa Anda. Lihat set karakter bahasa Anda untuk detailnya.

Daftar kosakata khusus tidak didukung dengan operasi. `CreateMedicalVocabulary` Jika membuat kosakata medis khusus, Anda harus menggunakan format tabel; lihat Membuat kosakata khusus menggunakan tabel untuk instruksi.

Membuat daftar kosakata khusus

Untuk memproses daftar kosakata khusus untuk digunakan Amazon Transcribe, lihat contoh berikut:

AWS CLI

Contoh ini menggunakan perintah create-vocabulary dengan file kosakata kustom yang diformat daftar. Untuk informasi selengkapnya, lihat `CreateVocabulary`.

```
aws transcribe create-vocabulary \
--vocabulary-name my-first-vocabulary \
--language-code en-US \
--phrases {CLI,Eva-Maria,ABCs}
```

Berikut contoh lain menggunakan perintah create-vocabulary, dan badan permintaan yang membuat kosakata kustom Anda.

```
aws transcribe create-vocabulary \
```

```
--cli-input-json file://filepath/my-first-vocab-list.json
```

File my-first-vocab-list.json berisi badan permintaan berikut.

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

Setelah `VocabularyState` berubah dari `PENDING` ke`READY`, kosakata khusus Anda siap digunakan dengan transkripsi. Untuk melihat status kosakata kustom Anda saat ini, jalankan:

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

AWS SDK untuk Python (Boto3)

Contoh ini menggunakan AWS SDK untuk Python (Boto3) untuk membuat kosakata kustom dari daftar menggunakan metode create_vocabulary. Untuk informasi selengkapnya, lihat `CreateVocabulary`.

Untuk contoh tambahan menggunakan AWS SDKs, termasuk contoh khusus fitur, skenario, dan lintas layanan, lihat bagian ini. Contoh kode untuk Amazon Transcribe menggunakan AWS SDKs

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

while True:
```

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

> **Note**
>
> Jika Anda membuat Amazon S3 bucket baru untuk file kosakata kustom Anda, pastikan IAM peran yang membuat `CreateVocabulary`permintaan memiliki izin untuk mengakses bucket ini. Jika peran tidak memiliki izin yang benar, permintaan Anda gagal. Anda dapat secara opsional menentukan IAM peran dalam permintaan Anda dengan menyertakan `DataAccessRoleArn` parameter. Untuk informasi selengkapnya tentang IAM peran dan kebijakan di Amazon Transcribe, lihatAmazon Transcribe contoh kebijakan berbasis identitas.

Menggunakan kosakata khusus

Setelah kosakata kustom Anda dibuat, Anda dapat memasukkannya ke dalam permintaan transkripsi Anda; lihat bagian berikut untuk contoh.

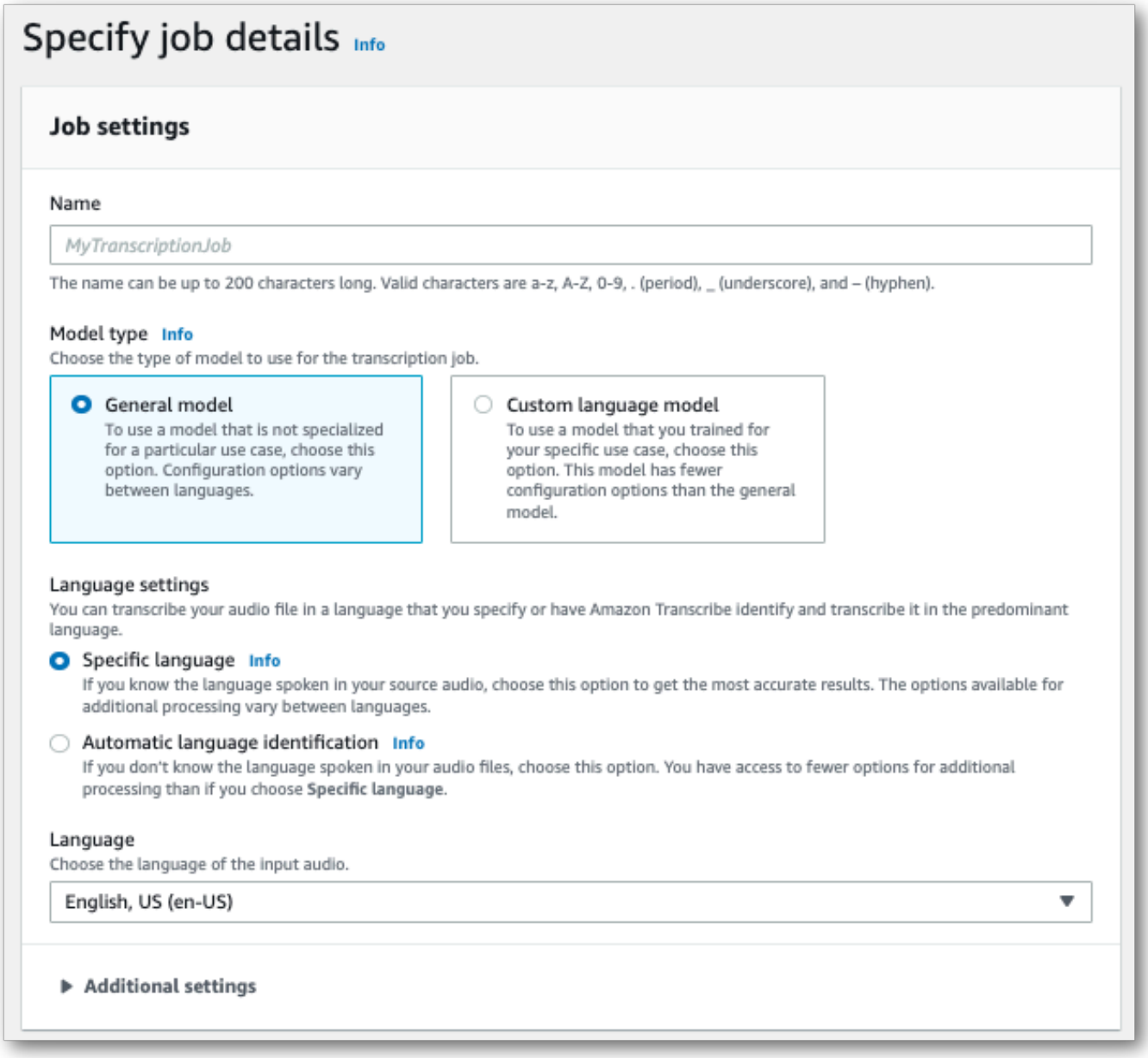
Bahasa kosakata kustom yang Anda sertakan dalam permintaan Anda harus cocok dengan kode bahasa yang Anda tentukan untuk media Anda. Jika bahasa tidak cocok, kosakata khusus Anda tidak diterapkan pada transkripsi Anda dan tidak ada peringatan atau kesalahan.

Menggunakan kosakata khusus dalam transkripsi batch

Untuk menggunakan kosakata khusus dengan transkripsi batch, lihat contoh berikut ini:

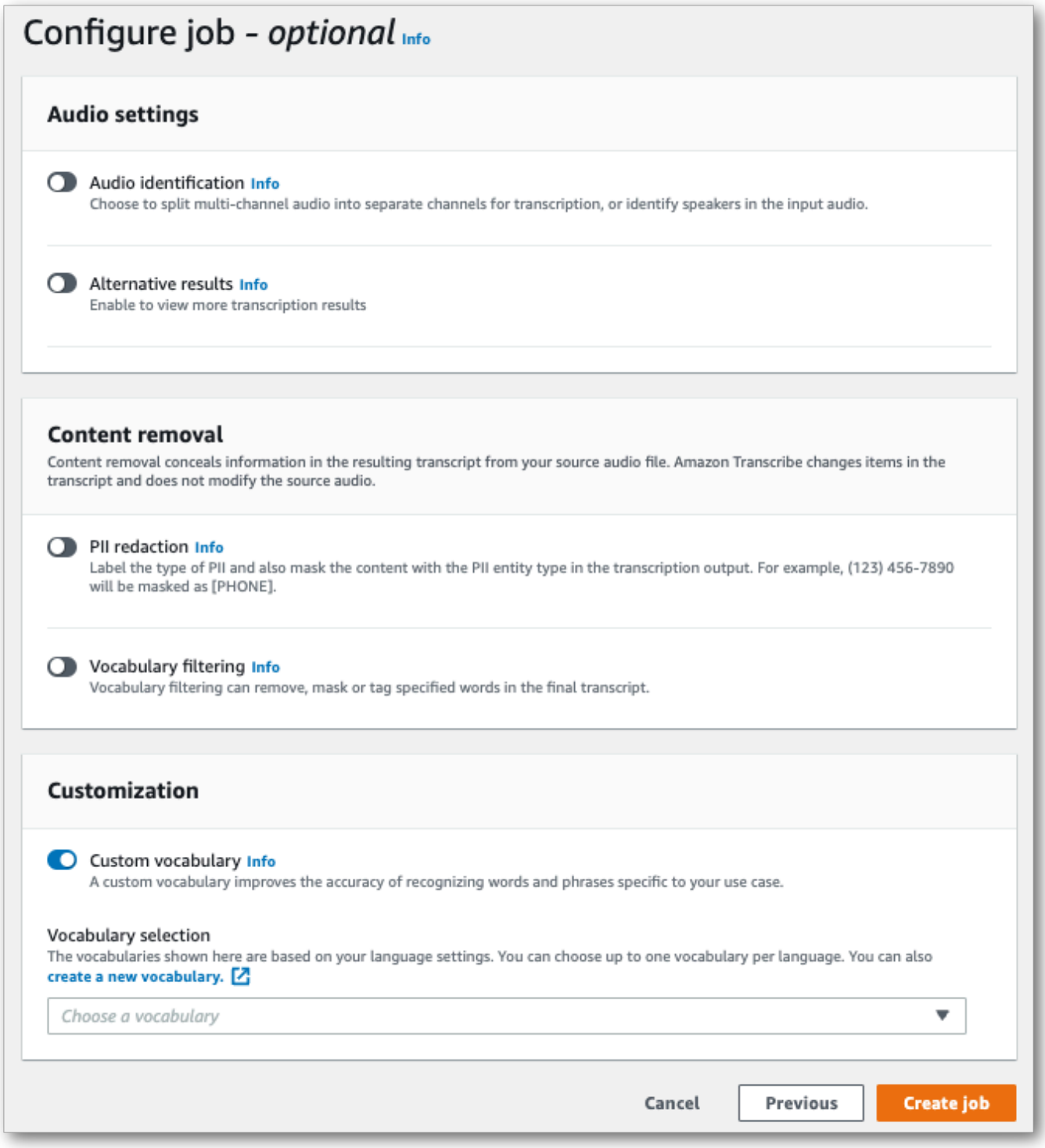
AWS Management Console

1. Masuk ke AWS Management Console.
2. Di panel navigasi, pilih Pekerjaan transkripsi, lalu pilih Buat pekerjaan (kanan atas). Ini membuka halaman Tentukan detail pekerjaan.



Beri nama pekerjaan Anda dan tentukan media input Anda. Secara opsional sertakan bidang lain, lalu pilih Berikutnya.

3. Di bagian bawah halaman Configure job, di panel Kustomisasi, aktifkan kosakata kustom.



4. Pilih kosakata kustom Anda dari menu dropdown.

   Pilih Buat pekerjaan untuk menjalankan pekerjaan transkripsi Anda.

AWS CLI

Contoh ini menggunakan start-transcription-jobperintah dan `Settings` parameter dengan `VocabularyName` sub-parameter. Untuk informasi selengkapnya, silakan lihat `StartTranscriptionJob` dan `Settings`.

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

Berikut contoh lain menggunakan start-transcription-jobperintah, dan badan permintaan yang menyertakan kosakata kustom Anda dengan pekerjaan itu.

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

File my-first-vocabulary-job.json berisi badan permintaan berikut.

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

AWS SDK untuk Python (Boto3)

Contoh ini menggunakan AWS SDK untuk Python (Boto3) untuk menyertakan kosakata khusus menggunakan `Settings` argumen untuk metode start_transcription_job. Untuk informasi selengkapnya, silakan lihat `StartTranscriptionJob` dan `Settings`.

Untuk contoh tambahan menggunakan AWS SDKs, termasuk contoh khusus fitur, skenario, dan lintas layanan, lihat bagian ini. Contoh kode untuk Amazon Transcribe menggunakan AWS SDKs

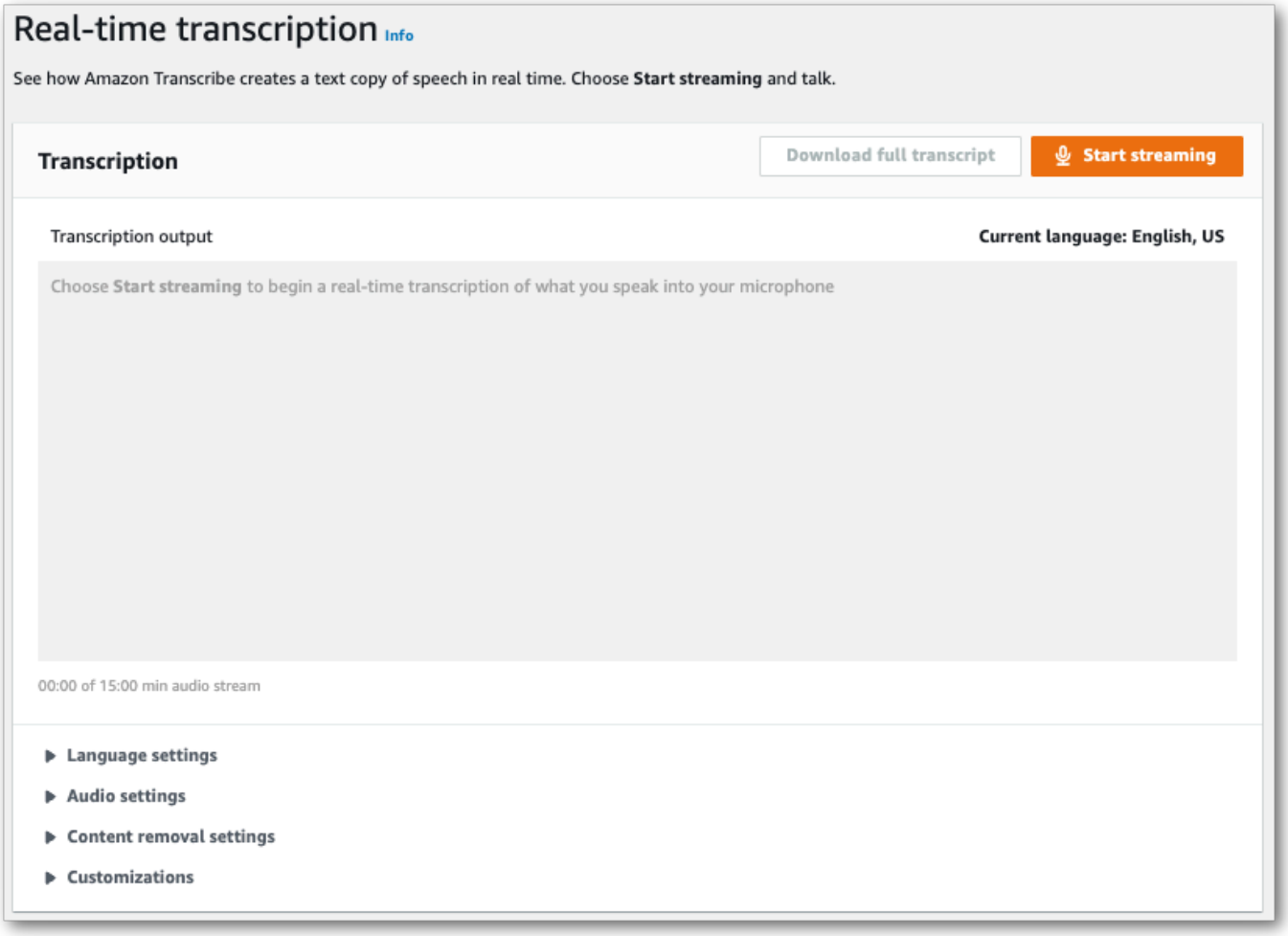
```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
job_name = "my-first-transcription-job"
job_uri = "s3://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
transcribe.start_transcription_job(
    TranscriptionJobName = job_name,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

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

Menggunakan kosakata khusus dalam transkripsi streaming

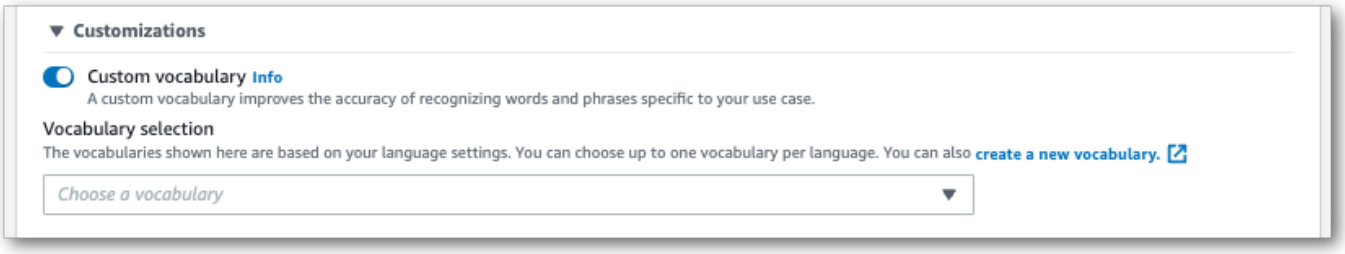
Untuk menggunakan kosakata khusus dengan transkripsi streaming, lihat contoh berikut ini:

AWS Management Console

1. Masuk ke AWS Management Console.
2. Di panel navigasi, pilih Transkripsi waktu nyata. Gulir ke bawah ke Kustomisasi dan perluas bidang ini jika diminimalkan.



3. Aktifkan kosakata kustom dan pilih kosakata kustom dari menu tarik-turun.



   Sertakan pengaturan lain yang ingin Anda terapkan ke streaming Anda.

4. Anda sekarang siap untuk mentranskripsikan aliran Anda. Pilih Mulai streaming dan mulai berbicara. Untuk mengakhiri dikte Anda, pilih Hentikan streaming.

Aliran HTTP/2

Contoh ini membuat permintaan HTTP/2 yang menyertakan kosakata kustom Anda. Untuk informasi lebih lanjut tentang menggunakan streaming HTTP/2 dengan Amazon Transcribe, lihat. Menyiapkan aliran HTTP/2 Untuk detail selengkapnya tentang parameter dan header khusus untuk Amazon Transcribe, lihat `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
```

Definisi parameter dapat ditemukan di Referensi API; parameter yang umum untuk semua operasi AWS API tercantum di bagian Parameter Umum.

WebSocket aliran

Contoh ini membuat URL presigned yang menerapkan kosakata kustom Anda ke stream. WebSocket Jeda baris telah ditambahkan untuk keterbacaan. Untuk informasi selengkapnya tentang penggunaan WebSocket stream dengan Amazon Transcribe, lihatMenyiapkan WebSocket aliran. Untuk detail lebih lanjut tentang parameter, lihat `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
```

Definisi parameter dapat ditemukan di Referensi API; parameter yang umum untuk semua operasi AWS API tercantum di bagian Parameter Umum.

Model bahasa khusus

Model bahasa khusus dirancang untuk meningkatkan akurasi transkripsi untuk pidato khusus domain. Ini termasuk konten apa pun di luar apa yang akan Anda dengar dalam percakapan sehari-hari yang normal. Misalnya, jika Anda menyalin prosiding dari konferensi ilmiah, transkripsi standar tidak mungkin mengenali banyak istilah ilmiah yang digunakan oleh presenter. Dalam hal ini, Anda dapat melatih model bahasa khusus untuk mengenali istilah khusus yang digunakan dalam disiplin Anda.

Tidak seperti kosakata khusus, yang meningkatkan pengenalan kata dengan memberikan petunjuk (seperti pengucapan), model bahasa khusus mempelajari konteks yang terkait dengan kata tertentu. Ini termasuk bagaimana dan kapan sebuah kata digunakan, dan hubungan sebuah kata dengan kata lain. Misalnya, jika Anda melatih model Anda menggunakan makalah penelitian ilmu iklim, model Anda mungkin belajar bahwa 'gumpalan es' adalah pasangan kata yang lebih mungkin daripada 'aliran es'.

Untuk melihat bahasa yang didukung untuk model bahasa kustom, lihatBahasa yang didukung dan fitur khusus bahasa. Perhatikan bahwa jika Anda menyertakan model bahasa kustom dalam permintaan Anda, Anda tidak dapat mengaktifkan identifikasi bahasa (Anda harus menentukan kode bahasa).

> Operasi API khusus untuk model bahasa khusus
>
> `CreateLanguageModel`, `DeleteLanguageModel`, `DescribeLanguageModel`, `ListLanguageModels`

Sumber data

Anda dapat menggunakan semua jenis data teks yang ingin Anda latih model Anda. Namun, semakin dekat konten teks Anda dengan konten audio Anda, semakin akurat model Anda. Oleh karena itu,

penting untuk memilih data teks yang menggunakan istilah yang sama dalam konteks yang sama dengan audio Anda.

Data terbaik untuk melatih model adalah transkrip yang akurat. Ini dianggap sebagai data dalam domain. Data teks dalam domain memiliki istilah, penggunaan, dan konteks yang sama persis dengan audio yang ingin Anda transkripsikan.

Jika Anda tidak memiliki transkrip yang akurat, gunakan artikel jurnal, laporan teknis, whitepaper, prosiding konferensi, instruksi manual, artikel berita, konten situs web, dan teks lain yang berisi istilah yang diinginkan yang digunakan dalam konteks yang mirip dengan audio Anda. Ini dianggap sebagai data terkait domain.

Membuat model bahasa kustom yang kuat mungkin memerlukan sejumlah besar data teks, yang harus berisi istilah yang diucapkan dalam audio Anda. Anda dapat Amazon Transcribe menyediakan data teks hingga 2 GB untuk melatih model Anda—ini disebut sebagai data pelatihan. Secara opsional, jika Anda tidak memiliki (atau sedikit) transkrip dalam domain, Anda dapat menyediakan Amazon Transcribe hingga 200 MB data teks untuk menyetel model Anda—ini disebut sebagai data penyetelan.

Pelatihan versus penyetelan data

Tujuan dari data pelatihan adalah Amazon Transcribe untuk mengajarkan mengenali istilah-istilah baru dan mempelajari konteks di mana istilah-istilah ini digunakan. Untuk membuat model yang kuat, Amazon Transcribe mungkin memerlukan volume besar data teks yang relevan. Menyediakan data pelatihan sebanyak mungkin, hingga batas 2 GB, sangat disarankan.

Tujuan penyetelan data adalah untuk membantu menyempurnakan dan mengoptimalkan hubungan kontekstual yang dipelajari dari data pelatihan Anda. Data penyetelan tidak diperlukan untuk membuat model bahasa khusus.

Terserah Anda untuk memutuskan cara terbaik untuk memilih pelatihan dan, secara opsional, menyetel data. Setiap kasus unik dan tergantung pada jenis dan jumlah data yang Anda miliki. Data penyetelan disarankan saat Anda kekurangan data pelatihan dalam domain.

Jika Anda memilih untuk menyertakan kedua tipe data, jangan tumpang tindih dengan data pelatihan dan penyetelan Anda; data pelatihan dan penyetelan harus unik. Data yang tumpang tindih dapat membuat bias dan memiringkan model bahasa kustom Anda, memengaruhi keakuratannya.

Sebagai panduan umum, sebaiknya gunakan teks dalam domain yang akurat sebagai data pelatihan bila memungkinkan. Berikut adalah beberapa skenario umum, tercantum dalam urutan preferensi:

- Jika Anda memiliki lebih dari 10.000 kata teks transkrip dalam domain yang akurat, gunakan itu sebagai data pelatihan. Dalam hal ini, tidak perlu memasukkan data tuning. Ini adalah skenario ideal untuk melatih model bahasa khusus.
- Jika Anda memiliki teks transkrip dalam domain yang akurat yang berisi kurang dari 10.000 kata dan tidak mendapatkan hasil yang diinginkan, pertimbangkan untuk menambah data pelatihan Anda dengan teks tertulis terkait domain, seperti laporan teknis. Dalam hal ini, pesan sebagian kecil (10-25%) dari data transkrip dalam domain Anda untuk digunakan sebagai data penyetelan.
- Jika Anda tidak memiliki teks transkrip dalam domain, unggah semua teks terkait domain Anda sebagai data pelatihan. Dalam hal ini, teks bergaya transkrip lebih disukai daripada teks tertulis. Ini adalah skenario yang paling tidak efektif untuk melatih model bahasa khusus.

Ketika Anda siap untuk membuat model Anda, lihatMembuat model bahasa khusus.

Membuat model bahasa khusus

Sebelum Anda dapat membuat model bahasa kustom Anda, Anda harus:

- Siapkan data Anda. Data harus disimpan dalam format teks biasa dan tidak dapat berisi karakter khusus apa pun.
- Unggah data Anda ke dalam Amazon S3 ember. Disarankan untuk membuat folder terpisah untuk pelatihan dan penyetelan data.
- Pastikan Anda Amazon Transcribe memiliki akses ke Amazon S3 bucket Anda. Anda harus menentukan IAM peran yang memiliki izin akses untuk menggunakan data Anda.

Mempersiapkan data Anda

Anda dapat mengkompilasi semua data Anda dalam satu file atau menyimpannya sebagai beberapa file. Perhatikan bahwa jika Anda memilih untuk menyertakan data penyetelan, data tersebut harus disimpan dalam file terpisah dari data pelatihan Anda.

Tidak masalah berapa banyak file teks yang Anda gunakan untuk pelatihan atau penyetelan data Anda. Mengunggah satu file dengan 100.000 kata menghasilkan hasil yang sama dengan mengunggah 10 file dengan 10.000 kata. Siapkan data teks Anda dengan cara yang paling nyaman bagi Anda.

Pastikan semua file data Anda memenuhi kriteria berikut:

- Mereka semua dalam bahasa yang sama dengan model yang ingin Anda buat. Misalnya, jika Anda ingin membuat model bahasa khusus yang mentranskripsikan audio dalam bahasa Inggris AS (`en-US`), semua data teks Anda harus dalam bahasa Inggris AS.
- Mereka dalam format teks biasa dengan pengkodean UTF-8.
- Mereka tidak mengandung karakter khusus atau format, seperti tag HTML.
- Mereka berjumlah total gabungan maksimum 2 GB untuk data pelatihan dan 200 MB untuk menyetel data.

Jika salah satu kriteria ini tidak terpenuhi, model Anda gagal.

Mengunggah data Anda

Sebelum mengunggah data Anda, buat folder baru untuk data pelatihan Anda. Jika menggunakan data penyetelan, buat folder terpisah lainnya.

URIs Untuk ember Anda mungkin terlihat seperti:

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

Unggah data pelatihan dan tuning Anda ke dalam bucket yang sesuai.

Anda dapat menambahkan lebih banyak data ke bucket ini di kemudian hari. Namun, jika Anda melakukannya, Anda perlu membuat ulang model Anda dengan data baru. Model yang ada tidak dapat diperbarui dengan data baru.

Mengizinkan akses ke data Anda

Untuk membuat model bahasa kustom, Anda harus menentukan IAM peran yang memiliki izin untuk mengakses Amazon S3 bucket Anda. Jika Anda belum memiliki peran dengan akses ke Amazon S3 bucket tempat Anda menempatkan data pelatihan, Anda harus membuatnya. Setelah membuat peran, Anda dapat melampirkan kebijakan untuk memberikan izin peran tersebut. Jangan melampirkan kebijakan ke pengguna.

Untuk kebijakan-kebijakan contoh, lihat Amazon Transcribe contoh kebijakan berbasis identitas.

Untuk mempelajari cara membuat IAM identitas baru, lihat IAM Identitas (pengguna, grup pengguna, dan peran).

Untuk mempelajari lebih lanjut tentang kebijakan, lihat:

- Kebijakan dan izin di IAM
- Membuat IAM kebijakan
- Manajemen akses untuk sumber daya AWS

Membuat model bahasa kustom Anda

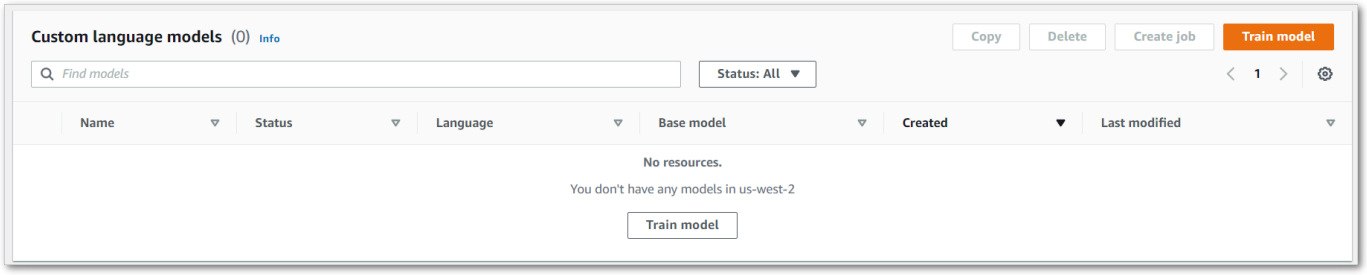
Saat membuat model bahasa kustom Anda, Anda harus memilih model dasar. Ada dua opsi model dasar:

- `NarrowBand`: Gunakan opsi ini untuk audio dengan laju sampel kurang dari 16.000 Hz. Jenis model ini biasanya digunakan untuk percakapan telepon yang direkam pada 8.000 Hz.
- `WideBand`: Gunakan opsi ini untuk audio dengan laju sampel lebih besar dari atau sama dengan 16.000 Hz.

Anda dapat membuat model bahasa kustom menggunakan AWS Management Console AWS CLI,, atau AWS SDKs.; lihat contoh berikut:

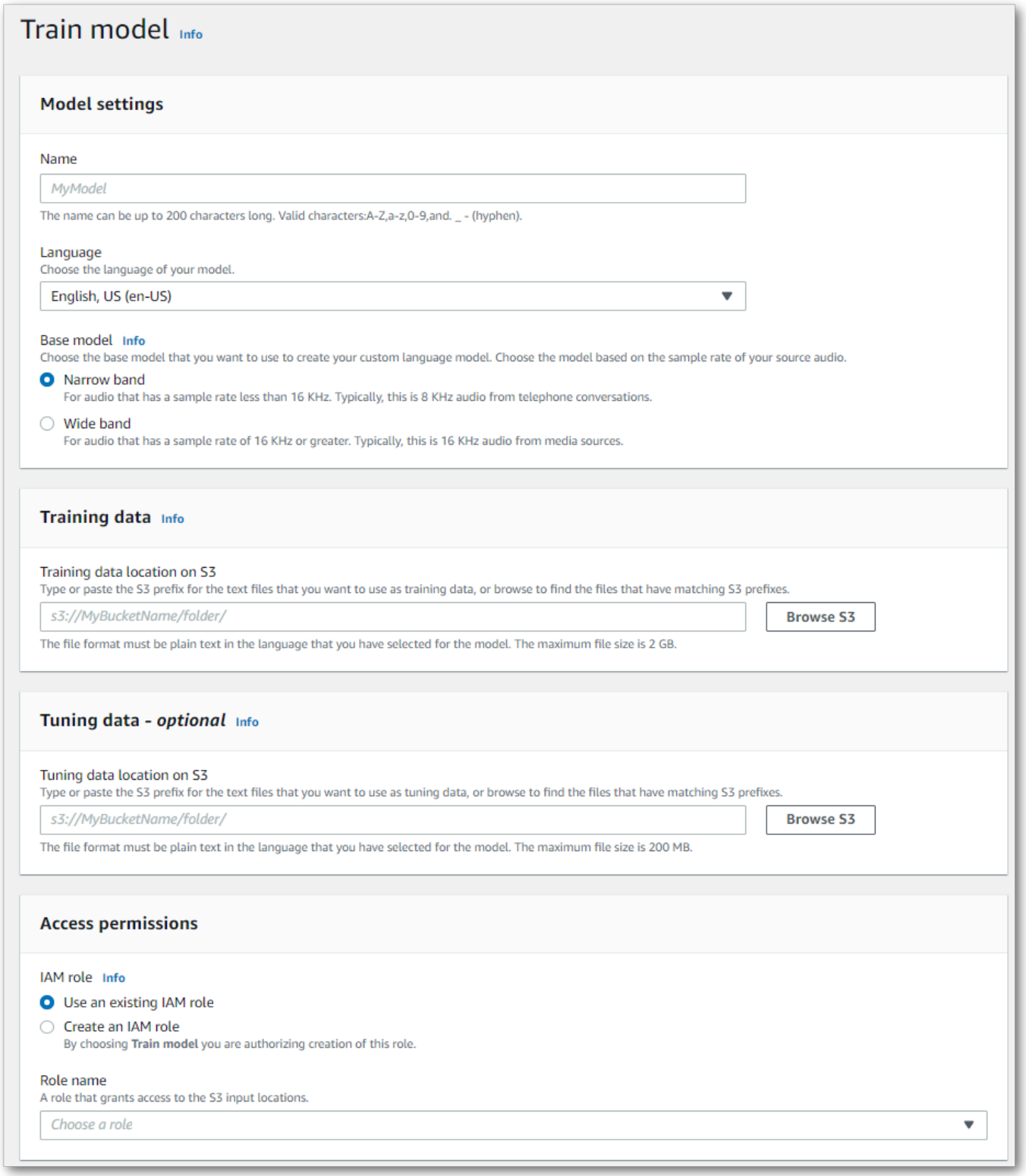
AWS Management Console

1. Masuk ke AWS Management Console.
2. Di panel navigasi, pilih Model bahasa kustom. Ini membuka halaman model bahasa kustom di mana Anda dapat melihat model bahasa kustom yang ada atau melatih model bahasa kustom baru.
3. Untuk melatih model baru, pilih Model kereta.



   Ini membawa Anda ke halaman model Kereta. Tambahkan nama, tentukan bahasa, dan pilih model dasar yang Anda inginkan untuk model Anda. Kemudian, tambahkan jalur ke pelatihan

Anda dan, secara opsional, data penyetelan Anda. Anda harus menyertakan IAM peran yang memiliki izin untuk mengakses data Anda.



4. Setelah semua bidang selesai, pilih Model kereta api di bagian bawah halaman.

AWS CLI

Contoh ini menggunakan create-language-modelperintah. Untuk informasi selengkapnya, silakan lihat `CreateLanguageModel` dan `LanguageModel`.

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

Berikut contoh lain menggunakan create-language-modelperintah, dan badan permintaan yang membuat model bahasa kustom Anda.

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

File my-first-language-model.json berisi badan permintaan berikut.

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

AWS SDK untuk Python (Boto3)

Contoh ini menggunakan AWS SDK untuk Python (Boto3) untuk membuat CLM menggunakan metode create_language_model. Untuk informasi selengkapnya, silakan lihat `CreateLanguageModel` dan `LanguageModel`.

Untuk contoh tambahan menggunakan AWS SDKs, termasuk contoh khusus fitur, skenario, dan lintas layanan, lihat bagian ini. Contoh kode untuk Amazon Transcribe menggunakan AWS SDKs

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

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

Memperbarui model bahasa kustom Anda

Amazon Transcribe terus memperbarui model dasar yang tersedia untuk model bahasa khusus. Untuk mendapatkan manfaat dari pembaruan ini, kami merekomendasikan pelatihan model bahasa kustom baru setiap 6 hingga 12 bulan.

Untuk melihat apakah model bahasa kustom Anda menggunakan model dasar terbaru, jalankan `DescribeLanguageModel`permintaan menggunakan AWS CLI atau AWS SDK, lalu temukan `UpgradeAvailability` bidang dalam respons Anda.

Jika `UpgradeAvailability` ya`true`, model Anda tidak menjalankan versi terbaru dari model dasar. Untuk menggunakan model dasar terbaru dalam model bahasa kustom, Anda harus membuat model bahasa kustom baru. Model bahasa khusus tidak dapat ditingkatkan.

Menggunakan model bahasa khusus

Setelah Anda membuat model bahasa kustom Anda, Anda dapat memasukkannya ke dalam permintaan transkripsi Anda; lihat bagian berikut untuk contoh.

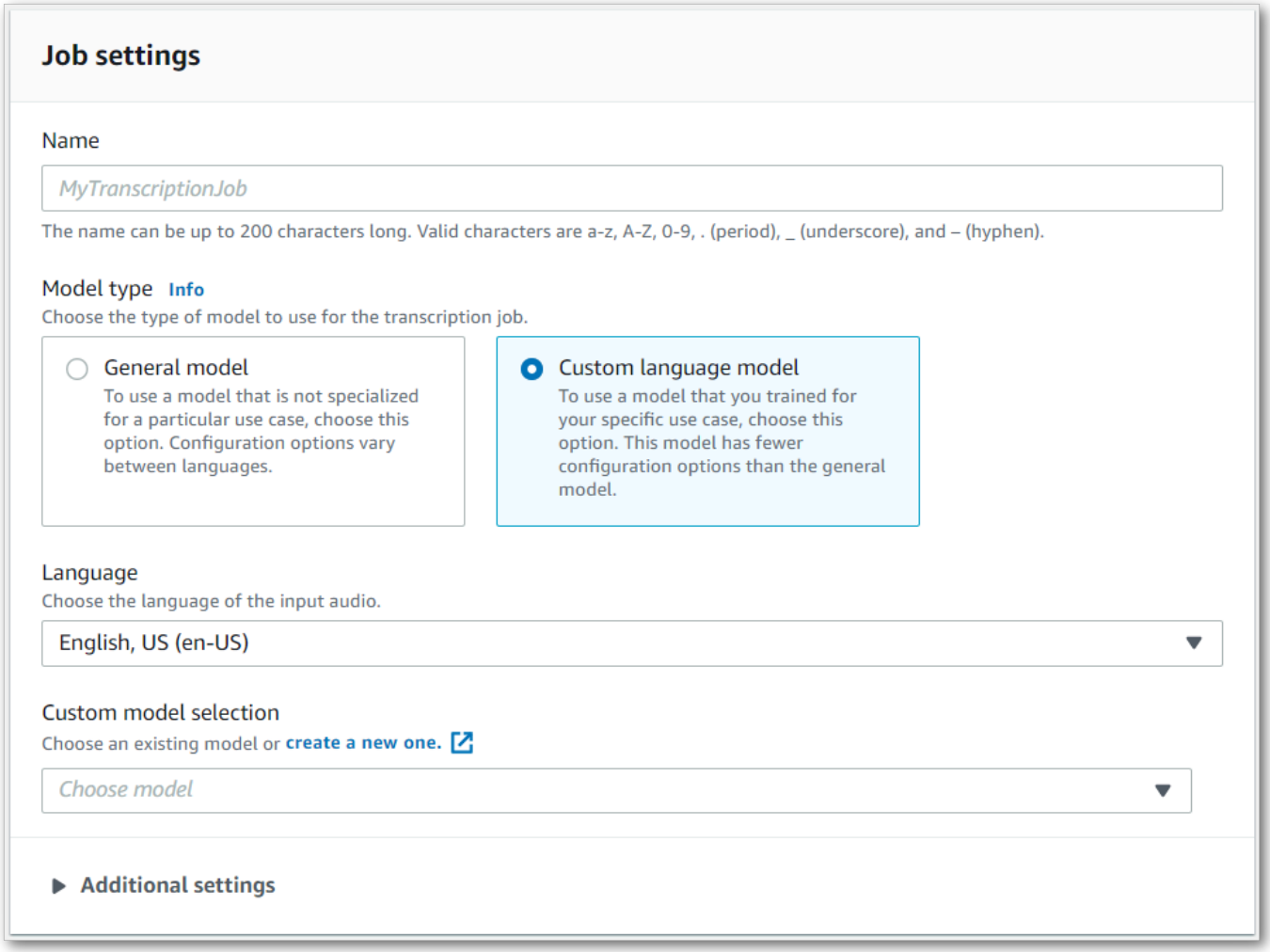
Bahasa model yang Anda sertakan dalam permintaan harus sesuai dengan kode bahasa yang Anda tentukan untuk media Anda. Jika bahasa tidak cocok, model bahasa kustom Anda tidak diterapkan ke transkripsi Anda dan tidak ada peringatan atau kesalahan.

Menggunakan model bahasa khusus dalam transkripsi batch

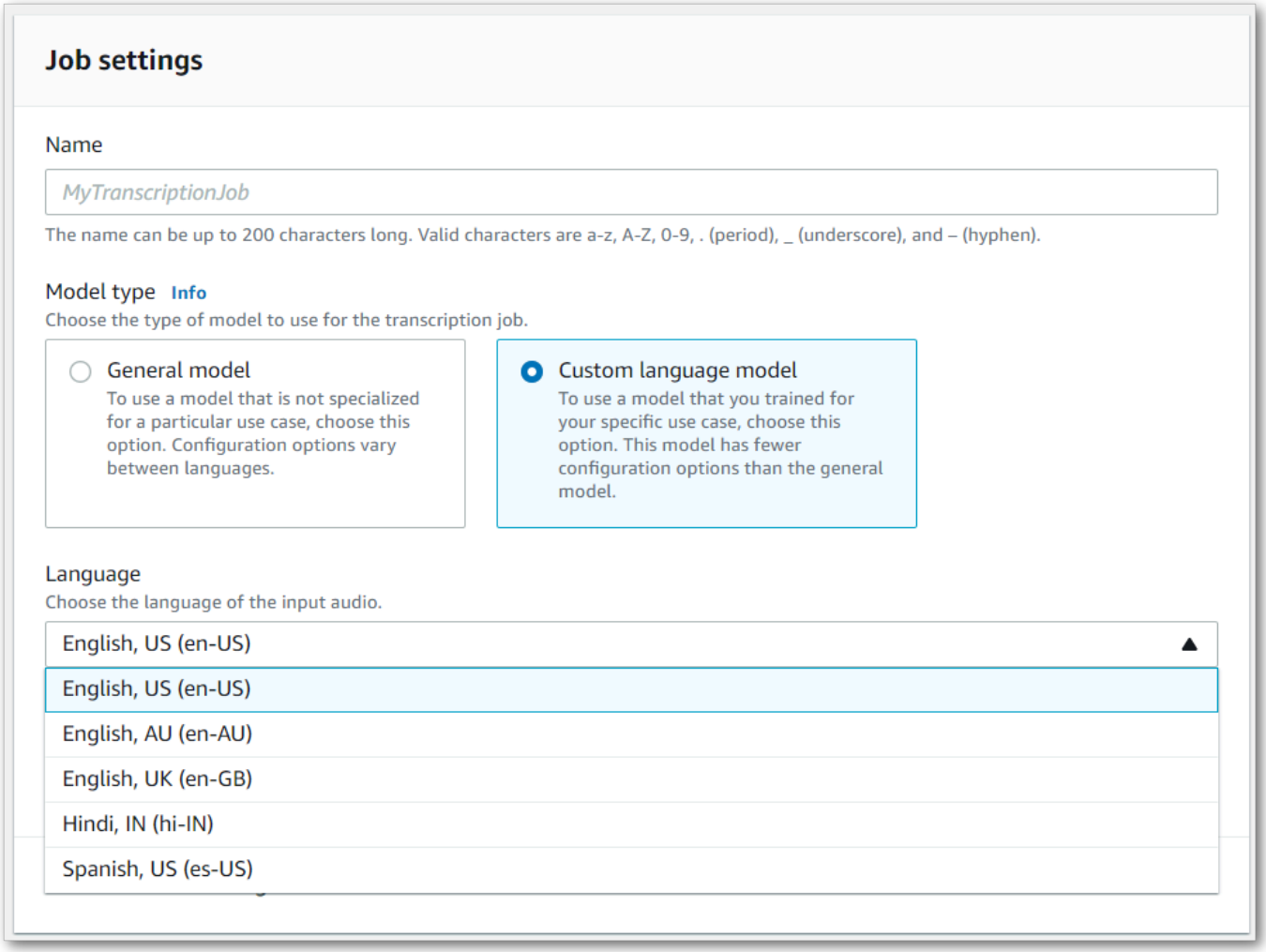
Untuk menggunakan model bahasa khusus dengan transkripsi batch, lihat contoh berikut:

AWS Management Console

1. Masuk ke AWS Management Console.
2. Di panel navigasi, pilih Pekerjaan transkripsi, lalu pilih Buat pekerjaan (kanan atas). Ini membuka halaman Tentukan detail pekerjaan.
3. Di panel Pengaturan Job di bawah Jenis model, pilih kotak Model bahasa kustom.



Anda juga harus memilih bahasa input dari menu dropdown.



4. Di bawah Pemilihan model kustom, pilih model bahasa kustom yang ada dari menu tarik-turun atau Buat yang baru.

   Tambahkan Amazon S3 lokasi file input Anda di panel data Input.

5. Pilih Berikutnya untuk opsi konfigurasi tambahan.

   Pilih Buat pekerjaan untuk menjalankan pekerjaan transkripsi Anda.

AWS CLI

Contoh ini menggunakan start-transcription-jobperintah dan `ModelSettings` parameter dengan `VocabularyName` sub-parameter. Untuk informasi selengkapnya, silakan lihat `StartTranscriptionJob` dan `ModelSettings`.

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

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

Berikut contoh lain menggunakan start-transcription-jobperintah, dan badan permintaan yang menyertakan model bahasa kustom Anda dengan pekerjaan itu.

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

File my-first-model-job.json berisi badan permintaan berikut.

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

AWS SDK untuk Python (Boto3)

Contoh ini menggunakan AWS SDK untuk Python (Boto3) untuk menyertakan model bahasa kustom menggunakan `ModelSettings` argumen untuk metode start_transcription_job. Untuk informasi selengkapnya, silakan lihat `StartTranscriptionJob` dan `ModelSettings`.

Untuk contoh tambahan menggunakan AWS SDKs, termasuk contoh khusus fitur, skenario, dan lintas layanan, lihat bagian ini. Contoh kode untuk Amazon Transcribe menggunakan AWS SDKs

```
from __future__ import print_function
import time
```

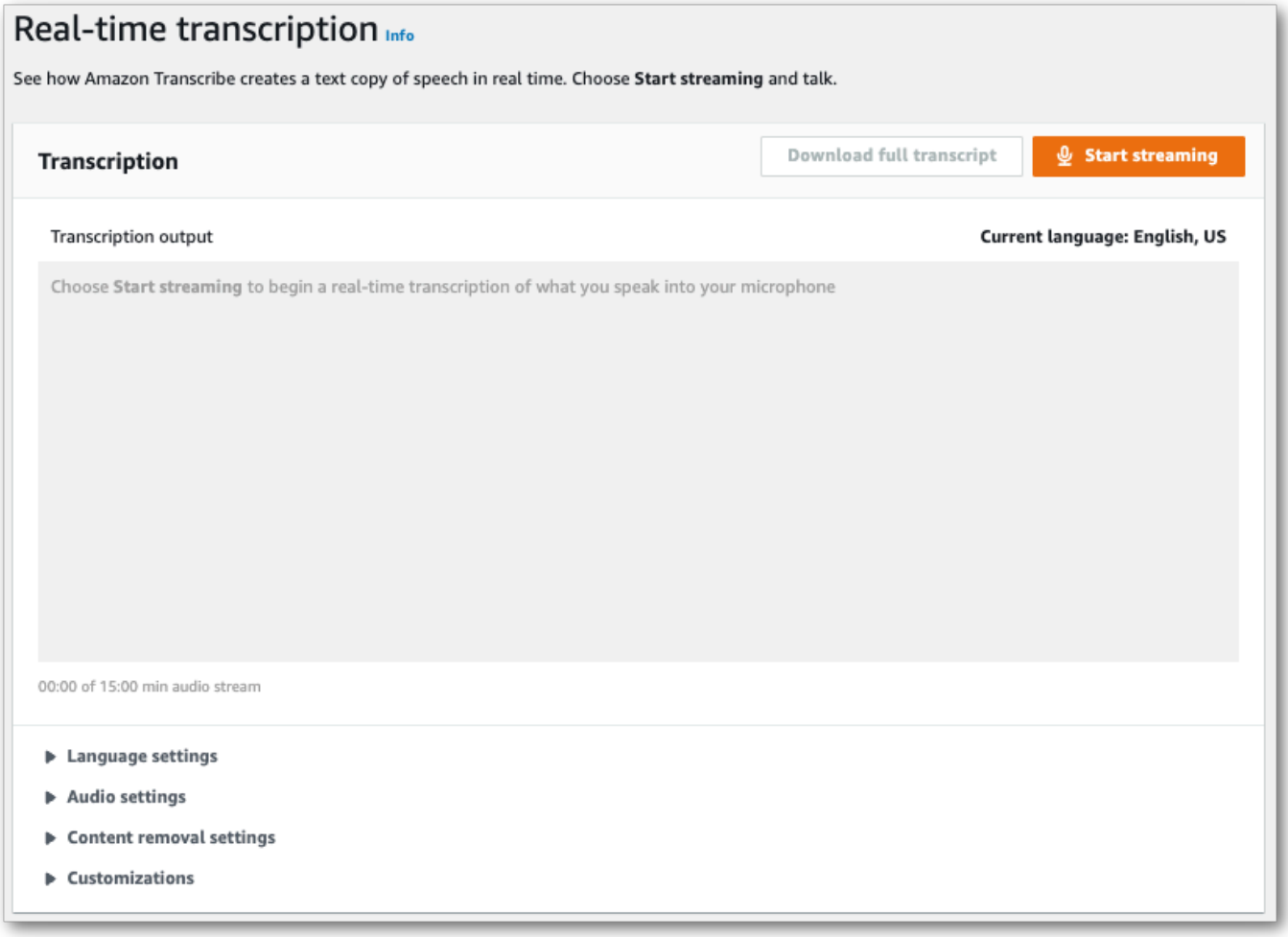
```
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
job_name = "my-first-transcription-job"
job_uri = "s3://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
transcribe.start_transcription_job(
    TranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
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

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

Menggunakan model bahasa khusus dalam transkripsi streaming

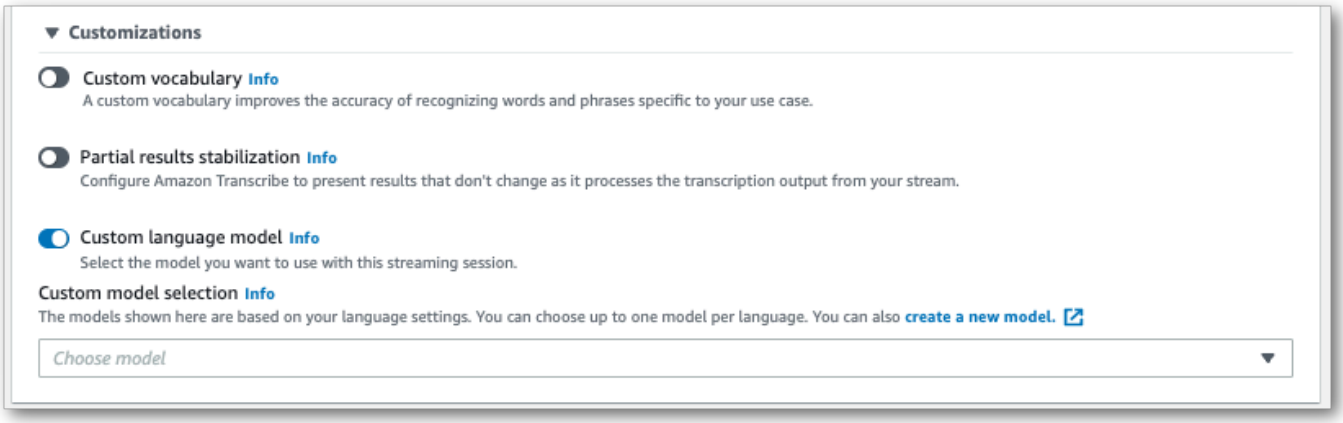
Untuk menggunakan model bahasa khusus dengan transkripsi streaming, lihat contoh berikut ini:

AWS Management Console

1. Masuk ke AWS Management Console.
2. Di panel navigasi, pilih Transkripsi waktu nyata. Gulir ke bawah ke Kustomisasi dan perluas bidang ini jika diminimalkan.



3. Beralih pada model bahasa Kustom dan pilih model dari menu tarik-turun.



   Sertakan pengaturan lain yang ingin Anda terapkan ke streaming Anda.

4. Anda sekarang siap untuk mentranskripsikan aliran Anda. Pilih Mulai streaming dan mulai berbicara. Untuk mengakhiri dikte Anda, pilih Hentikan streaming.

Aliran HTTP/2

Contoh ini membuat permintaan HTTP/2 yang menyertakan model bahasa kustom Anda. Untuk informasi lebih lanjut tentang menggunakan streaming HTTP/2 dengan Amazon Transcribe, lihat. Menyiapkan aliran HTTP/2 Untuk detail selengkapnya tentang parameter dan header khusus untuk Amazon Transcribe, lihat 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
```

Definisi parameter dapat ditemukan di Referensi API; parameter yang umum untuk semua operasi AWS API tercantum di bagian Parameter Umum.

WebSocket aliran

Contoh ini membuat URL presigned yang menerapkan model bahasa kustom Anda ke WebSocket stream. Jeda baris telah ditambahkan untuk keterbacaan. Untuk informasi selengkapnya tentang penggunaan WebSocket stream dengan Amazon Transcribe, lihatMenyiapkan WebSocket aliran. Untuk detail lebih lanjut tentang parameter, lihat 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
```

Definisi parameter dapat ditemukan di Referensi API; parameter yang umum untuk semua operasi AWS API tercantum di bagian Parameter Umum.

Menggunakan filter kosakata khusus untuk menghapus, menutupi, atau menandai kata-kata

Filter kosakata khusus adalah file teks yang berisi daftar kustom kata-kata individual yang ingin Anda modifikasi dalam output transkripsi Anda.

Kasus penggunaan yang umum adalah penghapusan istilah ofensif atau profan; Namun, filter kosakata khusus sepenuhnya khusus, sehingga Anda dapat memilih kata apa pun yang Anda inginkan. Misalnya, jika Anda memiliki produk baru yang akan diluncurkan, Anda dapat menutupi nama produk dalam transkrip rapat. Dalam hal ini, Anda menjaga pemangku kepentingan up-to-date sambil merahasiakan nama produk hingga diluncurkan.

Pemfilteran kosakata memiliki tiga metode tampilan:`mask`,, `remove` dan. `tag` Lihat contoh berikut untuk melihat cara kerja masing-masing.

- Topeng: Mengganti kata-kata tertentu dengan tiga tanda bintang (***).

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

- Hapus: Menghapus kata-kata yang ditentukan, tidak meninggalkan apa pun di tempatnya.

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

- Tag: Menambahkan tag (`"vocabularyFilterMatch": true`) ke setiap kata yang ditentukan, tetapi tidak mengubah kata itu sendiri. Penandaan memungkinkan penggantian dan pengeditan transkrip yang cepat.

```
"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
```

Saat Anda mengirimkan permintaan transkripsi, Anda dapat menentukan filter kosakata khusus dan metode filtrasi yang ingin Anda terapkan. Amazon Transcribe kemudian memodifikasi kecocokan kata yang tepat ketika mereka muncul di transkrip Anda, sesuai dengan metode filtrasi yang Anda tentukan.

Filter kosakata khusus dapat diterapkan pada permintaan transkripsi batch dan streaming. Untuk mempelajari cara membuat filter kosakata khusus, lihat. Membuat filter kosakata Untuk mempelajari cara menerapkan filter kosakata kustom Anda, lihat. Menggunakan filter kosakata khusus

> **Note**
>
> Amazon Transcribe secara otomatis menutupi istilah sensitif rasial, meskipun Anda dapat memilih keluar dari filter default ini dengan menghubungi AWS Dukungan Teknis.

Untuk panduan video tentang pemfilteran kosakata, lihat Menggunakan filter kosakata.

> **Operasi API khusus untuk pemfilteran kosakata**
>
> `CreateVocabularyFilter`, `DeleteVocabularyFilter`, `GetVocabularyFilter`, `ListVocabularyFilters`, `UpdateVocabularyFilter`

Membuat filter kosakata

Ada dua opsi untuk membuat filter kosakata khusus:

1. Simpan daftar kata-kata yang dipisahkan baris sebagai file teks biasa dengan pengkodean UTF-8.
   - Anda dapat menggunakan pendekatan ini dengan AWS Management Console, AWS CLI, atau AWS SDKs.
   - Jika menggunakan AWS Management Console, Anda dapat memberikan jalur lokal atau Amazon S3 URI untuk file kosakata kustom Anda.
   - Jika menggunakan AWS CLI or AWS SDKs, Anda harus mengunggah file kosakata kustom Anda ke Amazon S3 bucket dan menyertakan Amazon S3 URI dalam permintaan Anda.
2. Sertakan daftar kata yang dipisahkan koma secara langsung dalam permintaan API Anda.

- Anda dapat menggunakan pendekatan ini dengan AWS CLI atau AWS SDKs menggunakan `Words`parameter.

Untuk contoh setiap metode, lihat Membuat filter kosakata khusus

Hal-hal yang perlu diperhatikan saat membuat filter kosakata khusus Anda:

- Kata-kata tidak peka huruf besar/kecil. Misalnya, “kutukan” dan “KUTUKAN” diperlakukan sama.
- Hanya kecocokan kata yang tepat yang difilter. Misalnya, jika filter Anda menyertakan “sumpah” tetapi media Anda berisi kata “sumpah” atau “sumpah”, ini tidak disaring. Hanya contoh “sumpah” yang disaring. Karena itu Anda harus menyertakan semua variasi kata yang ingin Anda filter.
- Filter tidak berlaku untuk kata-kata yang terkandung dalam kata lain. Misalnya, jika filter kosakata khusus berisi “laut” tetapi bukan “kapal selam”, “kapal selam” tidak diubah dalam transkrip.
- Setiap entri hanya dapat berisi satu kata (tidak ada spasi).
- Jika Anda menyimpan filter kosakata kustom Anda sebagai file teks, itu harus dalam format teks biasa dengan pengkodean UTF-8.
- Anda dapat memiliki hingga 100 filter kosakata khusus per Akun AWS dan masing-masing dapat berukuran hingga 50 Kb.
- Anda hanya dapat menggunakan karakter yang didukung untuk bahasa Anda. Lihat set karakter bahasa Anda untuk detailnya.

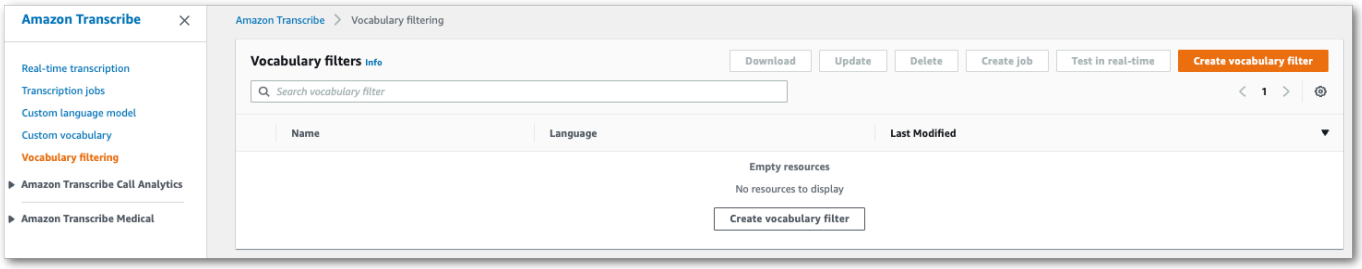
Membuat filter kosakata khusus

Untuk memproses filter kosakata khusus untuk digunakan Amazon Transcribe, lihat contoh berikut:

AWS Management Console

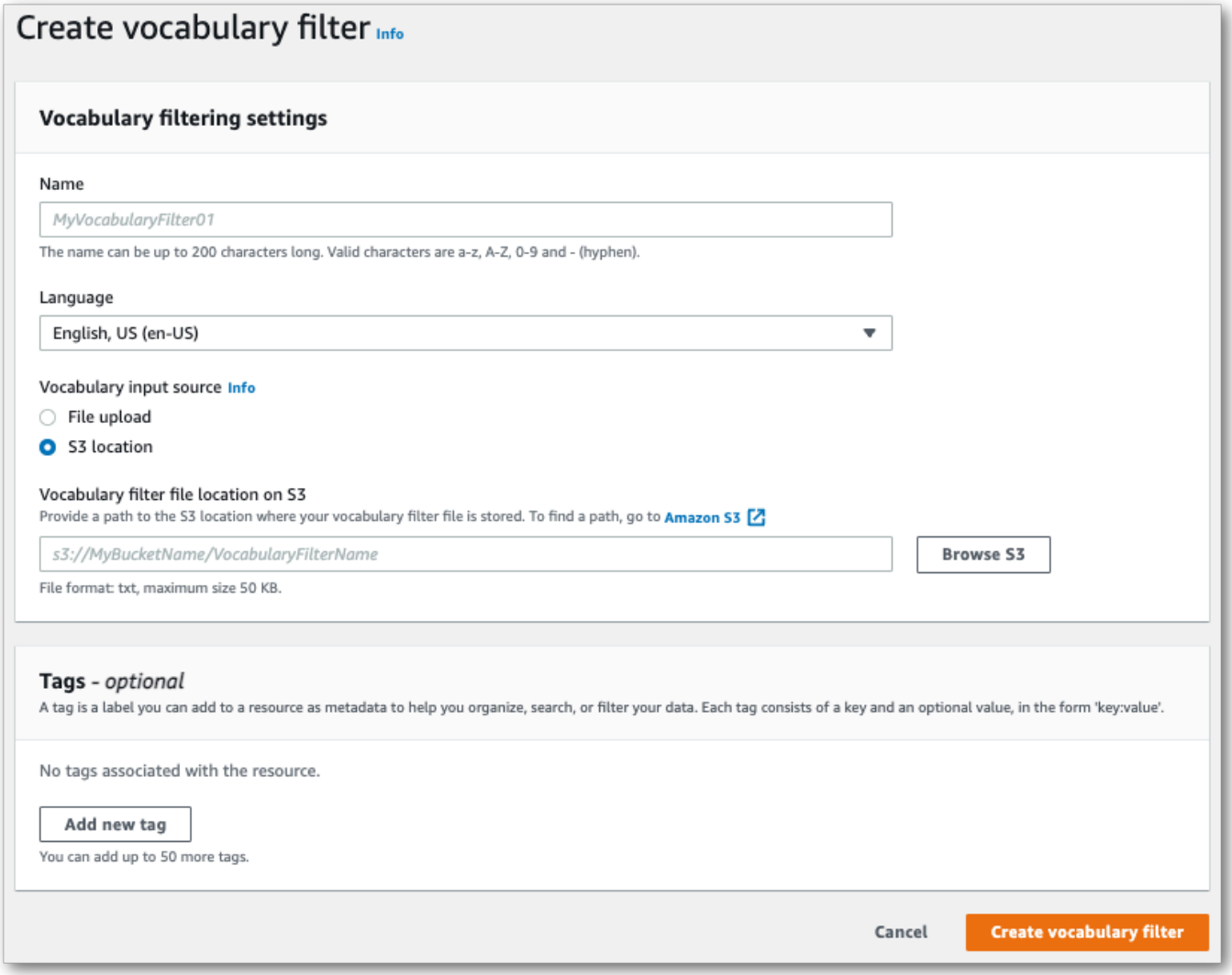
Sebelum melanjutkan, simpan filter kosakata kustom Anda sebagai file teks (*.txt). Anda dapat secara opsional mengunggah file Anda ke Amazon S3 ember.

1. Masuk ke AWS Management Console.
2. Di panel navigasi, pilih Pemfilteran kosakata. Ini membuka halaman filter Kosakata di mana Anda dapat melihat filter kosakata kustom yang ada atau membuat yang baru.
3. Pilih Buat filter kosakata.



Ini membawa Anda ke halaman filter Buat kosakata. Masukkan nama untuk filter kosakata kustom baru Anda.

Pilih opsi Unggah file atau lokasi S3 di bawah Sumber input kosakata. Kemudian tentukan lokasi file kosakata khusus Anda.



4. Secara opsional, tambahkan tag ke filter kosakata khusus Anda. Setelah semua bidang selesai, pilih Buat filter kosakata di bagian bawah halaman. Jika tidak ada kesalahan dalam memproses file Anda, ini akan membawa Anda kembali ke halaman filter Kosakata.

   Filter kosakata kustom Anda sekarang siap digunakan.

AWS CLI

Contoh ini menggunakan create-vocabulary-filterperintah untuk memproses daftar kata menjadi filter kosakata kustom yang dapat digunakan. Untuk informasi selengkapnya, lihat `CreateVocabularyFilter`.

Opsi 1: Anda dapat memasukkan daftar kata ke permintaan Anda menggunakan `words` parameter.

```
aws transcribe create-vocabulary-filter \
--vocabulary-filter-name my-first-vocabulary-filter \
--language-code en-US \
--words profane,offensive,Amazon,Transcribe
```

Opsi 2: Anda dapat menyimpan daftar kata sebagai file teks dan mengunggahnya ke Amazon S3 bucket, lalu menyertakan URI file dalam permintaan Anda menggunakan `vocabulary-filter-file-uri` parameter.

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

Berikut contoh lain menggunakan create-vocabulary-filterperintah, dan badan permintaan yang membuat filter kosakata kustom Anda.

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

File my-first-vocab-filter.json berisi badan permintaan berikut.

Opsi 1: Anda dapat memasukkan daftar kata ke permintaan Anda menggunakan `Words` parameter.

```
{
```

```
  "VocabularyFilterName": "my-first-vocabulary-filter",
  "LanguageCode": "en-US",
  "Words": [
        "profane","offensive","Amazon","Transcribe"
  ]
}
```

Opsi 2: Anda dapat menyimpan daftar kata sebagai file teks dan mengunggahnya ke Amazon S3 bucket, lalu menyertakan URI file dalam permintaan Anda menggunakan `VocabularyFilterFileUri` parameter.

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

> **Note**
>
> Jika Anda memasukkan `VocabularyFilterFileUri` dalam permintaan Anda, Anda tidak dapat menggunakan`Words`; Anda harus memilih satu atau yang lain.

AWS SDK untuk Python (Boto3)

Contoh ini menggunakan AWS SDK untuk Python (Boto3) untuk membuat filter kosakata kustom menggunakan metode create_vocabulary_filter. Untuk informasi selengkapnya, lihat `CreateVocabularyFilter`.

Untuk contoh tambahan menggunakan AWS SDKs, termasuk contoh khusus fitur, skenario, dan lintas layanan, lihat bagian ini. Contoh kode untuk Amazon Transcribe menggunakan AWS SDKs

Opsi 1: Anda dapat memasukkan daftar kata ke permintaan Anda menggunakan `Words` parameter.

```
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'
    ]
)
```

Opsi 2: Anda dapat menyimpan daftar kata sebagai file teks dan mengunggahnya ke Amazon S3 bucket, lalu menyertakan URI file dalam permintaan Anda menggunakan `VocabularyFilterFileUri` parameter.

```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
vocab_name = "my-first-vocabulary-filter"
response = transcribe.create_vocabulary_filter(
    LanguageCode = 'en-US',
    VocabularyFilterName = vocab_name,
    VocabularyFilterFileUri = 's3://amzn-s3-demo-bucket/my-vocabulary-filters/my-vocabulary-filter.txt'
)
```

> **Note**
>
> Jika Anda memasukkan `VocabularyFilterFileUri` dalam permintaan Anda, Anda tidak dapat menggunakan`Words`; Anda harus memilih satu atau yang lain.

> **Note**
>
> Jika Anda membuat Amazon S3 bucket baru untuk file filter kosakata kustom Anda, pastikan IAM peran yang membuat `CreateVocabularyFilter`permintaan memiliki izin untuk mengakses bucket ini. Jika peran tidak memiliki izin yang benar, permintaan Anda gagal. Anda dapat secara opsional menentukan IAM peran dalam permintaan Anda dengan menyertakan `DataAccessRoleArn` parameter. Untuk informasi selengkapnya tentang IAM peran dan kebijakan di Amazon Transcribe, lihatAmazon Transcribe contoh kebijakan berbasis identitas.

Menggunakan filter kosakata khusus

Setelah filter kosakata kustom Anda dibuat, Anda dapat memasukkannya ke dalam permintaan transkripsi Anda; lihat bagian berikut untuk contoh.

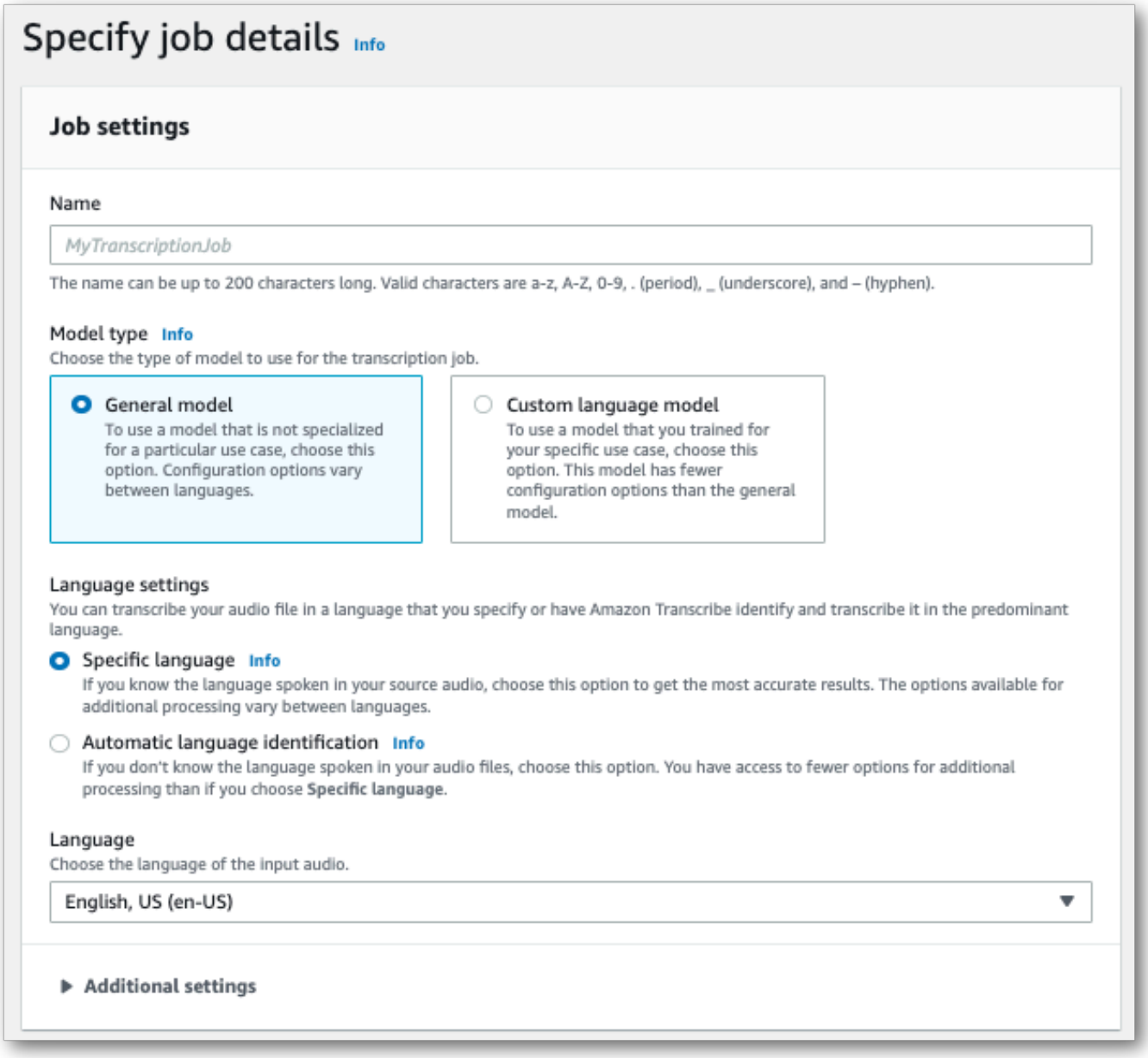
Bahasa filter kosakata kustom yang Anda sertakan dalam permintaan Anda harus cocok dengan kode bahasa yang Anda tentukan untuk media Anda. Jika Anda menggunakan identifikasi bahasa dan menentukan beberapa opsi bahasa, Anda dapat menyertakan satu filter kosakata khusus per bahasa tertentu. Jika bahasa filter kosakata kustom Anda tidak cocok dengan bahasa yang diidentifikasi dalam audio Anda, filter Anda tidak diterapkan ke transkripsi Anda dan tidak ada peringatan atau kesalahan.

Menggunakan filter kosakata khusus dalam transkripsi batch

Untuk menggunakan filter kosakata khusus dengan transkripsi batch, lihat contoh berikut ini:

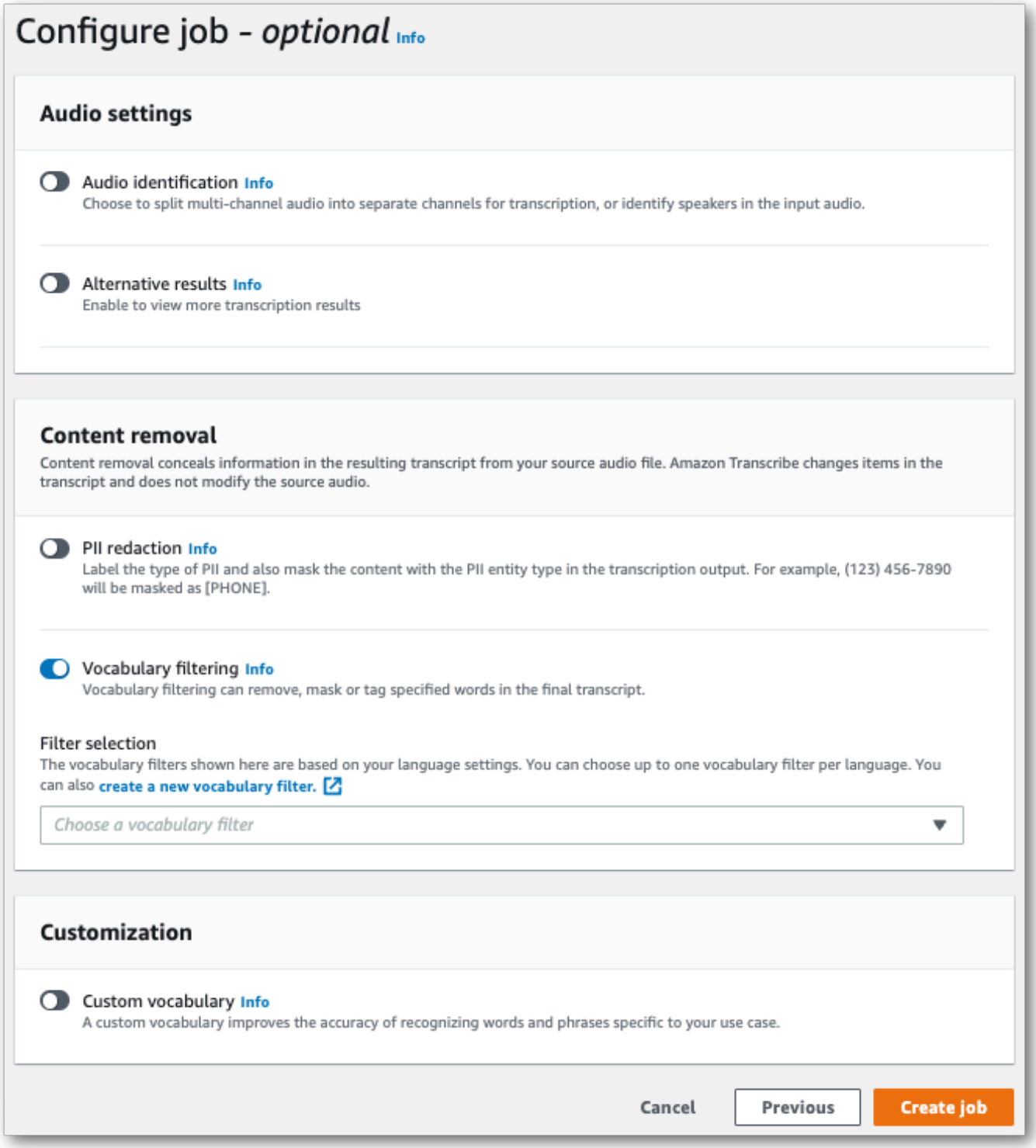
AWS Management Console

1. Masuk ke AWS Management Console.
2. Di panel navigasi, pilih Pekerjaan transkripsi, lalu pilih Buat pekerjaan (kanan atas). Ini membuka halaman Tentukan detail pekerjaan.

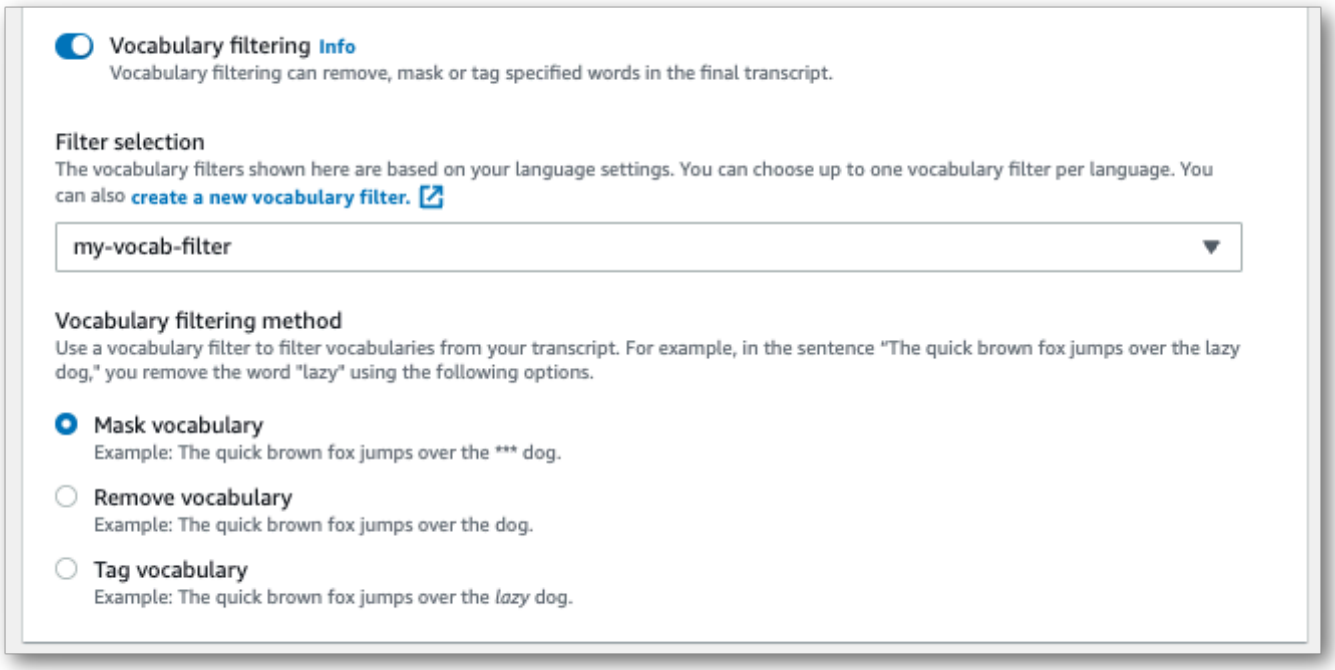


Beri nama pekerjaan Anda dan tentukan media masukan Anda. Secara opsional sertakan bidang lain, lalu pilih Berikutnya.

3. Pada halaman Configure job, di panel Penghapusan konten, aktifkan pemfilteran Kosakata.



4. Pilih filter kosakata kustom Anda dari menu tarik-turun dan tentukan metode filtrasi.



5. Pilih Buat pekerjaan untuk menjalankan pekerjaan transkripsi Anda.

AWS CLI

Contoh ini menggunakan start-transcription-jobperintah dan `Settings` parameter dengan `VocabularyFilterName` dan `VocabularyFilterMethod` sub-parameter. Untuk informasi selengkapnya, silakan lihat `StartTranscriptionJob` dan `Settings`.

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

Berikut contoh lain menggunakan start-transcription-jobperintah, dan badan permintaan yang menyertakan filter kosakata kustom Anda dengan pekerjaan itu.

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

File my-first-vocabulary-filter-job.json berisi badan permintaan berikut.

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

AWS SDK untuk Python (Boto3)

Contoh ini menggunakan AWS SDK untuk Python (Boto3) untuk menyertakan filter kosakata khusus menggunakan `Settings` argumen untuk metode start_transcription_job. Untuk informasi selengkapnya, silakan lihat `StartTranscriptionJob` dan `Settings`.

Untuk contoh tambahan menggunakan AWS SDKs, termasuk contoh khusus fitur, skenario, dan lintas layanan, lihat bagian ini. Contoh kode untuk Amazon Transcribe menggunakan AWS SDKs

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

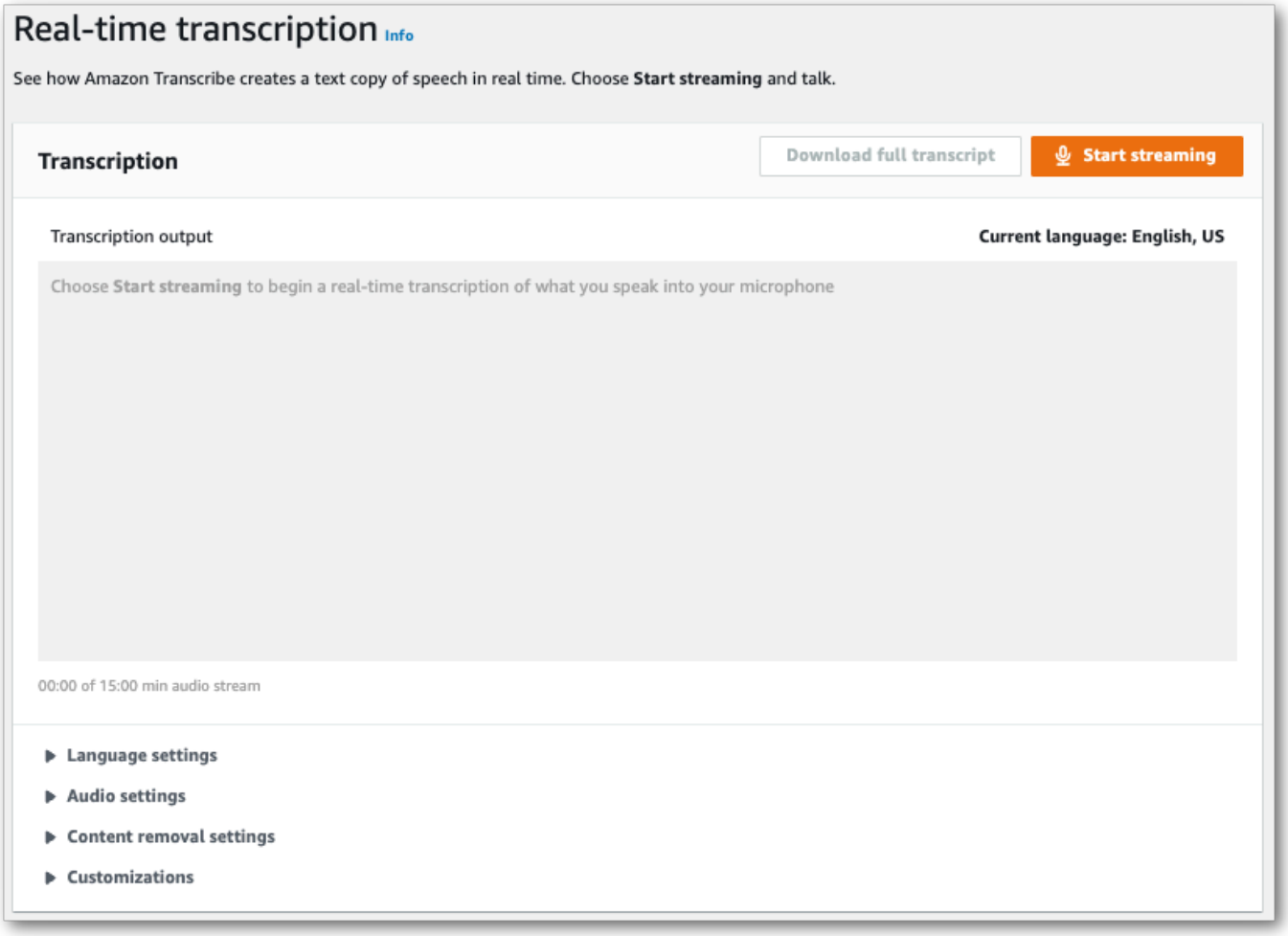
```
    }
)

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

Menggunakan filter kosakata khusus dalam transkripsi streaming

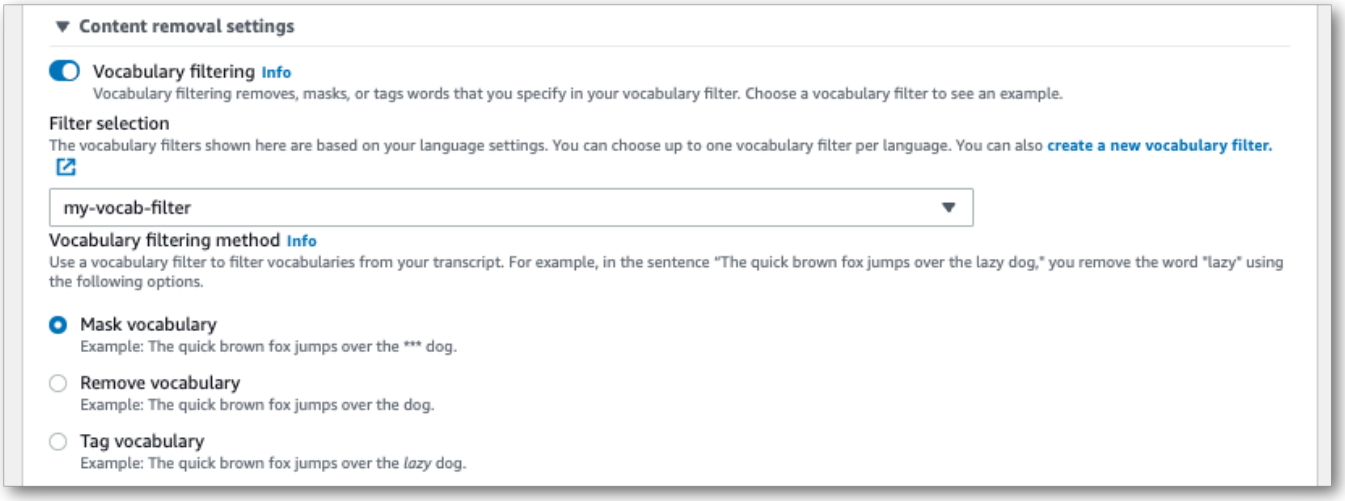
Untuk menggunakan filter kosakata khusus dengan transkripsi streaming, lihat contoh berikut ini:

AWS Management Console

1. Masuk ke AWS Management Console.
2. Di panel navigasi, pilih Transkripsi waktu nyata. Gulir ke bawah ke pengaturan penghapusan konten dan perluas bidang ini jika diminimalkan.



3. Beralih pada pemfilteran Kosakata. Pilih filter kosakata khusus dari menu tarik-turun dan tentukan metode filtrasi.



Sertakan pengaturan lain yang ingin Anda terapkan ke streaming Anda.

4. Anda sekarang siap untuk mentranskripsikan aliran Anda. Pilih Mulai streaming dan mulai berbicara. Untuk mengakhiri dikte Anda, pilih Hentikan streaming.

Aliran HTTP/2

Contoh ini membuat permintaan HTTP/2 yang menyertakan filter kosakata kustom dan metode filter Anda. Untuk informasi lebih lanjut tentang menggunakan streaming HTTP/2 dengan Amazon Transcribe, lihat. Menyiapkan aliran HTTP/2 Untuk detail selengkapnya tentang parameter dan header khusus untuk Amazon Transcribe, lihat `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
```

Definisi parameter dapat ditemukan di Referensi API; parameter yang umum untuk semua operasi AWS API tercantum di bagian Parameter Umum.

WebSocket aliran

Contoh ini membuat URL presigned yang menerapkan filter kosakata kustom Anda ke aliran. WebSocket Jeda baris telah ditambahkan untuk keterbacaan. Untuk informasi selengkapnya tentang penggunaan WebSocket stream dengan Amazon Transcribe, lihatMenyiapkan WebSocket aliran. Untuk detail lebih lanjut tentang parameter, lihat `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
```

Definisi parameter dapat ditemukan di Referensi API; parameter yang umum untuk semua operasi AWS API tercantum di bagian Parameter Umum.

Mendeteksi ucapan beracun

Deteksi ucapan beracun dirancang untuk membantu memoderasi platform media sosial yang melibatkan peer-to-peer dialog, seperti game online dan platform obrolan sosial. Penggunaan pidato beracun dapat sangat merugikan individu, kelompok sebaya, dan komunitas. Menandai bahasa berbahaya membantu organisasi menjaga percakapan tetap sipil dan menjaga lingkungan online yang aman dan inklusif bagi pengguna untuk membuat, berbagi, dan berpartisipasi secara bebas.

Amazon Transcribe Deteksi Toksisitas memanfaatkan isyarat berbasis audio dan teks untuk mengidentifikasi dan mengklasifikasikan konten beracun berbasis suara di tujuh kategori termasuk pelecehan seksual, ujaran kebencian, ancaman, pelecehan, kata-kata kotor, penghinaan, dan grafik. Selain teks, Deteksi Amazon Transcribe Toksisitas menggunakan isyarat ucapan, seperti nada dan nada untuk mengasah niat beracun dalam berbicara. Ini adalah peningkatan dari sistem moderasi konten standar yang dirancang untuk fokus hanya pada persyaratan tertentu, tanpa memperhitungkan niat.

Amazon Transcribe menandai dan mengkategorikan ucapan beracun, yang meminimalkan volume data yang harus diproses secara manual. Hal ini memungkinkan moderator konten untuk dengan cepat dan efisien mengelola wacana di platform mereka.

Kategori pidato beracun meliputi:

- Kata-kata kotor: Pidato yang berisi kata-kata, frasa, atau akronim yang tidak sopan, vulgar, atau menyinggung.
- Ucapan kebencian: Pidato yang mengkritik, menghina, mencela, atau merendahkan seseorang atau kelompok berdasarkan identitas (seperti ras, etnis, jenis kelamin, agama, orientasi seksual, kemampuan, dan asal kebangsaan).
- Seksual: Pidato yang menunjukkan minat seksual, aktivitas, atau gairah menggunakan referensi langsung atau tidak langsung ke bagian tubuh, sifat fisik, atau jenis kelamin.
- Penghinaan: Pidato yang mencakup bahasa yang merendahkan, mempermalukan, mengejek, menghina, atau meremehkan. Jenis bahasa ini juga diberi label sebagai bullying.
- Kekerasan atau ancaman: Pidato yang mencakup ancaman yang berusaha menimbulkan rasa sakit, cedera, atau permusuhan terhadap seseorang atau kelompok.
- Grafis: Pidato yang menggunakan citra visual deskriptif dan jelas yang tidak menyenangkan. Jenis bahasa ini sering sengaja bertele-tele untuk memperkuat ketidaknyamanan penerima.

- Pelecehan atau pelecehan: Pidato dimaksudkan untuk memengaruhi kesejahteraan psikologis penerima, termasuk istilah yang merendahkan dan mengobjektifikasi. Jenis bahasa ini juga diberi label sebagai pelecehan.

Deteksi toksisitas menganalisis segmen bicara (ucapan antara jeda alami) dan memberikan skor kepercayaan pada segmen ini. Skor kepercayaan adalah nilai antara 0 dan 1. Skor kepercayaan yang lebih besar menunjukkan kemungkinan yang lebih besar bahwa konten tersebut adalah ucapan beracun dalam kategori terkait. Anda dapat menggunakan skor kepercayaan ini untuk menetapkan ambang batas deteksi toksisitas yang sesuai untuk kasus penggunaan Anda.

> **Note**
>
> Deteksi toksisitas hanya tersedia untuk transkripsi batch dalam bahasa Inggris AS. (`en-US`)

Lihat contoh output dalam format JSON.

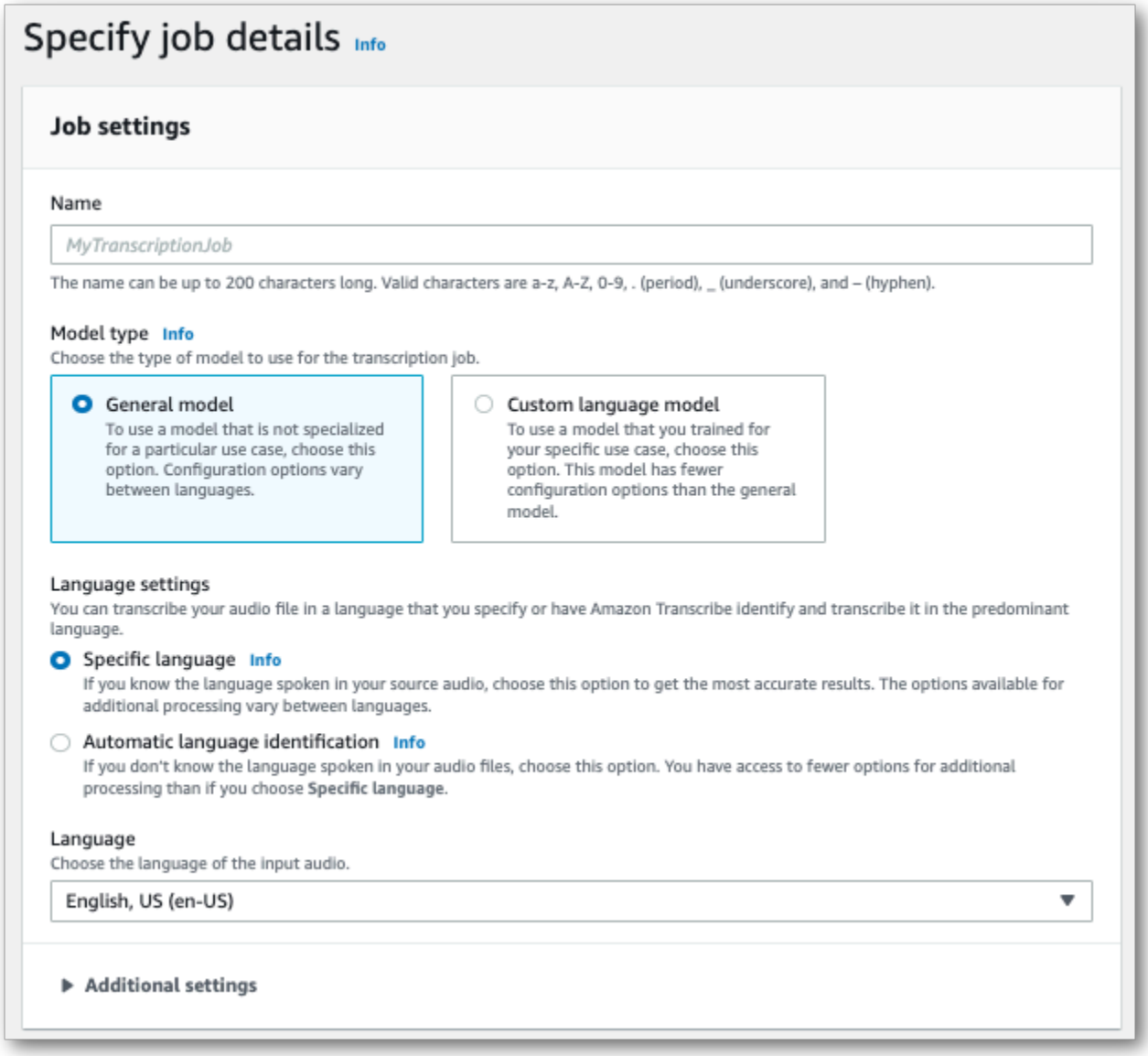
Menggunakan deteksi ucapan beracun

Menggunakan deteksi ucapan beracun dalam transkripsi batch

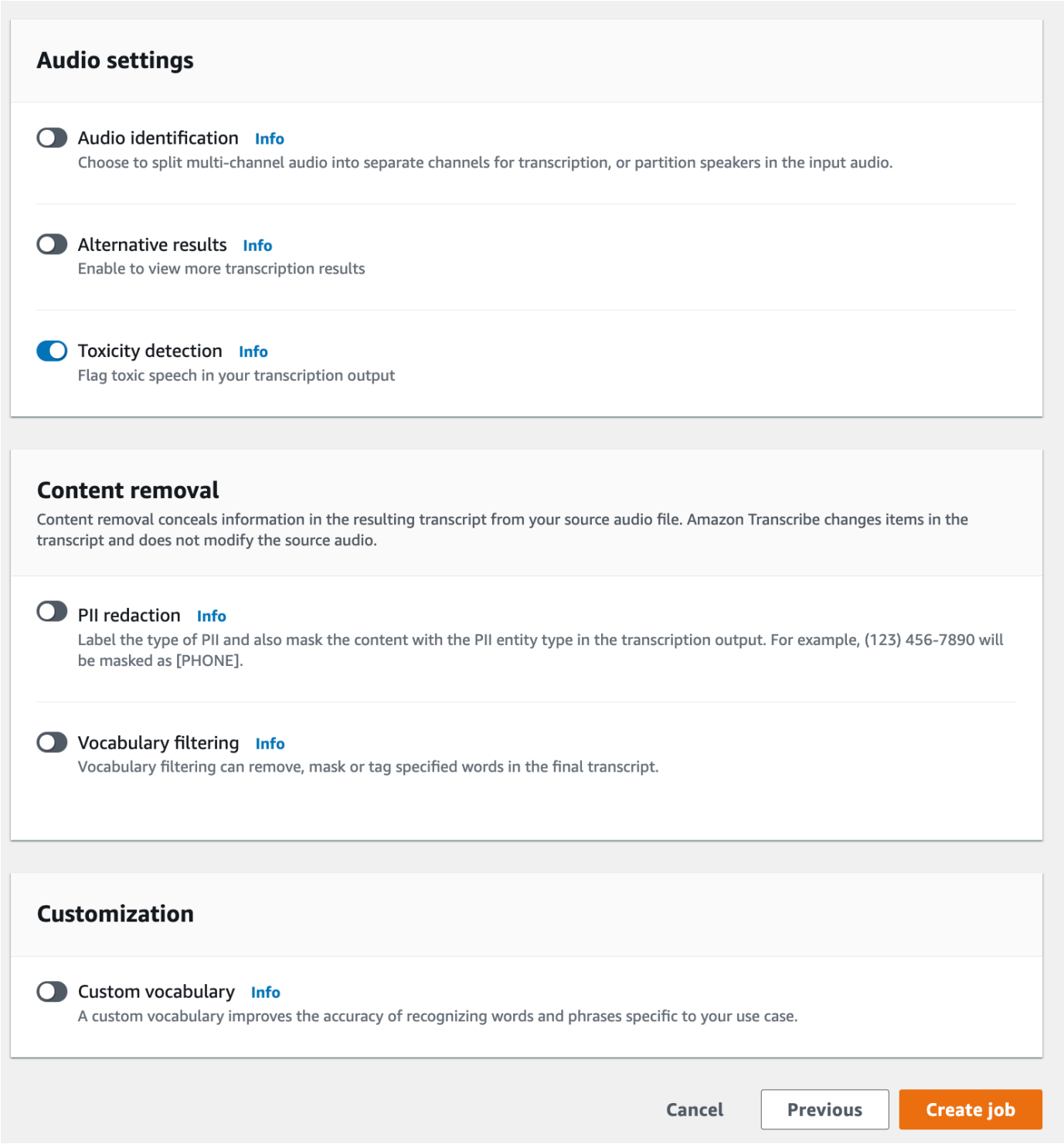
Untuk menggunakan deteksi ucapan beracun dengan transkripsi batch, lihat contoh berikut ini:

AWS Management Console

1. Masuk ke AWS Management Console.
2. Di panel navigasi, pilih Pekerjaan transkripsi, lalu pilih Buat pekerjaan (kanan atas). Ini membuka halaman Tentukan detail pekerjaan.



3. Pada halaman Tentukan detail pekerjaan, Anda juga dapat mengaktifkan redaksi PII jika Anda mau. Perhatikan bahwa opsi lain yang tercantum tidak didukung dengan deteksi Toksisitas. Pilih Selanjutnya. Ini membawa Anda ke halaman Konfigurasi pekerjaan - opsional. Di panel Pengaturan audio, pilih Deteksi toksisitas.



4. Pilih Buat pekerjaan untuk menjalankan pekerjaan transkripsi Anda.
5. Setelah pekerjaan transkripsi Anda selesai, Anda dapat mengunduh transkrip Anda dari Unduh menu tarik-turun di halaman detail pekerjaan transkripsi.

AWS CLI

Contoh ini menggunakan start-transcription-jobperintah dan `ToxicityDetection` parameter. Untuk informasi selengkapnya, silakan lihat `StartTranscriptionJob` dan `ToxicityDetection`.

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

Berikut contoh lain menggunakan start-transcription-jobperintah, dan badan permintaan yang menyertakan deteksi toksisitas.

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

File my-first-toxicity-job.json berisi badan permintaan berikut.

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

AWS SDK untuk Python (Boto3)

Contoh ini menggunakan AWS SDK untuk Python (Boto3) `ToxicityDetection` untuk mengaktifkan metode start_transcription_job. Untuk informasi selengkapnya, silakan lihat `StartTranscriptionJob` dan `ToxicityDetection`.

Untuk contoh tambahan menggunakan AWS SDKs, termasuk contoh khusus fitur, skenario, dan lintas layanan, lihat bagian ini. Contoh kode untuk Amazon Transcribe menggunakan AWS SDKs

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

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

Contoh Output

Pidato beracun ditandai dan dikategorikan dalam keluaran transkripsi Anda. Setiap contoh ucapan beracun dikategorikan dan diberi skor kepercayaan (nilai antara 0 dan 1). Nilai kepercayaan yang lebih besar menunjukkan kemungkinan yang lebih besar bahwa konten tersebut adalah ucapan beracun dalam kategori yang ditentukan.

Contoh keluaran (JSON)

Berikut ini adalah contoh keluaran dalam format JSON yang menunjukkan ucapan beracun yang dikategorikan dengan skor kepercayaan terkait.


```
{
    "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"
}
```

Menyunting atau mengidentifikasi informasi yang dapat diidentifikasi secara pribadi

Redaksi digunakan untuk menutupi atau menghapus konten sensitif, dalam bentuk informasi identitas pribadi (PII), dari transkrip Anda. Jenis PII yang Amazon Transcribe dapat menyunting bervariasi antara transkripsi batch dan streaming. Untuk melihat daftar PII untuk setiap metode transkripsi, lihat dan. Menyunting PII dalam pekerjaan batch Anda Menyunting atau mengidentifikasi PII dalam streaming waktu nyata Dengan transkripsi streaming, Anda juga memiliki opsi untuk menandai PII tanpa menyuntingnya; lihat Contoh keluaran identifikasi PII untuk contoh keluaran.

Saat redaksi diaktifkan, Anda memiliki opsi untuk hanya menghasilkan transkrip yang disunting atau transkrip yang disunting dan transkrip yang tidak disunting. Jika Anda memilih untuk membuat hanya transkrip yang disunting, perhatikan bahwa media Anda adalah satu-satunya tempat penyimpanan percakapan lengkap. Jika Anda menghapus media asli Anda, tidak ada catatan PII yang belum disunting. Karena itu, mungkin bijaksana untuk menghasilkan transkrip yang tidak disunting selain yang disunting.

Untuk mempelajari lebih lanjut tentang redaksi PII dengan transkripsi batch, lihat:. Menyunting PII dalam pekerjaan batch Anda

Untuk mempelajari lebih lanjut tentang redaksi atau identifikasi PII dengan transkripsi streaming, lihat:. Menyunting atau mengidentifikasi PII dalam streaming waktu nyata

> **Important**
>
> Fitur redaksi dirancang untuk mengidentifikasi dan menghapus data sensitif. Namun, karena sifat prediktif pembelajaran mesin, Amazon Transcribe mungkin tidak mengidentifikasi dan menghapus semua contoh data sensitif dalam transkrip Anda. Kami sangat menyarankan Anda meninjau output yang disunting untuk memastikannya memenuhi kebutuhan Anda. Fitur redaksi tidak memenuhi persyaratan untuk de-identifikasi berdasarkan undang-undang privasi medis, seperti Undang-Undang Portabilitas dan Akuntabilitas Asuransi Kesehatan AS tahun 1996 (HIPAA).

Untuk panduan video tentang fitur redaksi, lihat Menggunakan redaksi konten untuk mengidentifikasi Amazon Transcribe& menyunting PII.

Menyunting PII dalam pekerjaan batch Anda

Saat menyunting informasi identitas pribadi (PII) dari transkrip selama pekerjaan transkripsi batch, Amazon Transcribe ganti setiap instance PII yang diidentifikasi dengan `[PII]` di badan teks utama transkrip Anda. Anda juga dapat melihat jenis PII yang diedit di word-for-word bagian output transkripsi. Untuk sampel keluaran, lihatContoh keluaran yang disunting (batch).

Redaksi dengan transkripsi batch tersedia dengan bahasa Inggris AS (`en-US`) dan Spanyol AS (`es-US`). Redaksi tidak kompatibel dengan identifikasi bahasa.

Transkrip yang diedit dan tidak disunting disimpan dalam ember keluaran yang sama. Amazon S3 Amazon Transcribe menyimpannya dalam bucket yang Anda tentukan atau di Amazon S3 bucket default yang dikelola oleh layanan.

Jenis PII Amazon Transcribe dapat dikenali untuk transkripsi batch

Jenis PII	Deskripsi
`ADDRESS`	Alamat fisik, seperti 100 Main Street, Anytown, USA atau Suite #12, Building 123. Alamat dapat mencakup jalan, bangunan, lokasi, kota, negara bagian, negara, kabupaten, zip, kantor polisi, lingkungan, dan banyak lagi.
`ALL`	Menyunting atau mengidentifikasi semua jenis PII yang tercantum dalam tabel ini.
`BANK_ACCOUNT_NUMBER`	Nomor rekening bank AS. Ini biasanya antara 10 - 12 digit panjang, tetapi Amazon Transcrib e juga mengenali nomor rekening bank ketika hanya 4 digit terakhir yang ada.
`BANK_ROUTING`	Nomor perutean rekening bank AS. Ini biasanya 9 digit panjang, tetapi Amazon Transcribe juga mengenali nomor routing ketika hanya 4 digit terakhir yang ada.
`CREDIT_DEBIT_CVV`	Kode verifikasi kartu 3 digit (CVV) yang ada di VISA, MasterCard, serta Discover kartu kredit dan debit. Dalam kartu kredit atau debit

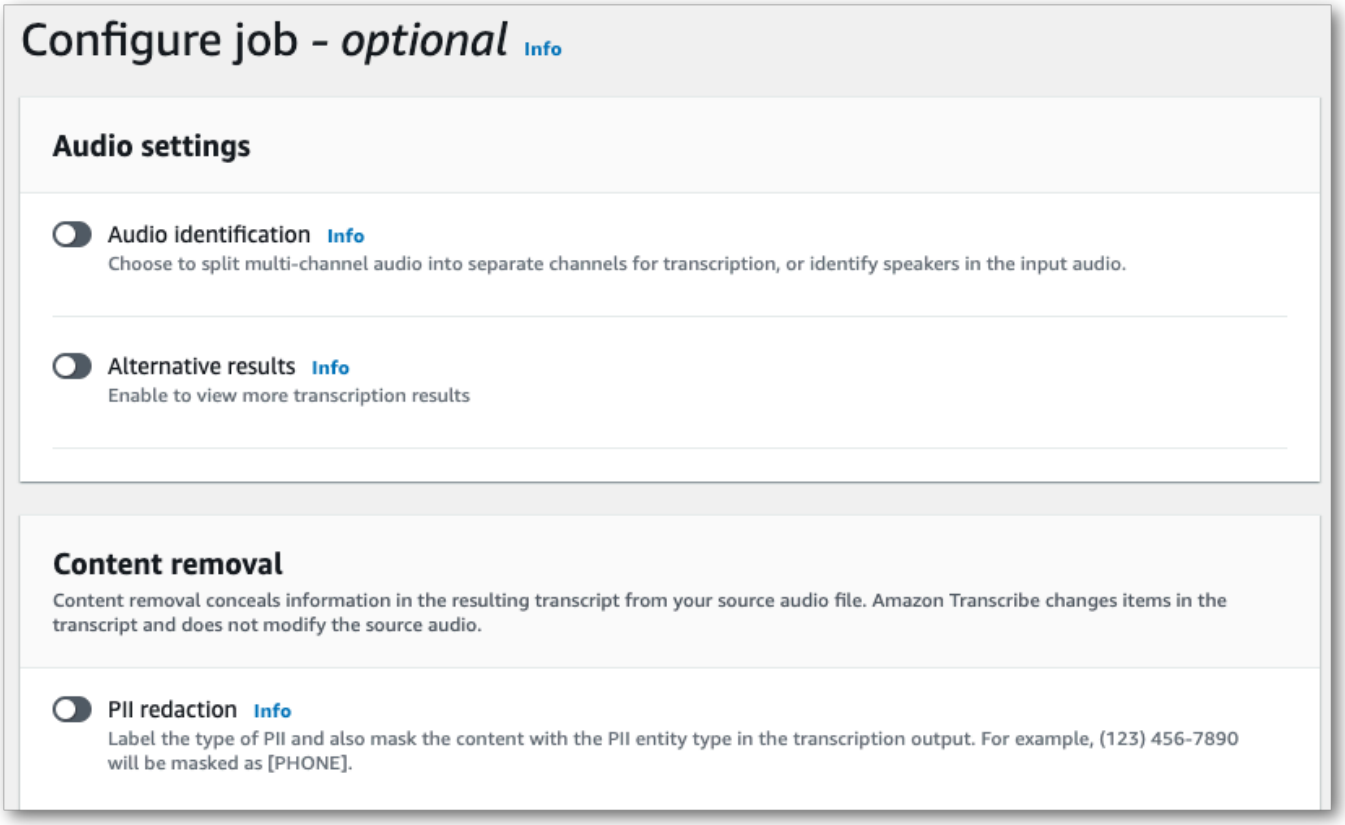
Jenis PII	Deskripsi
	American Express, ini adalah kode numerik 4 digit.
CREDIT_DEBIT_EXPIRY	Tanggal kedaluwarsa untuk kartu kredit atau debit. Angka ini biasanya panjangnya 4 digit dan diformat sebagai. month/year or MM/YY Misalnya, Amazon Transcribe dapat mengenali tanggal kedaluwarsa seperti 01/21, 01/2021, dan Jan 2021.
CREDIT_DEBIT_NUMBER	Nomor untuk kartu kredit atau debit. Angka-angka ini dapat bervariasi dari 13 hingga 16 digit panjangnya, tetapi Amazon Transcribe juga mengenali nomor kartu kredit atau debit ketika hanya 4 digit terakhir yang ada.
EMAIL	Alamat email, seperti efua.owusu@email.com.
NAME	Nama seorang individu. Jenis entitas ini tidak termasuk judul, seperti Mr., Mrs., Miss, atau Dr. Amazon Transcribe tidak menerapkan jenis entitas ini untuk nama yang merupakan bagian dari organisasi atau alamat. Misalnya, Amazon Transcribe mengakui Organisasi John Doe sebagai sebuah organisasi, dan Jane Doe Street sebagai alamat.
PHONE	Sebuah nomor telepon. Jenis entitas ini juga mencakup nomor faks dan pager.
PIN	4 digit nomor identifikasi pribadi (PIN) yang memungkinkan seseorang untuk mengakses informasi rekening bank mereka.

Jenis PII	Deskripsi
SSN	Nomor Jaminan Sosial (SSN) adalah nomor 9 digit yang dikeluarkan untuk warga negara AS, penduduk tetap, dan penduduk yang bekerja sementara. Amazon Transcribe juga mengenali Nomor Jaminan Sosial ketika hanya 4 digit terakhir yang ada.

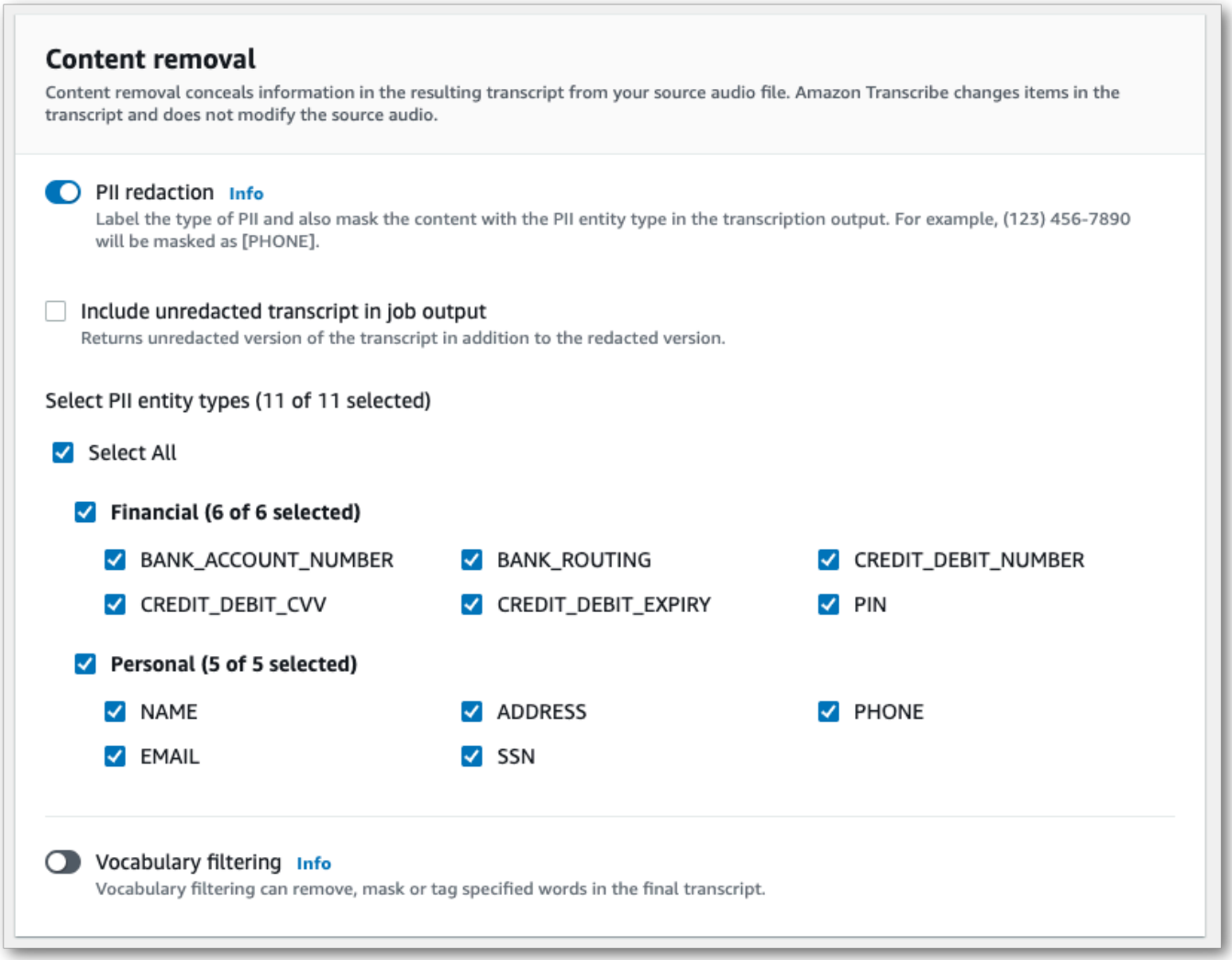
Anda dapat memulai pekerjaan transkripsi batch menggunakan AWS Management Console, AWS CLI, atau AWS SDK.

AWS Management Console

1. Masuk ke AWS Management Console.
2. Di panel navigasi, pilih Pekerjaan transkripsi, lalu pilih Buat pekerjaan (kanan atas). Ini akan membuka halaman Tentukan detail pekerjaan.
3. Setelah mengisi bidang yang Anda inginkan pada halaman Tentukan detail pekerjaan, pilih Berikutnya untuk pergi ke halaman Konfigurasi pekerjaan - opsional. Di sini Anda akan menemukan panel penghapusan konten dengan sakelar redaksi PII.



4. Setelah Anda memilih redaksi PII, Anda memiliki opsi untuk memilih semua jenis PII yang ingin Anda edit. Anda juga dapat memilih untuk memiliki transkrip yang tidak disunting jika Anda memilih Sertakan transkrip yang tidak disunting di kotak keluaran pekerjaan.



5. Pilih Buat pekerjaan untuk menjalankan pekerjaan transkripsi Anda.

AWS CLI

Contoh ini menggunakan start-transcription-jobperintah dan `content-redaction` parameter. Untuk informasi selengkapnya, silakan lihat `StartTranscriptionJob` dan `ContentRedaction`.

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

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

Berikut contoh lain menggunakan start-transcription-jobmetode ini, dan badan permintaan menyunting PII untuk pekerjaan itu.

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

File my-first-redaction-job.json berisi badan permintaan berikut.

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

AWS SDK untuk Python (Boto3)

Contoh ini menggunakan AWS SDK untuk Python (Boto3) untuk menyunting konten menggunakan `ContentRedaction` argumen untuk metode start_transcription_job. Untuk informasi selengkapnya, silakan lihat `StartTranscriptionJob` dan `ContentRedaction`.

Untuk contoh tambahan menggunakan AWS SDKs, termasuk contoh khusus fitur, skenario, dan lintas layanan, lihat bagian ini. Contoh kode untuk Amazon Transcribe menggunakan AWS SDKs

```
from __future__ import print_function
```

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

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

Note

Redaksi PII untuk pekerjaan batch hanya didukung dalam hal ini Wilayah AWS: Asia Pasifik (Hong Kong), Asia Pasifik (Mumbai), Asia Pasifik (Seoul), Asia Pasifik (Singapura), Asia Pasifik (Sydney), Asia Pasifik (Tokyo), (AS-Barat), Kanada GovCloud (Tengah), UE (Frankfurt), UE (Irlandia), UE (London), Uni Eropa (Paris), Timur Tengah (Bahia Bahrain), Amerika Selatan (Sao Paulo), AS Timur (Virginia N.), AS Timur (Ohio), AS Barat (Oregon), dan AS Barat (California N.).

Menyunting atau mengidentifikasi PII dalam streaming waktu nyata

Saat menyunting informasi identitas pribadi (PII) dari transkripsi streaming, Amazon Transcribe ganti setiap instance PII yang diidentifikasi dengan transkrip Anda. `[PII]`

Opsi tambahan yang tersedia untuk transkripsi streaming adalah identifikasi PII. Saat Anda mengaktifkan Identifikasi PII, beri Amazon Transcribe label PII dalam hasil transkripsi Anda di bawah suatu objek. `Entities` Untuk sampel keluaran, lihat Contoh output streaming yang disunting dan Contoh keluaran identifikasi PII.

Redaksi dan identifikasi PII dengan transkripsi streaming tersedia dengan dialek bahasa Inggris ini: Australia (`en-AU`), Inggris (), AS (`en-GB`) dan dialek AS Spanyol (`en-US`). `es-US`

Identifikasi dan redaksi PII untuk pekerjaan streaming dilakukan hanya setelah transkripsi lengkap segmen audio.

Jenis PII Amazon Transcribe dapat mengenali transkripsi streaming

Tipe PII	Deskripsi
`ADDRESS`	Alamat fisik, seperti 100 Main Street, Anytown, USA atau Suite #12, Building 123. Alamat dapat mencakup jalan, bangunan, lokasi, kota, negara bagian, negara, kabupaten, zip, kantor polisi, lingkungan, dan banyak lagi.
`ALL`	Menyunting atau mengidentifikasi semua jenis PII yang tercantum dalam tabel ini.
`BANK_ACCOUNT_NUMBER`	Nomor rekening bank AS. Ini biasanya antara 10 - 12 digit panjang, tetapi Amazon Transcrib e juga mengenali nomor rekening bank ketika hanya 4 digit terakhir yang ada.
`BANK_ROUTING`	Nomor perutean rekening bank AS. Ini biasanya 9 digit panjang, tetapi Amazon Transcribe juga mengenali nomor routing ketika hanya 4 digit terakhir yang ada.

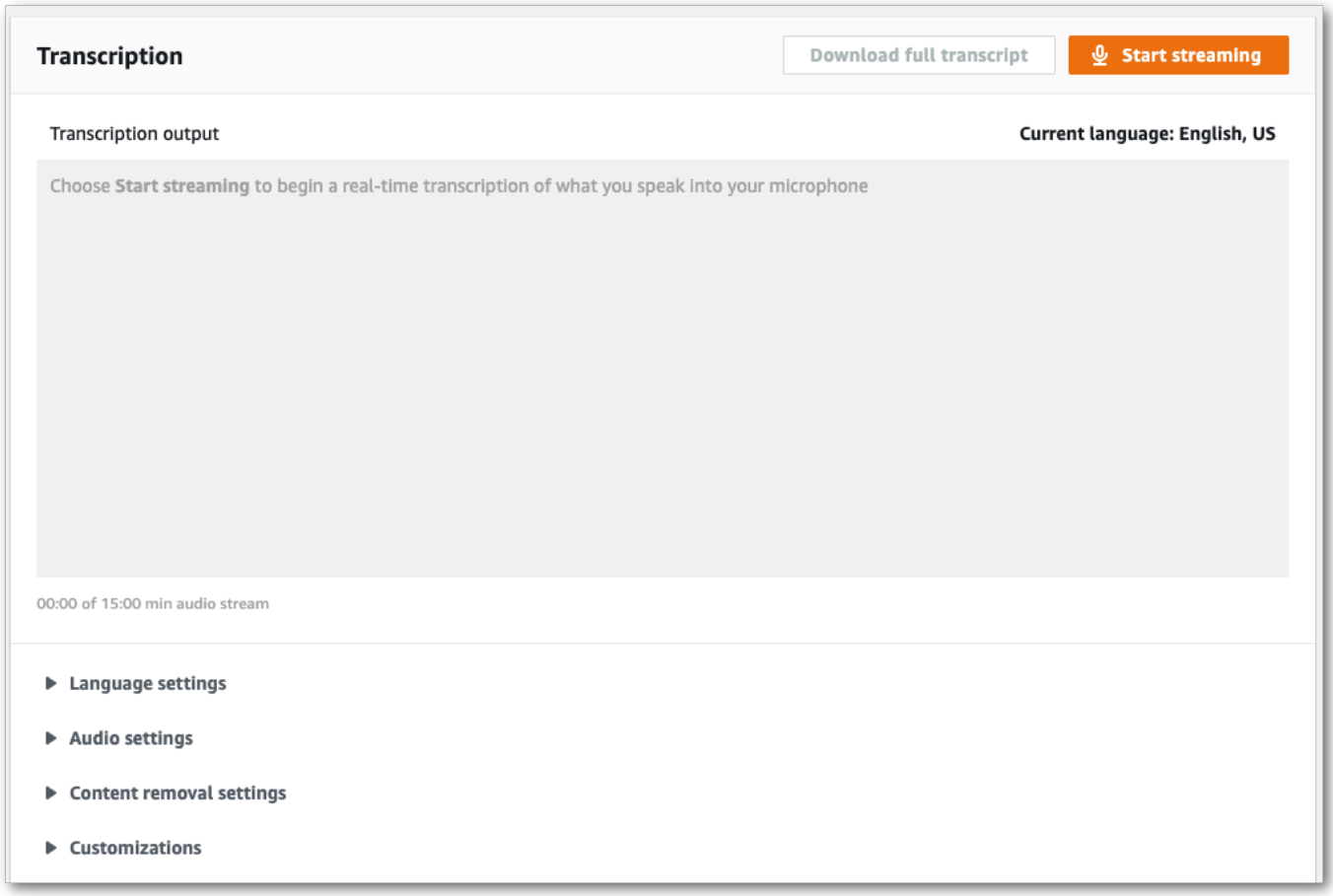
Tipe PII	Deskripsi
CREDIT_DEBIT_CVV	Kode verifikasi kartu 3 digit (CVV) yang ada di VISA, MasterCard, serta Discover kartu kredit dan debit. Dalam kartu kredit atau debit American Express, ini adalah kode numerik 4 digit.
CREDIT_DEBIT_EXPIRY	Tanggal kedaluwarsa untuk kartu kredit atau debit. Angka ini biasanya panjangnya 4 digit dan diformat sebagai. month/year or MM/YY Misalnya, Amazon Transcribe dapat mengenali tanggal kedaluwarsa seperti 01/21, 01/2021, dan Jan 2021.
CREDIT_DEBIT_NUMBER	Nomor untuk kartu kredit atau debit. Angka-angka ini dapat bervariasi dari 13 hingga 16 digit panjangnya, tetapi Amazon Transcribe juga mengenali nomor kartu kredit atau debit ketika hanya 4 digit terakhir yang ada.
EMAIL	Alamat email, seperti efua.owusu@email.com.
NAME	Nama seorang individu. Jenis entitas ini tidak termasuk judul, seperti Mr., Mrs., Miss, atau Dr. Amazon Transcribe tidak menerapkan jenis entitas ini untuk nama yang merupakan bagian dari organisasi atau alamat. Misalnya, Amazon Transcribe mengakui Organisasi John Doe sebagai sebuah organisasi, dan Jane Doe Street sebagai alamat.
PHONE	Sebuah nomor telepon. Jenis entitas ini juga mencakup nomor faks dan pager.
PIN	4 digit nomor identifikasi pribadi (PIN) yang memungkinkan seseorang untuk mengakses informasi rekening bank mereka.

Tipe PII	Deskripsi
SSN	Nomor Jaminan Sosial (SSN) adalah nomor 9 digit yang dikeluarkan untuk warga negara AS, penduduk tetap, dan penduduk yang bekerja sementara. Amazon Transcribe juga mengenali Nomor Jaminan Sosial ketika hanya 4 digit terakhir yang ada.

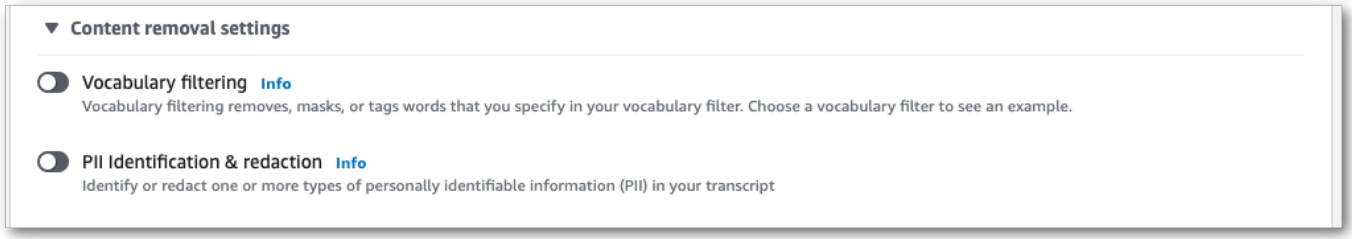
Anda dapat memulai transkripsi streaming menggunakan AWS Management Console, WebSocket, atau HTTP/2.

AWS Management Console

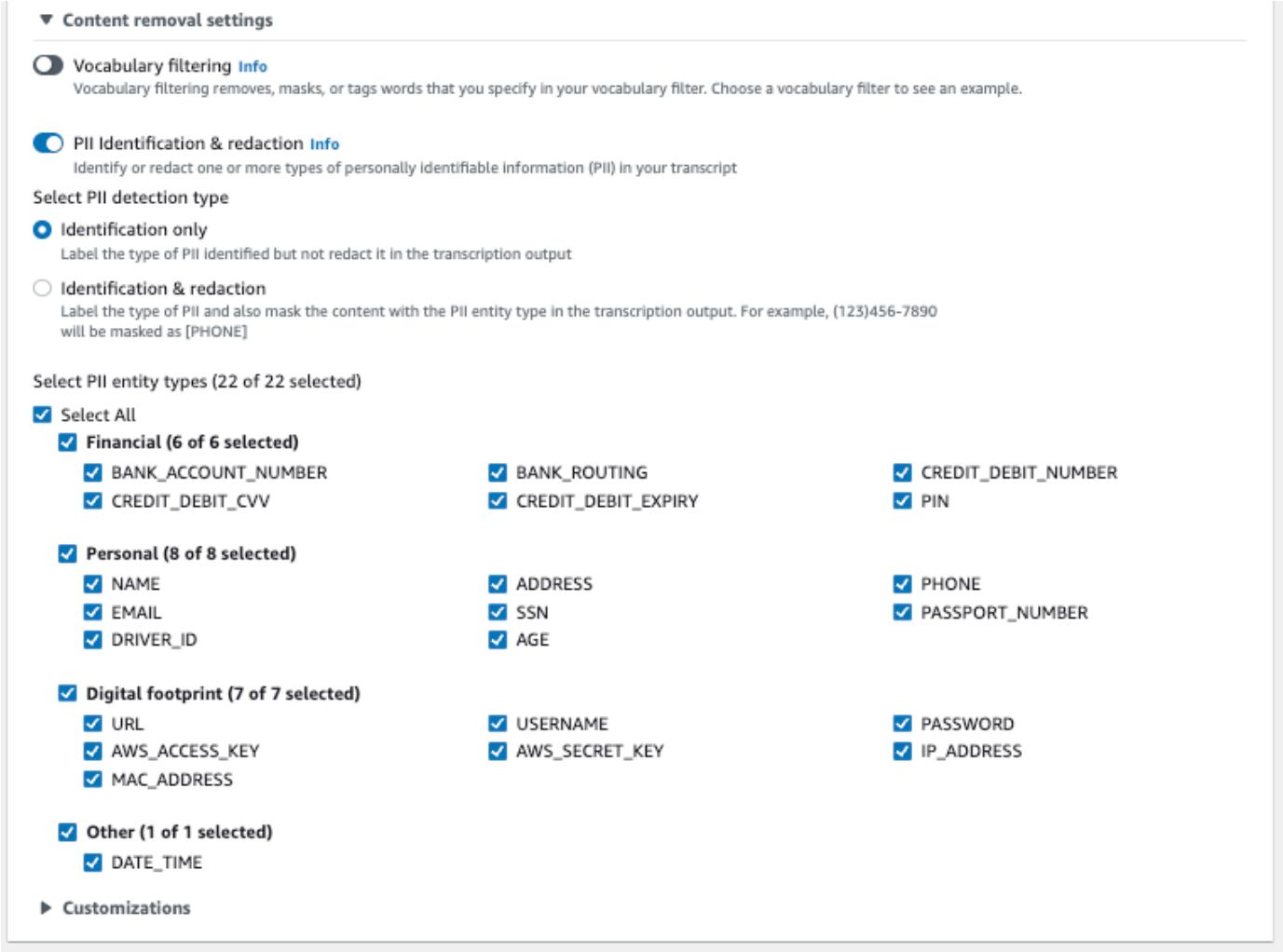
1. Masuk ke AWS Management Console.
2. Di panel navigasi, pilih Transkripsi waktu nyata. Gulir ke bawah ke pengaturan penghapusan konten dan perluas bidang ini jika diminimalkan.



3. Aktifkan Identifikasi & redaksi PII.



4. Pilih Identifikasi saja atau Identifikasi & redaksi, lalu pilih jenis entitas PII yang ingin Anda identifikasi atau edit dalam transkrip Anda.



5. Anda sekarang siap untuk mentranskripsikan aliran Anda. Pilih Mulai streaming dan mulai berbicara. Untuk mengakhiri dikte Anda, pilih Hentikan streaming.

WebSocket aliran

Contoh ini membuat URL presigned yang menggunakan redaksi PII (atau identifikasi PII) dalam aliran. WebSocket Jeda baris telah ditambahkan untuk keterbacaan. Untuk informasi selengkapnya tentang penggunaan WebSocket stream dengan Amazon Transcribe, lihatMenyiapkan WebSocket aliran. Untuk detail lebih lanjut tentang parameter, lihat `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)
```

Anda tidak dapat menggunakan keduanya `content-identification-type` dan `content-redaction-type` dalam permintaan yang sama.

Definisi parameter dapat ditemukan di Referensi API; parameter yang umum untuk semua operasi AWS API tercantum di bagian Parameter Umum.

Aliran HTTP/2

Contoh ini membuat permintaan HTTP/2 dengan identifikasi PII atau redaksi PII diaktifkan. Untuk informasi lebih lanjut tentang menggunakan streaming HTTP/2 dengan Amazon Transcribe, lihat. Menyiapkan aliran HTTP/2 Untuk detail selengkapnya tentang parameter dan header khusus untuk Amazon Transcribe, lihat `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
```

Anda tidak dapat menggunakan keduanya `content-identification-type` dan `content-redaction-type` dalam permintaan yang sama.

Definisi parameter dapat ditemukan di Referensi API; parameter yang umum untuk semua operasi AWS API tercantum di bagian Parameter Umum.

> **Note**
>
> Redaksi PII untuk streaming hanya didukung dalam hal ini Wilayah AWS: Asia Pasifik (Seoul), Asia Pasifik (Sydney), Asia Pasifik (Tokyo), Kanada (Tengah), UE (Frankfurt), UE (Irlandia), UE (London), AS Timur (Virginia N.), AS Timur (Ohio), dan AS Barat (Oregon).

Contoh output redaksi dan identifikasi PII

Contoh berikut menunjukkan output yang disunting dari pekerjaan batch dan streaming, dan identifikasi PII dari pekerjaan streaming.

Pekerjaan transkripsi menggunakan redaksi konten menghasilkan dua jenis nilai. `confidence` Keyakinan Automatic Speech Recognition (ASR) menunjukkan item `type` yang memiliki `pronunciation` atau `punctuation` merupakan ucapan tertentu. Dalam output transkrip berikut, kata tersebut Good memiliki a `confidence` dari. `1.0` Nilai kepercayaan ini menunjukkan bahwa Amazon Transcribe 100 persen yakin bahwa kata yang diucapkan dalam transkrip ini adalah 'Baik'. `confidence`Nilai untuk sebuah `[PII]` tag adalah keyakinan bahwa pidato yang ditandai untuk redaksi benar-benar PII. Dalam output transkrip berikut, `confidence` dari `0.9999` menunjukkan bahwa 99,99 persen yakin bahwa Amazon Transcribe entitas yang disuntingnya dalam transkrip adalah PII.

Contoh keluaran yang disunting (batch)

```
{
    "jobName": "my-first-transcription-job",
    "accountId": "111122223333",
    "isRedacted": true,
    "results": {
        "transcripts": [
            {
                "transcript": "Good morning, everybody. My name is [PII], and today I
 feel like
                sharing a whole lot of personal information with you. Let's start with
 my Social
                Security number [PII]. My credit card number is [PII] and my C V V code
 is [PII].
```

```
                I hope that Amazon Transcribe is doing a good job at redacting that
 personal
                information away. Let's check."
            }
        ],
        "items": [
            {
                "id": 0,
                "start_time": "2.86",
                "end_time": "3.35",
                "alternatives": [
                    {
                        "confidence": "1.0",
                        "content": "Good"
                    }
                ],
                "type": "pronunciation"
            },
            Items removed for brevity
            {
                "id": 8,
                "start_time": "5.56",
                "end_time": "6.25",
                "alternatives": [
                    {
                        "content": "[PII]",
                        "redactions": [
                            {
                                "confidence": "0.9999",
                                "type": "NAME",
                                "category": "PII"
                            }
                        ]
                    }
                ],
                "type": "pronunciation"
            },
            Items removed for brevity
        ],
    },
    "status": "COMPLETED"
}
```

Berikut transkrip yang belum disunting untuk perbandingan:


```
{
    "jobName": "job id",
    "accountId": "111122223333",
    "isRedacted": false,
    "results": {
        "transcripts": [
            {
                "transcript": "Good morning, everybody. My name is Mike, and today I
 feel like
                sharing a whole lot of personal information with you. Let's start with
 my Social
                Security number 000000000. My credit card number is 5555555555555555
                and my C V V code is 000. I hope that Amazon Transcribe is doing a good
 job
                at redacting that personal information away. Let's check."
            }
        ],
        "items": [
            {
                "id": 0,
                "start_time": "2.86",
                "end_time": "3.35",
                "alternatives": [
                    {
                        "confidence": "1.0",
                        "content": "Good"
                    }
                ],
                "type": "pronunciation"
            },
            Items removed for brevity
            {
                "id": 8,
                "start_time": "5.56",
                "end_time": "6.25",
                "alternatives": [
                    {
                        "confidence": "0.9999",
                        "content": "Mike",
                     {
                ],
                "type": "pronunciation"
```

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

Contoh output streaming yang disunting

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

Contoh keluaran identifikasi PII

Identifikasi PII adalah fitur tambahan yang dapat Anda gunakan dengan pekerjaan transkripsi streaming Anda. PII yang diidentifikasi tercantum di `Entities` bagian setiap segmen.

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

```
}
```

Membuat subtitle video

Amazon Transcribe mendukung keluaran WebVTT (*.vtt) dan (*.srt) SubRip untuk digunakan sebagai subtitle video. Anda dapat memilih satu atau kedua jenis file saat menyiapkan pekerjaan transkripsi video batch Anda. Saat menggunakan fitur subtitle, file subtitle yang Anda pilih dan file transkrip biasa (berisi informasi tambahan) diproduksi. File subtitle dan transkripsi dihasilkan ke tujuan yang sama.

Subtitle ditampilkan pada saat yang sama teks diucapkan, dan tetap terlihat sampai ada jeda alami atau pembicara selesai berbicara. Perhatikan bahwa jika Anda mengaktifkan subtitle dalam permintaan transkripsi dan audio Anda tidak berisi ucapan, file subtitle tidak akan dibuat.

> **Important**
>
> Amazon Transcribe menggunakan indeks awal default `0` untuk output subtitle, yang berbeda dari nilai yang lebih banyak digunakan. 1 Jika Anda memerlukan indeks awal1, Anda dapat menentukan ini di AWS Management Console atau di permintaan API Anda menggunakan `OutputStartIndex`parameter.

Menggunakan indeks awal yang salah dapat mengakibatkan kesalahan kompatibilitas dengan layanan lain, jadi pastikan untuk memverifikasi indeks awal yang Anda butuhkan sebelum membuat subtitle Anda. Jika Anda tidak yakin nilai mana yang akan digunakan, sebaiknya pilih1. Lihat `Subtitles`untuk informasi lebih lanjut.

Fitur yang didukung dengan subtitle:

- Redaksi konten - Setiap konten yang disunting tercermin sebagai '`PII`' di file keluaran subtitle dan transkrip reguler Anda. Audio tidak diubah.
- Filter kosakata — File subtitle dihasilkan dari file transkripsi, jadi kata apa pun yang Anda filter dalam keluaran transkripsi standar Anda juga difilter dalam subtitle Anda. Konten yang difilter ditampilkan sebagai spasi putih atau `***` dalam file transkrip dan subtitle Anda. Audio tidak diubah.
- Diarisasi speaker — Jika ada beberapa speaker di segmen subtitle tertentu, tanda hubung digunakan untuk membedakan setiap pembicara. Ini berlaku untuk WebVTT SubRip dan format; misalnya:
  - -- Teks yang diucapkan oleh Orang 1
  - -- Teks yang diucapkan oleh Orang 2

File subtitle disimpan di Amazon S3 lokasi yang sama dengan output transkripsi Anda.

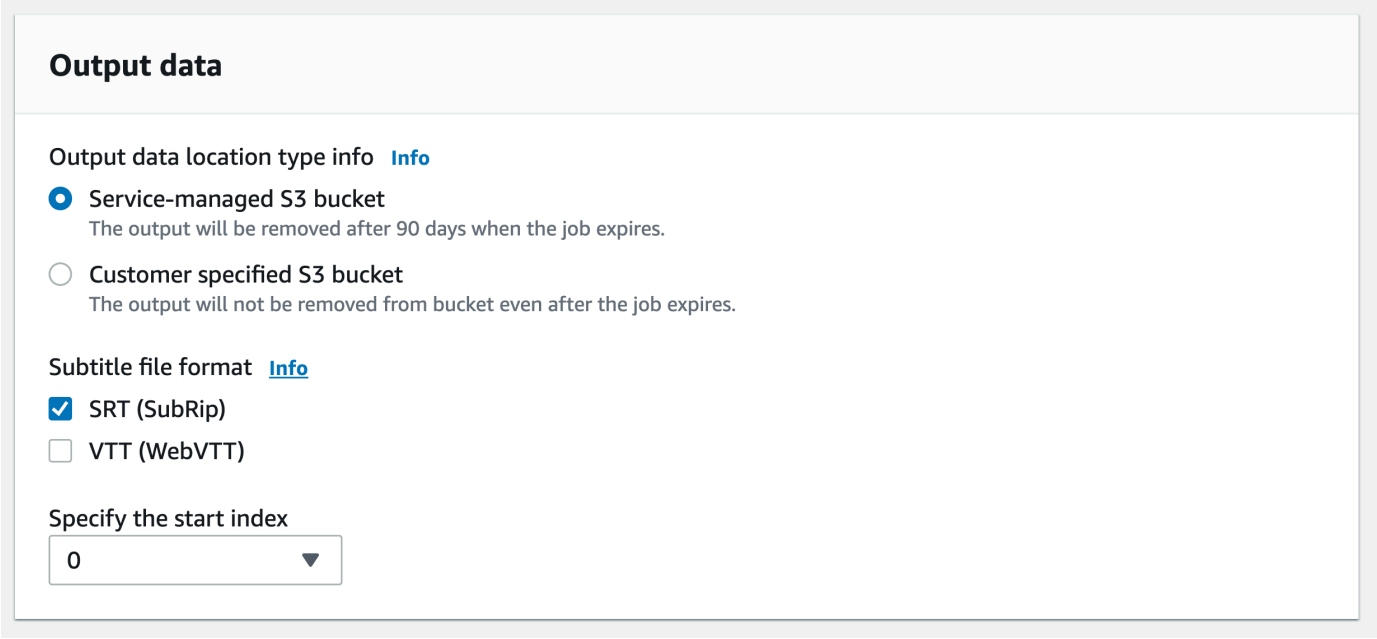
Lihat Amazon Transcribe Video Snacks: Membuat Subtitle Video Tanpa Menulis Kode Apa Pun untuk panduan video dalam membuat subtitle.

Menghasilkan file subtitle

Anda dapat membuat file subtitle menggunakan AWS Management Console, AWS CLI, atau AWS SDKs; lihat contoh berikut:

AWS Management Console

1. Masuk ke AWS Management Console.
2. Di panel navigasi, pilih Pekerjaan transkripsi, lalu pilih Buat pekerjaan (kanan atas). Ini membuka halaman Tentukan detail pekerjaan. Opsi subtitle terletak di panel data Output.
3. Pilih format yang Anda inginkan untuk file subtitle Anda, lalu pilih nilai untuk indeks awal Anda. Perhatikan bahwa Amazon Transcribe defaultnya 1 adalah0, tetapi lebih banyak digunakan. Jika Anda tidak yakin nilai mana yang akan digunakan, sebaiknya pilih1, karena ini dapat meningkatkan kompatibilitas dengan layanan lain.



4. Isi kolom lain yang ingin Anda sertakan di halaman Tentukan detail pekerjaan, lalu pilih Berikutnya. Ini membawa Anda ke halaman Konfigurasi pekerjaan - opsional.
5. Pilih Buat pekerjaan untuk menjalankan pekerjaan transkripsi Anda.

AWS CLI

Contoh ini menggunakan start-transcription-jobperintah dan `Subtitles` parameter. Untuk informasi selengkapnya, silakan lihat `StartTranscriptionJob` dan `Subtitles`.

```
aws transcribe start-transcription-job \
--region us-west-2 \
--transcription-job-name my-first-transcription-job \
--media MediaFileUri=s3://amzn-s3-demo-bucket/my-input-files/my-media-file.flac \
--output-bucket-name amzn-s3-demo-bucket \
--output-key my-output-files/ \
--language-code en-US \
--subtitles Formats=vtt,srt,OutputStartIndex=1
```

Berikut contoh lain menggunakan start-transcription-jobperintah, dan badan permintaan yang menambahkan subtitle ke pekerjaan itu.

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

File my-first-subtitle-job.json berisi badan permintaan berikut.

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

AWS SDK untuk Python (Boto3)

Contoh ini menggunakan AWS SDK untuk Python (Boto3) untuk menambahkan subtitle menggunakan `Subtitles` argumen untuk metode start_transcription_job. Untuk informasi selengkapnya, silakan lihat `StartTranscriptionJob` dan `Subtitles`.

Untuk contoh tambahan menggunakan AWS SDKs, termasuk contoh khusus fitur, skenario, dan lintas layanan, lihat bagian ini. Contoh kode untuk Amazon Transcribe menggunakan AWS SDKs

```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
job_name = "my-first-transcription-job"
job_uri = "s3://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
transcribe.start_transcription_job(
    TranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'amzn-s3-demo-bucket',
    OutputKey = 'my-output-files/',
    LanguageCode = 'en-US',
    Subtitles = {
        'Formats': [
            'vtt','srt'
        ],
        'OutputStartIndex': 1
   }
)

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

Menganalisis audio call center dengan Call Analytics

Gunakan Amazon Transcribe Call Analytics untuk mendapatkan wawasan tentang interaksi agen-pelanggan. Call Analytics dirancang khusus untuk audio call center dan secara otomatis memberi Anda data berharga yang berkaitan dengan setiap panggilan dan setiap peserta. Anda juga dapat mengasah data pada titik-titik tertentu selama panggilan. Misalnya, Anda dapat membandingkan sentimen pelanggan dalam beberapa detik pertama panggilan dengan kuartal terakhir panggilan untuk melihat apakah agen Anda memberikan pengalaman positif. Contoh kasus penggunaan lainnya tercantum di bagian berikut.

Call Analytics tersedia untuk transkripsi pasca-panggilan dan waktu nyata. Jika Anda menyalin file yang terletak di Amazon S3 ember, Anda melakukan transkripsi pasca-panggilan. Jika Anda menyalin aliran audio, Anda melakukan transkripsi waktu nyata. Kedua metode transkripsi ini menawarkan wawasan dan fitur Call Analytics yang berbeda. Untuk detail lebih lanjut tentang setiap metode, lihat Analisis pasca-panggilan danAnalisis Panggilan Waktu Nyata.

Dengan transkripsi Analytics Panggilan real-time, Anda juga dapat menyertakan analitik pasca-panggilan dalam permintaan Anda. Transkrip analitik pasca-panggilan Anda disimpan di Amazon S3 bucket yang Anda tentukan dalam permintaan Anda. Lihat Analisis pasca-panggilan dengan transkripsi waktu nyata untuk informasi lebih lanjut.

> Operasi API khusus untuk Call Analytics
>
> Pasca-panggilan: `CreateCallAnalyticsCategory`DeleteCallAnalyticsCategory, `DeleteCallAnalyticsJob`, `GetCallAnalyticsCategory`,, `GetCallAnalyticsJob`, `ListCallAnalyticsCategories`, `ListCallAnalyticsJobs`, `StartCallAnalyticsJob`UpdateCallAnalyticsCategory
> Waktu nyata: `StartCallAnalyticsStreamTranscription`, StartCallAnalyticsStreamTranscriptionWebSocket

Kasus penggunaan umum

Transkripsi pasca-panggilan:

- Pantau frekuensi masalah dari waktu ke waktu: Gunakan kategorisasi panggilan untuk mengidentifikasi kata kunci berulang dalam transkrip Anda.

- Dapatkan wawasan tentang pengalaman layanan pelanggan Anda: Gunakan karakteristik panggilan (waktu non-bicara, waktu bicara, interupsi, kenyaringan suara, kecepatan bicara) dan analisis sentimen untuk menentukan apakah masalah pelanggan diselesaikan dengan tepat selama panggilan.
- Pastikan kepatuhan terhadap peraturan atau kepatuhan terhadap kebijakan perusahaan: Tetapkan kata kunci dan frasa untuk salam atau penafian khusus perusahaan untuk memverifikasi bahwa agen Anda memenuhi persyaratan peraturan.
- Tingkatkan penanganan data pribadi pelanggan: Gunakan redaksi PII dalam output transkripsi atau file audio Anda untuk membantu melindungi privasi pelanggan.
- Tingkatkan pelatihan staf: Gunakan kriteria (sentimen, waktu non-bicara, interupsi, kecepatan bicara) untuk menandai transkrip yang dapat digunakan sebagai contoh interaksi pelanggan positif atau negatif.
- Ukur kemanjuran staf dalam menciptakan pengalaman pelanggan yang positif: Gunakan analisis sentimen untuk mengukur apakah agen Anda dapat mengubah sentimen pelanggan negatif menjadi positif seiring kemajuan panggilan.
- Tingkatkan organisasi data: Beri label dan urutkan panggilan berdasarkan kategori khusus (termasuk kata kunci dan frasa, sentimen, waktu bicara, dan interupsi).
- Ringkas aspek penting dari panggilan menggunakan Generative AI: Gunakan ringkasan panggilan generatif untuk mendapatkan ringkasan ringkas transkrip, yang mencakup komponen kunci seperti masalah, item tindakan, dan hasil yang dibahas dalam panggilan.

Transkripsi waktu nyata:

- Mengurangi eskalasi secara real time: Siapkan peringatan waktu nyata untuk frasa kunci — seperti pelanggan yang mengatakan “berbicara dengan manajer” — untuk menandai panggilan saat mereka mulai meningkat. Anda dapat membuat peringatan waktu nyata menggunakan kecocokan kategori waktu nyata.
- Tingkatkan penanganan data pelanggan: Gunakan identifikasi PII atau redaksi PII dalam output transkripsi Anda untuk membantu melindungi privasi pelanggan.
- Identifikasi kata kunci dan frasa khusus: Gunakan kategori khusus untuk menandai kata kunci tertentu dalam panggilan.
- Identifikasi masalah secara otomatis: Gunakan deteksi masalah otomatis untuk mendapatkan ringkasan singkat dari semua masalah yang diidentifikasi dalam panggilan.

- Ukur kemanjuran staf dalam menciptakan pengalaman pelanggan yang positif: Gunakan analisis sentimen untuk mengukur apakah agen Anda dapat mengubah sentimen pelanggan negatif menjadi positif seiring kemajuan panggilan.
- Siapkan bantuan agen: Gunakan wawasan pilihan Anda untuk memberikan bantuan proaktif kepada agen Anda dalam menyelesaikan panggilan pelanggan. Lihat Live Call Analytics dan bantuan agen untuk pusat kontak Anda dengan layanan AI bahasa Amazon untuk informasi selengkapnya.

Untuk membandingkan fitur yang tersedia dengan Call Analytics dengan fitur for Amazon Transcribe dan Amazon Transcribe Medical, lihat tabel fitur.

Untuk memulai, lihat Memulai transkripsi analitik pasca-panggilan danMemulai transkripsi Analisis Panggilan waktu-nyata. Output Call Analytics mirip dengan pekerjaan transkripsi standar, tetapi berisi data analitik tambahan. Untuk melihat output sampel, lihat Output analitik pasca-panggilan danOutput Analisis Panggilan Waktu Nyata.

Pertimbangan dan informasi tambahan

Sebelum menggunakan Call Analytics, perhatikan bahwa:

- Call Analytics hanya mendukung audio dua saluran, di mana agen hadir di satu saluran dan pelanggan hadir di saluran kedua.
- Antrian pekerjaanselalu diaktifkan untuk pekerjaan analitik pasca-panggilan, jadi Anda dibatasi pada 100 pekerjaan Analytics Panggilan bersamaan. Jika Anda ingin meminta kenaikan kuota, lihat kuota AWS layanan.
- File input untuk pekerjaan analitik pasca-panggilan tidak boleh lebih dari 500 MB dan harus kurang dari 4 jam. Perhatikan bahwa batas ukuran file mungkin lebih kecil untuk format file audio non-WAV terkompresi tertentu.
- Jika menggunakan kategori, Anda harus membuat semua kategori yang diinginkan sebelum memulai transkripsi Call Analytics. Setiap kategori baru tidak dapat diterapkan pada transkripsi yang ada. Untuk mempelajari cara membuat kategori baru, lihat Membuat kategori untuk transkripsi pasca-panggilan danMembuat kategori untuk transkripsi waktu nyata.
- Beberapa kuota Call Analytics berbeda dari Amazon Transcribe dan Amazon Transcribe Medis; lihat Referensi AWS Umum untuk detailnya.

Menyelam lebih dalam dengan Blog AWS Machine Learning

Untuk mempelajari lebih lanjut tentang opsi Call Analytics, lihat:

- Posting Analisis Panggilan untuk pusat kontak Anda dengan layanan AI bahasa Amazon
- Live Call Analytics dan bantuan agen untuk pusat kontak Anda dengan layanan AI bahasa Amazon

Untuk melihat contoh keluaran dan fitur Call Analytics, lihat GitHubdemo kami. Kami juga menawarkan aplikasi dokumen JSON ke Word untuk mengonversi transkrip Anda menjadi easy-to-read format.

Ketersediaan dan kuota wilayah

Call Analytics didukung sebagai berikut Wilayah AWS:

Wilayah	Jenis transkripsi
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 (Singapura)	post-call
ap-southeast-2 (Sydney)	post-call, real-time
ca-central-1 (Kanada, Tengah)	post-call, real-time
eu-central-1 (Frankfurt am Main)	post-call, real-time
eu-west-2 (London)	post-call, real-time
us-east-1 (Virginia Utara)	post-call, real-time
us-west-2 (Oregon)	post-call, real-time

Perhatikan bahwa dukungan Wilayah berbeda untuk Amazon TranscribeAmazon Transcribe Medical, dan Call Analytics.

Untuk mendapatkan titik akhir untuk setiap Wilayah yang didukung, lihat Titik akhir Layanan di Referensi AWS Umum.

Untuk daftar kuota yang berkaitan dengan transkripsi Anda, lihat kuota Layanan di Referensi Umum.AWS Beberapa kuota dapat diubah berdasarkan permintaan. Jika kolom Adjustable berisi 'Ya', Anda dapat meminta peningkatan. Untuk melakukannya, pilih tautan yang disediakan.

Analisis pasca-panggilan

Call Analytics menyediakan analisis pasca-panggilan, yang berguna untuk memantau tren layanan pelanggan.

Transkripsi pasca-panggilan menawarkan wawasan berikut:

- Karakteristik panggilan, termasuk waktu bicara, waktu non-bicara, kenyaringan speaker, interupsi, kecepatan bicara, masalah, hasil, dan item tindakan
- Ringkasan panggilan generatif, yang membuat ringkasan ringkas dari seluruh panggilan
- Kategorisasi kustom dengan aturan yang dapat Anda gunakan untuk mengasah kata kunci dan kriteria tertentu
- Redaksi PII dari transkrip teks dan file audio Anda
- Sentimen pembicara untuk setiap penelepon di berbagai titik dalam panggilan

Wawasan pasca-panggilan

Bagian ini merinci wawasan yang tersedia untuk transkripsi analitik pasca-panggilan.

Karakteristik panggilan

Fitur karakteristik panggilan mengukur kualitas interaksi agen-pelanggan menggunakan kriteria berikut:

- Interupsi: Mengukur jika dan ketika satu peserta memotong peserta lain di tengah kalimat. Gangguan yang sering dapat dikaitkan dengan kekasaran atau kemarahan, dan dapat berkorelasi dengan sentimen negatif untuk satu atau kedua peserta.

- Kenyaringan: Mengukur volume di mana setiap peserta berbicara. Gunakan metrik ini untuk melihat apakah penelepon atau agen berbicara keras atau berteriak, yang sering menunjukkan kesal. Metrik ini direpresentasikan sebagai nilai yang dinormalisasi (tingkat ucapan per detik ucapan di segmen tertentu) pada skala dari 0 hingga 100, di mana nilai yang lebih tinggi menunjukkan suara yang lebih keras.
- Waktu non-bicara: Mengukur periode waktu yang tidak mengandung ucapan. Gunakan metrik ini untuk melihat apakah ada periode keheningan yang lama, seperti agen yang menahan pelanggan untuk waktu yang berlebihan.
- Kecepatan bicara: Mengukur kecepatan di mana kedua peserta berbicara. Pemahaman dapat terpengaruh jika satu peserta berbicara terlalu cepat. Metrik ini diukur dalam kata-kata per menit.
- Waktu bicara: Mengukur jumlah waktu (dalam milidetik) setiap peserta berbicara selama panggilan. Gunakan metrik ini untuk membantu mengidentifikasi apakah satu peserta mendominasi panggilan atau jika dialog seimbang.
- Masalah, Hasil, dan Item Tindakan: Mengidentifikasi masalah, hasil, dan item tindakan dari transkrip panggilan.

Berikut adalah contoh output.

Ringkasan panggilan generatif

Ringkasan panggilan generatif membuat ringkasan ringkas dari seluruh panggilan, menangkap komponen utama seperti alasan panggilan, langkah-langkah yang diambil untuk menyelesaikan masalah, dan langkah selanjutnya.

Menggunakan ringkasan panggilan generatif, Anda dapat:

- Kurangi kebutuhan akan pencatatan manual selama dan setelah panggilan.
- Tingkatkan efisiensi agen karena mereka dapat menghabiskan lebih banyak waktu berbicara dengan penelepon yang menunggu dalam antrian daripada terlibat dalam pekerjaan setelah panggilan.
- Mempercepat ulasan supervisor karena ringkasan panggilan jauh lebih cepat untuk ditinjau daripada seluruh transkrip.

Untuk menggunakan ringkasan panggilan generatif dengan pekerjaan analitik pasca-panggilan, lihat Mengaktifkan ringkasan panggilan generatif. Misalnya keluaran, lihat Contoh keluaran ringkasan

panggilan generatif. Ringkasan panggilan generatif diberi harga terpisah (silakan lihat halaman harga).

> **Note**
>
> Ringkasan panggilan generatif saat ini tersedia di `us-east-1` dan. `us-west-2` Kemampuan ini didukung dengan dialek bahasa Inggris ini: Australia (`en-AU`), Inggris (), India (`en-GB`), Irlandia (`en-INen-IE`), Skotlandia (`en-AB`), AS (`en-US`), dan Welsh (). `en-WL`

Kategorisasi kustom

Gunakan kategorisasi panggilan untuk menandai kata kunci, frasa, sentimen, atau tindakan dalam panggilan. Opsi kategorisasi kami dapat membantu Anda melakukan triase eskalasi, seperti panggilan sentimen negatif dengan banyak gangguan, atau mengatur panggilan ke dalam kategori tertentu, seperti departemen perusahaan.

Kriteria yang dapat Anda tambahkan ke kategori meliputi:

- Waktu non-bicara: Periode waktu ketika pelanggan maupun agen tidak berbicara.
- Interupsi: Ketika pelanggan atau agen mengganggu orang lain.
- Sentimen pelanggan atau agen: Bagaimana perasaan pelanggan atau agen selama periode waktu tertentu. Jika setidaknya 50 persen percakapan berubah (back-and-forthantara dua pembicara) dalam periode waktu tertentu sesuai dengan sentimen yang ditentukan, Amazon Transcribe anggap sentimen itu cocok.
- Kata kunci atau frasa: Cocokkan bagian transkripsi berdasarkan frasa yang tepat. Misalnya, jika Anda menyetel filter untuk frasa “Saya ingin berbicara dengan manajer”, Amazon Transcribe filter untuk frasa yang tepat itu.

Anda juga dapat menandai kebalikan dari kriteria sebelumnya (waktu bicara, kurangnya interupsi, sentimen tidak hadir, dan kurangnya frasa tertentu).

Berikut adalah contoh output.

Untuk informasi selengkapnya tentang kategori atau mempelajari cara membuat kategori baru, lihatMembuat kategori untuk transkripsi pasca-panggilan.

Redaksi data sensitif

Redaksi data sensitif menggantikan informasi yang dapat diidentifikasi secara pribadi (PII) dalam transkrip teks dan file audio. Transkrip yang disunting menggantikan teks asli dengan`[PII]`; file audio yang disunting menggantikan informasi pribadi yang diucapkan dengan keheningan. Parameter ini berguna untuk melindungi informasi pelanggan.

> **Note**
>
> Redaksi PII pasca-panggilan didukung dengan US English (`en-US`) dan US Spanish (). `es-US`

Untuk melihat daftar PII yang disunting menggunakan fitur ini, atau untuk mempelajari lebih lanjut tentang redaksi dengan Amazon Transcribe, lihat. Menyunting atau mengidentifikasi informasi yang dapat diidentifikasi secara pribadi

Berikut adalah contoh keluaran.

Analisis sentimen

Analisis sentimen memperkirakan bagaimana perasaan pelanggan dan agen selama panggilan berlangsung. Metrik ini direpresentasikan sebagai nilai kuantitatif (dengan rentang dari `5` ke`-5`) dan nilai kualitatif (`positive`,, `neutral`mixed, atau`negative`). Nilai kuantitatif disediakan per kuartal dan per panggilan; nilai kualitatif disediakan per giliran.

Metrik ini dapat membantu mengidentifikasi apakah agen Anda dapat menyenangkan pelanggan yang kesal pada saat panggilan berakhir.

Analisis sentimen berfungsi out-of-the-box dan karenanya tidak mendukung penyesuaian, seperti pelatihan model atau kategori khusus.

Berikut adalah contoh output.

Membuat kategori untuk transkripsi pasca-panggilan

Analisis pasca-panggilan mendukung pembuatan kategori khusus, memungkinkan Anda menyesuaikan analisis transkrip agar sesuai dengan kebutuhan bisnis spesifik Anda.

Anda dapat membuat kategori sebanyak yang Anda suka untuk mencakup berbagai skenario yang berbeda. Untuk setiap kategori yang Anda buat, Anda harus membuat antara 1 dan 20

aturan. Setiap aturan didasarkan pada salah satu dari empat kriteria: interupsi, kata kunci, waktu non-bicara, atau sentimen. Untuk detail lebih lanjut tentang penggunaan kriteria ini dengan `CreateCallAnalyticsCategory`operasi, lihat Kriteria aturan untuk kategori analitik pasca-panggilan bagian.

Jika konten di media Anda cocok dengan semua aturan yang Anda tentukan dalam kategori tertentu, beri Amazon Transcribe label output Anda dengan kategori tersebut. Lihat keluaran kategorisasi panggilan untuk contoh kecocokan kategori dalam keluaran JSON.

Berikut adalah beberapa contoh dari apa yang dapat Anda lakukan dengan kategori khusus:

- Mengisolasi panggilan dengan karakteristik tertentu, seperti panggilan yang diakhiri dengan sentimen pelanggan negatif
- Identifikasi tren dalam masalah pelanggan dengan menandai dan melacak set kata kunci tertentu
- Pantau kepatuhan, seperti agen yang berbicara (atau menghilangkan) frasa tertentu selama beberapa detik pertama panggilan
- Dapatkan wawasan tentang pengalaman pelanggan dengan menandai panggilan dengan banyak gangguan agen dan sentimen pelanggan negatif
- Bandingkan beberapa kategori untuk mengukur korelasi, seperti menganalisis apakah agen yang menggunakan frase selamat datang berkorelasi dengan sentimen pelanggan yang positif

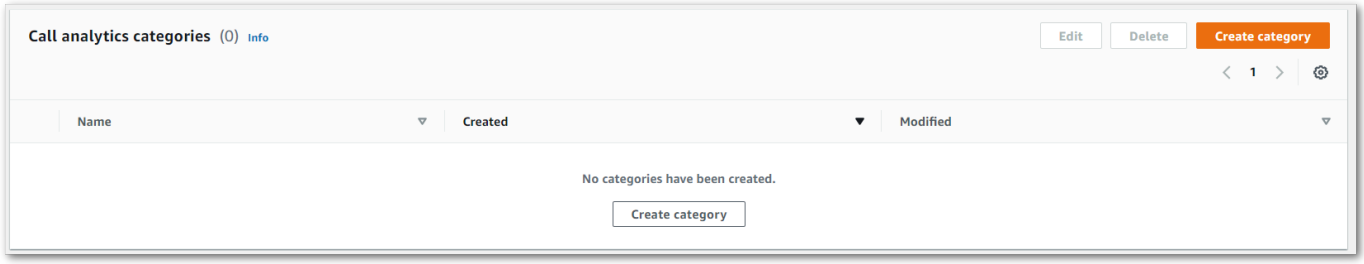
Kategori pasca-panggilan versus waktu nyata

Saat membuat kategori baru, Anda dapat menentukan apakah Anda ingin kategori tersebut dibuat sebagai kategori analitik pasca-panggilan (`POST_CALL`) atau sebagai kategori Analisis Panggilan real-time (`REAL_TIME`). Jika Anda tidak menentukan opsi, kategori Anda dibuat sebagai kategori pasca-panggilan secara default. Pencocokan kategori analitik pasca-panggilan tersedia di output Anda setelah menyelesaikan transkripsi analitik pasca-panggilan Anda.

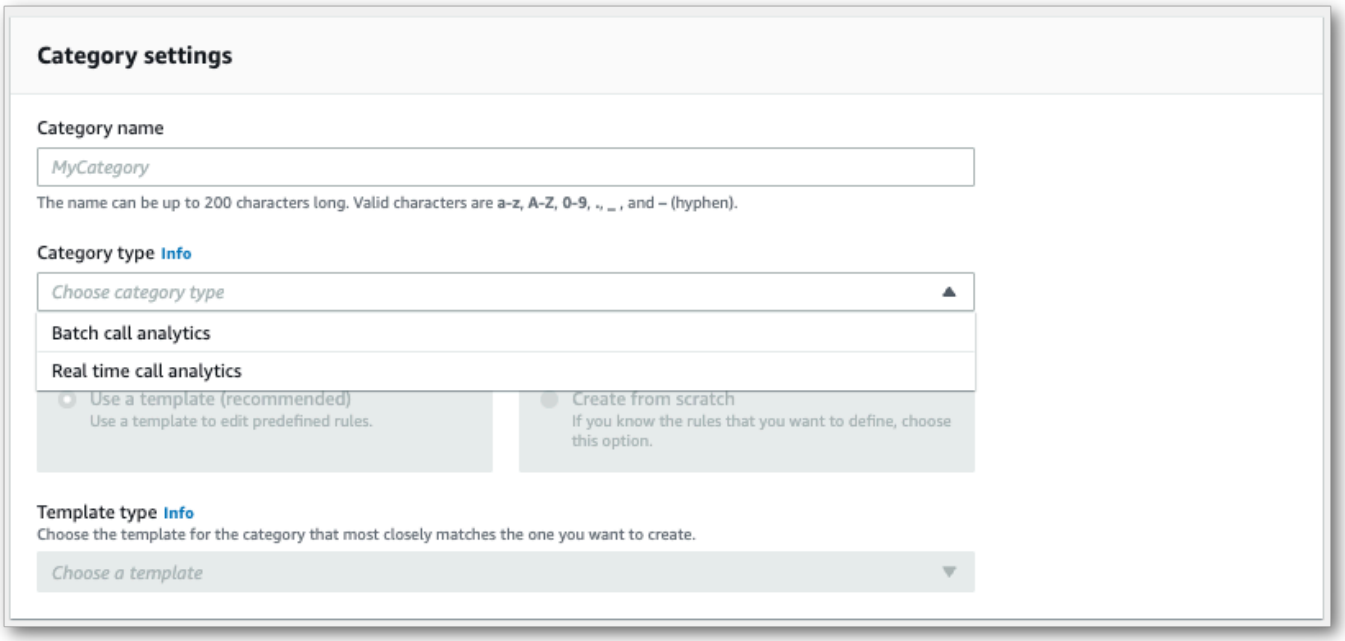
Untuk membuat kategori baru untuk analitik pasca-panggilan, Anda dapat menggunakan AWS Management ConsoleAWS CLI,, atau AWS SDKs; lihat contoh berikut:

AWS Management Console

1. Di panel navigasi, di bawah Amazon Transcribe, pilih Amazon Transcribe Call Analytics.
2. Pilih kategori analisis Panggilan, yang akan membawa Anda ke halaman kategori analisis Panggilan. Pilih Buat kategori.

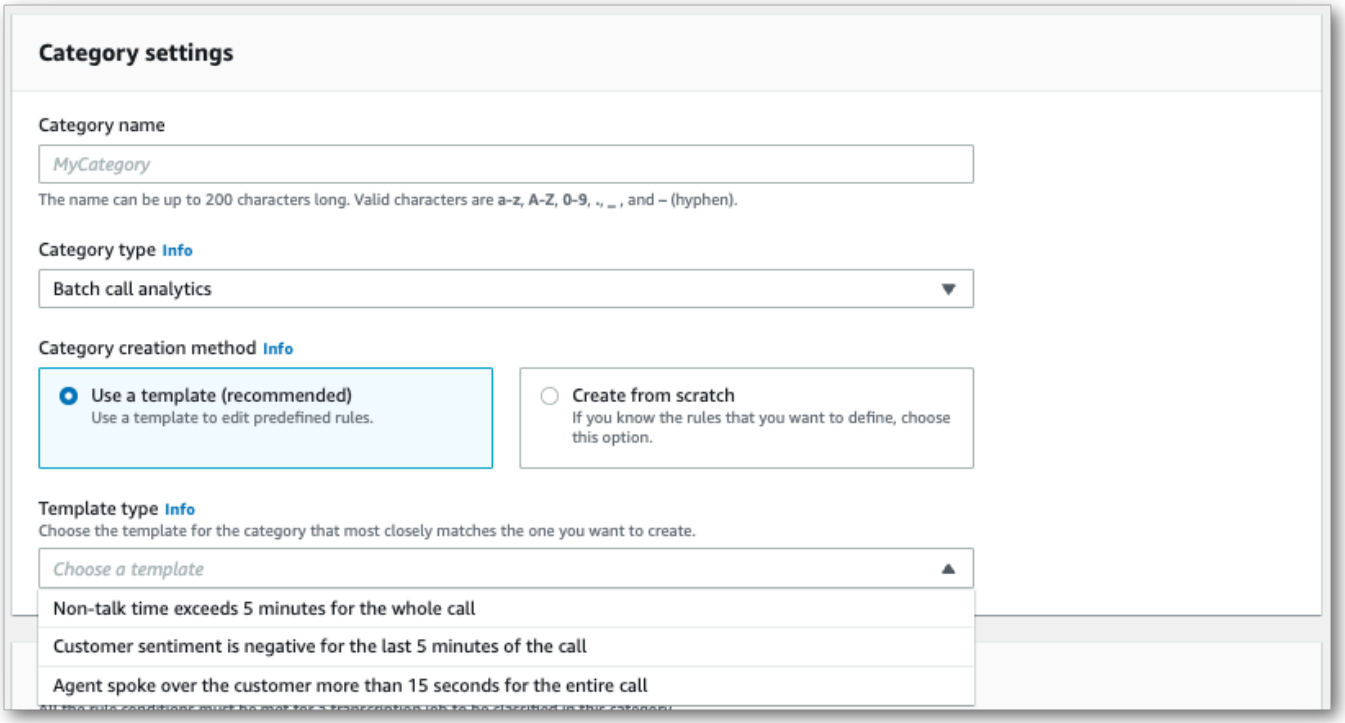


3. Anda sekarang berada di halaman kategori Buat. Masukkan nama untuk kategori Anda, lalu pilih 'Analisis panggilan batch' di menu tarik-turun tipe Kategori.

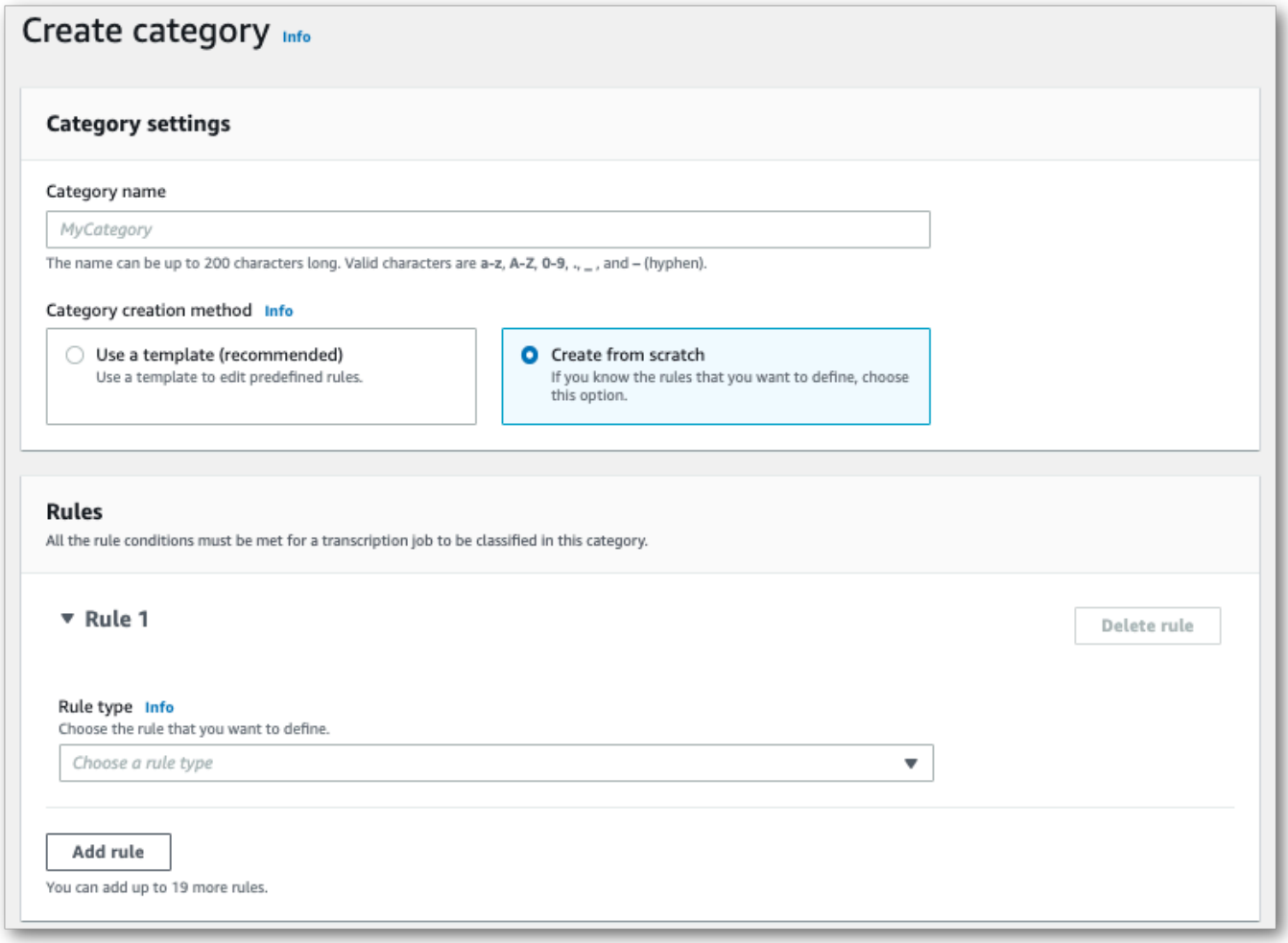


4. Anda dapat memilih template untuk membuat kategori Anda atau Anda dapat membuatnya dari awal.

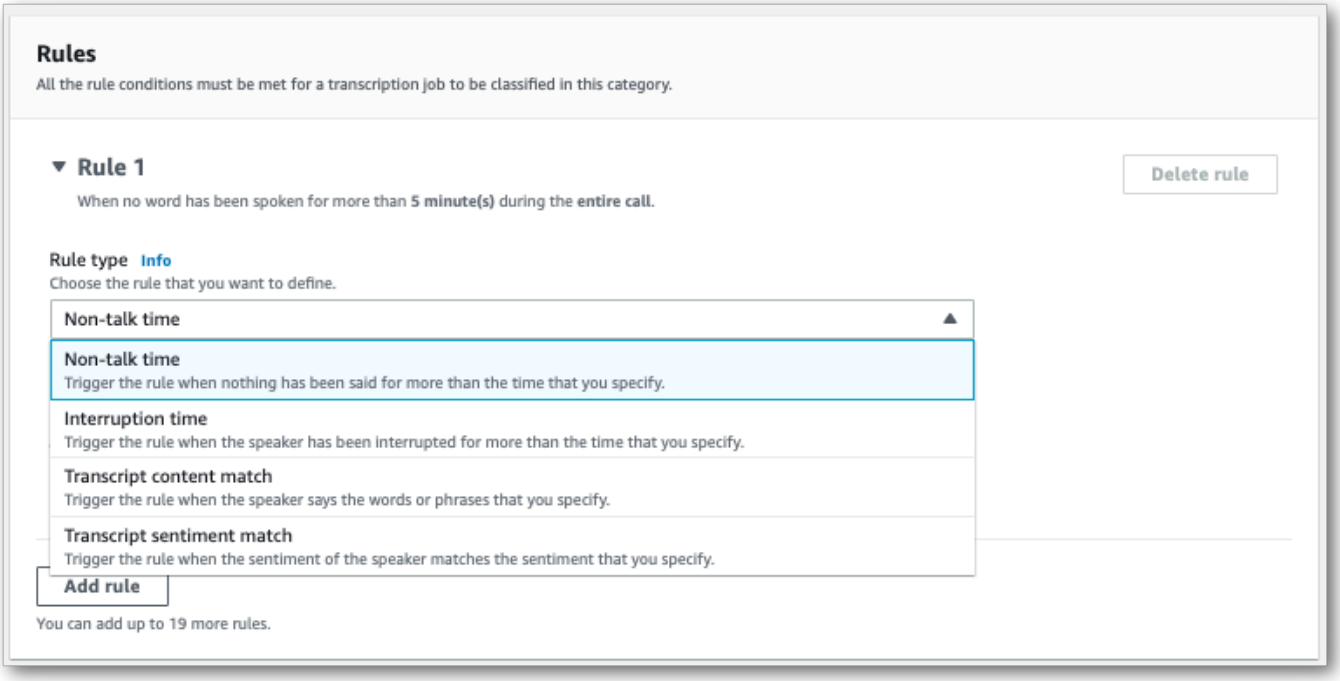
   Jika menggunakan templat: pilih Gunakan templat (disarankan), pilih templat yang Anda inginkan, lalu pilih Buat kategori.



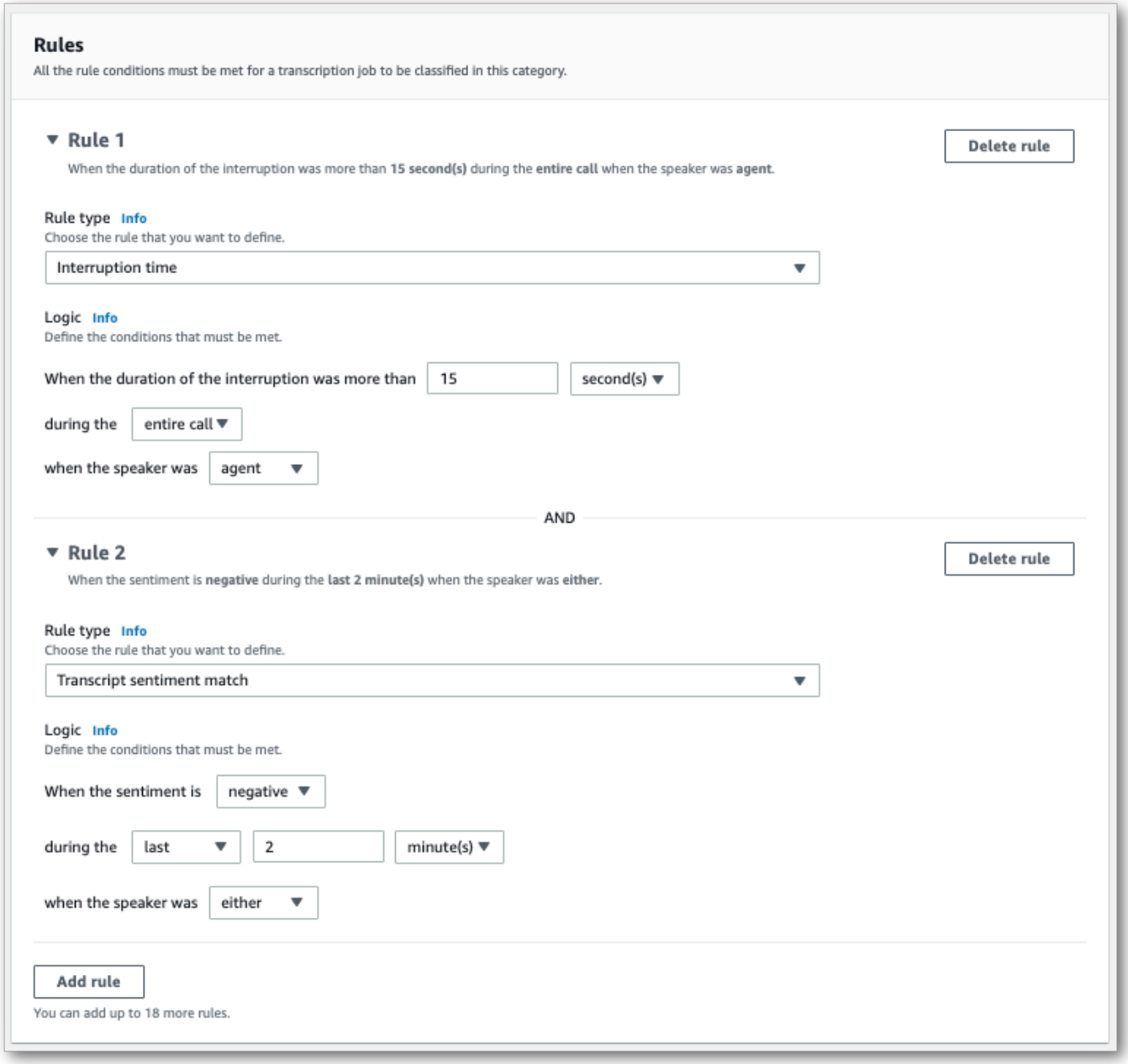
5. Jika membuat kategori kustom: pilih Buat dari awal.



6. Tambahkan aturan ke kategori Anda menggunakan menu tarik-turun. Anda dapat menambahkan hingga 20 aturan per kategori.



7. Berikut adalah contoh kategori dengan dua aturan: agen yang mengganggu pelanggan selama lebih dari 15 detik selama panggilan dan sentimen negatif yang dirasakan oleh pelanggan atau agen dalam dua menit terakhir panggilan.



8. Setelah selesai menambahkan aturan ke kategori, pilih Buat kategori.

AWS CLI

Contoh ini menggunakan create-call-analytics-categoryperintah. Lihat informasi selengkapnya di `CreateCallAnalyticsCategory`, `CategoryProperties`, dan `Rule`.

Contoh berikut membuat kategori dengan aturan:

- Pelanggan terganggu dalam 60.000 milidetik pertama. Durasi interupsi ini berlangsung setidaknya 10.000 milidetik.
- Ada periode hening yang berlangsung setidaknya 20.000 milidetik antara 10% dalam panggilan dan 80% ke dalam panggilan.
- Agen memiliki sentimen negatif di beberapa titik dalam panggilan.
- Kata “selamat datang” atau “halo” tidak digunakan dalam 10.000 milidetik pertama panggilan.

Contoh ini menggunakan create-call-analytics-categoryperintah, dan badan permintaan yang menambahkan beberapa aturan ke kategori Anda.

```
aws transcribe create-call-analytics-category \
--cli-input-json file://filepath/my-first-analytics-category.json
```

File my-first-analytics-category.json berisi badan permintaan berikut.

```
{
  "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 untuk Python (Boto3)

Contoh ini menggunakan AWS SDK untuk Python (Boto3) untuk membuat kategori menggunakan `CategoryName` dan `Rules` argumen untuk metode create_call_analytics_category. Lihat informasi selengkapnya di `CreateCallAnalyticsCategory`, `CategoryProperties`, dan `Rule`.

Untuk contoh tambahan menggunakan AWS SDKs, termasuk contoh khusus fitur, skenario, dan lintas layanan, lihat bagian ini. Contoh kode untuk Amazon Transcribe menggunakan AWS SDKs

Contoh berikut membuat kategori dengan aturan:

- Pelanggan terganggu dalam 60.000 milidetik pertama. Durasi interupsi ini berlangsung setidaknya 10.000 milidetik.
- Ada periode hening yang berlangsung setidaknya 20.000 milidetik antara 10% dalam panggilan dan 80% ke dalam panggilan.
- Agen memiliki sentimen negatif di beberapa titik dalam panggilan.
- Kata “selamat datang” atau “halo” tidak digunakan dalam 10.000 milidetik pertama panggilan.

```
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)
```

Kriteria aturan untuk kategori analitik pasca-panggilan

Bagian ini menguraikan jenis `POST_CALL` aturan kustom yang dapat Anda buat menggunakan operasi `CreateCallAnalyticsCategory`API.

Pertandingan interupsi

Aturan yang menggunakan interupsi (tipe `InterruptionFilter`data) dirancang agar sesuai:

- Contoh di mana agen mengganggu pelanggan
- Contoh di mana pelanggan mengganggu agen
- Setiap peserta menginterupsi yang lain
- Kurangnya interupsi

Berikut adalah contoh parameter yang tersedia dengan `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
},
```

Lihat CreateCallAnalyticsCategorydan InterruptionFilteruntuk informasi lebih lanjut tentang parameter ini dan nilai valid yang terkait dengan masing-masing parameter.

Pencocokan kata kunci

Aturan menggunakan kata kunci (tipe TranscriptFilterdata) dirancang untuk mencocokkan:

- Kata atau frasa khusus yang diucapkan oleh agen, pelanggan, atau keduanya
- Kata atau frasa khusus yang tidak diucapkan oleh agen, pelanggan, atau keduanya
- Kata atau frasa khusus yang terjadi dalam kerangka waktu tertentu

Berikut adalah contoh parameter yang tersedia dengan 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
}
```

Lihat CreateCallAnalyticsCategorydan TranscriptFilteruntuk informasi lebih lanjut tentang parameter ini dan nilai valid yang terkait dengan masing-masing parameter.

Pertandingan waktu non-bicara

Aturan yang menggunakan waktu non-bicara (tipe NonTalkTimeFilterdata) dirancang agar sesuai:

- Kehadiran keheningan pada periode tertentu selama panggilan
- Kehadiran pidato pada periode tertentu selama panggilan

Berikut adalah contoh parameter yang tersedia dengan 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
},
```

Lihat CreateCallAnalyticsCategorydan NonTalkTimeFilteruntuk informasi lebih lanjut tentang parameter ini dan nilai valid yang terkait dengan masing-masing parameter.

Pertandingan sentimen

Aturan yang menggunakan sentimen (tipe SentimentFilterdata) dirancang agar sesuai:

- Ada atau tidaknya sentimen positif yang diungkapkan oleh pelanggan, agen, atau keduanya pada titik-titik tertentu dalam panggilan
- Ada atau tidaknya sentimen negatif yang diungkapkan oleh pelanggan, agen, atau keduanya pada titik-titik tertentu dalam panggilan
- Ada atau tidaknya sentimen netral yang diungkapkan oleh pelanggan, agen, atau keduanya pada titik-titik tertentu dalam panggilan
- Ada atau tidaknya sentimen campuran yang diungkapkan oleh pelanggan, agen, atau keduanya pada titik-titik tertentu dalam panggilan

Berikut adalah contoh parameter yang tersedia dengan 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 ]
},
```

Lihat `CreateCallAnalyticsCategory`dan `SentimentFilter`untuk informasi lebih lanjut tentang parameter ini dan nilai valid yang terkait dengan masing-masing parameter.

Memulai transkripsi analitik pasca-panggilan

Sebelum memulai transkripsi analitik pasca-panggilan, Anda harus membuat semua kategori yang Amazon Transcribe ingin Anda cocokkan dalam audio Anda.

> **Note**
>
> Transkrip Call Analytics tidak dapat dicocokkan secara surut dengan kategori baru. Hanya kategori yang Anda buat sebelum memulai transkripsi Call Analytics yang dapat diterapkan ke output transkripsi tersebut.

Jika Anda telah membuat satu atau beberapa kategori, dan audio Anda cocok dengan semua aturan dalam setidaknya satu kategori, Amazon Transcribe tandai output Anda dengan kategori yang cocok. Jika Anda memilih untuk tidak menggunakan kategori, atau jika audio Anda tidak cocok dengan aturan yang ditentukan dalam kategori Anda, transkrip Anda tidak ditandai.

Untuk memulai transkripsi analitik pasca-panggilan, Anda dapat menggunakan AWS Management Console, AWS CLI, atau AWS SDKs; lihat contoh berikut ini:

AWS Management Console

Gunakan prosedur berikut untuk memulai pekerjaan analitik pasca-panggilan. Panggilan yang cocok dengan semua karakteristik yang ditentukan oleh kategori diberi label dengan kategori tersebut.

1. Di panel navigasi, di bawah Amazon Transcribe Call Analytics, pilih Pekerjaan analitik panggilan.
2. Pilih Buat tugas.

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

Search

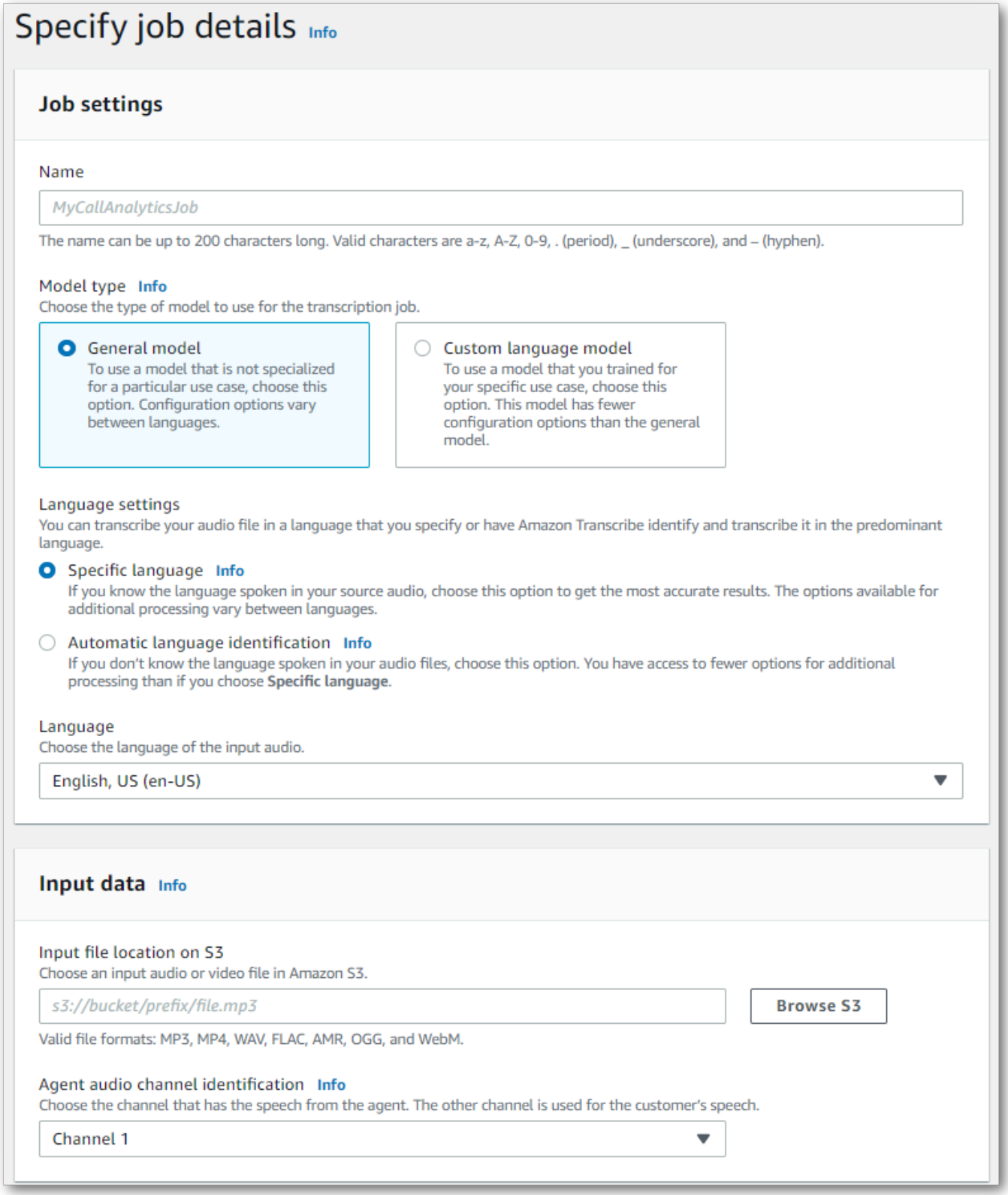
1

	Name	Type	Created	Modified
○	CatchNegativeSentiment	POST_CALL	February 17 2023, 10:43 (UTC-08:00)	February 17 2023, 10:43 (UTC-08:00)

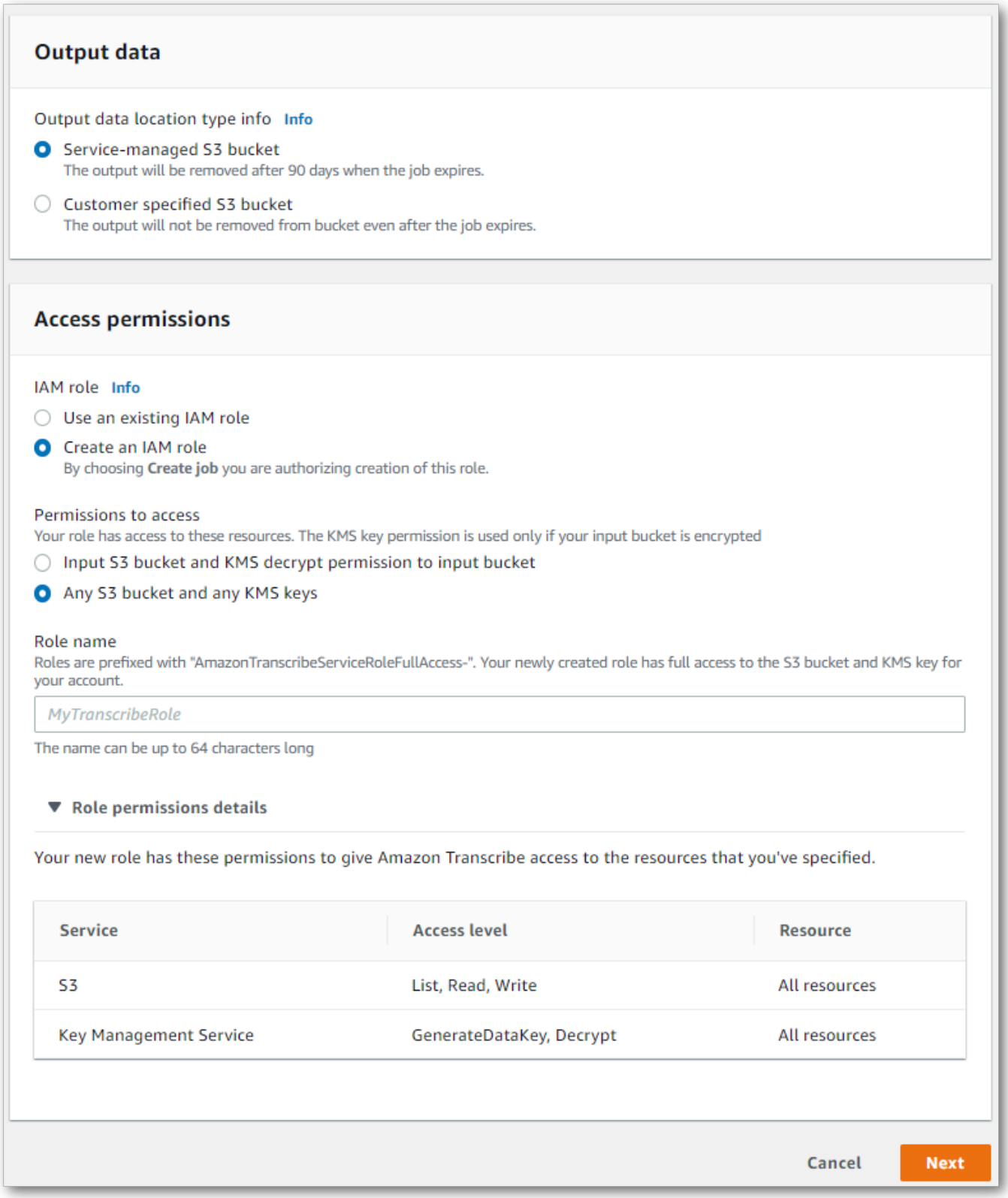
If the above categories aren't relevant to your use case, you can create a new category. Create a new category.

Cancel Previous Create job

3. Pada halaman Tentukan detail pekerjaan, berikan informasi tentang pekerjaan Analytics Panggilan Anda, termasuk lokasi data masukan Anda.



Tentukan Amazon S3 lokasi yang diinginkan dari data keluaran Anda dan IAM peran mana yang akan digunakan.



4. Pilih Berikutnya.

5. Untuk pekerjaan Konfigurasi, aktifkan fitur opsional apa pun yang ingin Anda sertakan dengan pekerjaan Analytics Panggilan Anda. Jika sebelumnya Anda membuat kategori, kategori tersebut muncul di panel Kategori dan secara otomatis diterapkan ke pekerjaan Analytics Panggilan Anda.

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

Search

1

	Name	Type	Created	Modified
○	CatchNegativeSentiment	POST_CALL	February 17 2023, 10:43 (UTC-08:00)	February 17 2023, 10:43 (UTC-08:00)

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

Cancel Previous Create job

6. Pilih Buat tugas.

AWS CLI

Contoh ini menggunakan start-call-analytics-jobperintah dan `channel-definitions` parameter. Untuk informasi selengkapnya, lihat `StartCallAnalyticsJob` dan `ChannelDefinition`.

```
aws transcribe start-call-analytics-job \
--region us-west-2 \
--call-analytics-job-name my-first-call-analytics-job \
--media MediaFileUri=s3://amzn-s3-demo-bucket/my-input-files/my-media-file.flac \
--output-location s3://amzn-s3-demo-bucket/my-output-files/ \
--data-access-role-arn arn:aws:iam::111122223333:role/ExampleRole \
--channel-definitions ChannelId=0,ParticipantRole=AGENT
 ChannelId=1,ParticipantRole=CUSTOMER
```

Berikut contoh lain menggunakan start-call-analytics-jobperintah, dan badan permintaan yang memungkinkan Call Analytics untuk pekerjaan itu.

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

File my-call-analytics-job.json berisi badan permintaan berikut.

```
{
      "CallAnalyticsJobName": "my-first-call-analytics-job",
      "DataAccessRoleArn": "arn:aws:iam::111122223333:role/ExampleRole",
      "Media": {
          "MediaFileUri": "s3://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
      },
      "OutputLocation": "s3://amzn-s3-demo-bucket/my-output-files/",
      "ChannelDefinitions": [
          {
              "ChannelId": 0,
              "ParticipantRole": "AGENT"
          },
          {
              "ChannelId": 1,
              "ParticipantRole": "CUSTOMER"
          }
```

```
      ]
}
```

AWS SDK untuk Python (Boto3)

Contoh ini menggunakan AWS SDK untuk Python (Boto3) untuk memulai pekerjaan Call Analytics menggunakan metode start_call_analytics_job. Untuk informasi selengkapnya, lihat StartCallAnalyticsJob dan ChannelDefinition.

Untuk contoh tambahan menggunakan AWS SDKs, termasuk contoh khusus fitur, skenario, dan lintas layanan, lihat bagian ini. Contoh kode untuk Amazon Transcribe menggunakan AWS SDKs

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

 while True:
   status = transcribe.get_call_analytics_job(CallAnalyticsJobName = job_name)
   if status['CallAnalyticsJob']['CallAnalyticsJobStatus'] in ['COMPLETED', 'FAILED']:
     break
```

```
    print("Not ready yet...")
    time.sleep(5)
print(status)
```

Output analitik pasca-panggilan

Transkrip analitik pasca-panggilan ditampilkan dalam turn-by-turn format berdasarkan segmen. Mereka termasuk kategorisasi panggilan, karakteristik panggilan (skor kenyaringan, interupsi, waktu non-bicara, kecepatan bicara), ringkasan panggilan (masalah, hasil, dan item tindakan), redaksi, dan sentimen. Selain itu, ringkasan karakteristik percakapan disediakan di akhir transkrip.

Untuk meningkatkan akurasi dan lebih lanjut menyesuaikan transkrip Anda dengan kasus penggunaan Anda, seperti menyertakan istilah khusus industri, tambahkan kosakata khusus atau model bahasa khusus ke permintaan Analisis Panggilan Anda. Untuk menutupi, menghapus, atau menandai kata-kata yang tidak Anda inginkan dalam hasil transkripsi Anda, seperti kata-kata kotor, tambahkan pemfilteran kosakata. Jika Anda tidak yakin kode bahasa yang akan diteruskan ke file media, Anda dapat mengaktifkan identifikasi bahasa batch untuk secara otomatis mengidentifikasi bahasa dalam file media Anda.

Bagian berikut menunjukkan contoh output JSON pada tingkat wawasan. Untuk output yang dikompilasi, lihatOutput analitik pasca-panggilan yang dikompilasi.

Kategorisasi panggilan

Inilah yang terlihat seperti kecocokan kategori dalam keluaran transkripsi Anda. Contoh ini menunjukkan bahwa audio dari stempel waktu 40040 milidetik ke stempel waktu 42460 milidetik cocok dengan kategori 'resolusi positif'. Dalam hal ini, kategori 'resolusi positif' khusus membutuhkan sentimen positif dalam beberapa detik terakhir pidato.


```
"Categories": {
    "MatchedDetails": {
        "positive-resolution": {
            "PointsOfInterest": [
                {
                    "BeginOffsetMillis":  40040,
                    "EndOffsetMillis":  42460
                }
            ]
        }
    },
```

```
    "MatchedCategories": [
        " positive-resolution"
    ]
},
```

Karakteristik panggilan

Inilah yang terlihat seperti karakteristik panggilan dalam keluaran transkripsi Anda. Perhatikan bahwa skor kenyaringan disediakan untuk setiap giliran percakapan, sementara semua karakteristik lainnya disediakan di akhir transkrip.

```
"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
    }
},
```

Masalah, Item Tindakan, dan Langkah Selanjutnya

- Dalam contoh berikut, masalah diidentifikasi sebagai mulai dari karakter 7 dan berakhir pada karakter 51, yang mengacu pada bagian teks ini: "Saya ingin membatalkan langganan resep saya”.

```
"Content": "Well, I would like to cancel my recipe subscription.",

"IssuesDetected": [
    {
```

```
        "CharacterOffsets": {
            "Begin": 7,
            "End": 51
        }
    }
],
```

- Dalam contoh berikut, hasil diidentifikasi sebagai mulai dari karakter 12 dan berakhir pada karakter 78, yang mengacu pada bagian teks ini: "Saya membuat semua perubahan pada akun Anda dan sekarang diskon ini diterapkan".

```
"Content": "Wonderful. I made all changes to your account and now this discount is
 applied, please check.",

"OutcomesDetected": [
    {
        "CharacterOffsets": {
            "Begin": 12,
            "End": 78
        }
    }
],
```

- Dalam contoh berikut, item tindakan diidentifikasi sebagai mulai dari karakter 0 dan berakhir pada karakter 103, yang mengacu pada bagian teks ini: "Saya akan mengirim email dengan semua detail kepada Anda hari ini, dan saya akan menelepon Anda kembali minggu depan untuk menindaklanjuti".

```
"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
        }
    }
],
```

Ringkasan panggilan generatif

Inilah yang terlihat seperti ringkasan panggilan generatif dalam keluaran transkripsi Anda:

```
"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."
        }
    }
}
```

Pekerjaan analitik akan selesai tanpa pembuatan ringkasan dalam kasus berikut:

- Konten percakapan tidak mencukupi: Percakapan harus mencakup setidaknya satu putaran dari agen dan pelanggan. Ketika konten percakapan tidak mencukupi, layanan akan mengembalikan kode kesalahan INSUFFICIENT_CONVERSATION_CONTENT.
- Pagar pengaman: Percakapan harus memenuhi pagar pengaman di tempat untuk memastikan ringkasan yang tepat dihasilkan. Ketika pagar pembatas ini tidak terpenuhi, layanan akan mengembalikan kode kesalahan FAILED_SAFETY_GUIDELINES.

Kode kesalahan dapat ditemukan di `Skipped` bagian `AnalyticsJobDetails` dalam output. Anda juga dapat menemukan alasan kesalahan `CallAnalyticsJobDetails` dalam Respons `GetCallAnalyticsJob`API.

Contoh Kesalahan Output

```
{
    "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",
    ...
}
```

Analisis sentimen

Inilah yang terlihat seperti analisis sentimen dalam output transkripsi Anda.

- Nilai turn-by-turn sentimen kualitatif:

```
"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"
```

- Nilai sentimen kuantitatif untuk seluruh panggilan:

```
"Sentiment": {
    "OverallSentiment": {
        "AGENT": 2.5,
        "CUSTOMER": 2.1
    },
```

- Nilai sentimen kuantitatif per peserta dan per kuartal panggilan:

```
"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
            }
        ]
    }
}
```

Redaksi PII

Inilah yang tampak seperti redaksi PII dalam keluaran transkripsi Anda.

```
"Content": "[PII], my name is [PII], how can I help?",
"Redaction": [{
    "Confidence": "0.9998",
    "Type": "NAME",
    "Category": "PII"
}]
```

Untuk informasi selengkapnya, lihat Menyunting PII di pekerjaan batch Anda.

Identifikasi bahasa

Seperti inilah tampilan Identifikasi Bahasa dalam keluaran transkripsi Anda jika fitur tersebut diaktifkan.

```
"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"
}]
```

Dalam contoh keluaran di atas, Identifikasi Bahasa akan mengisi kode bahasa dengan skor kepercayaan. Hasil dengan skor tertinggi akan dipilih sebagai kode bahasa untuk transkripsi. Untuk detail mode lihat Mengidentifikasi bahasa dominan di media Anda.

Output analitik pasca-panggilan yang dikompilasi

Untuk singkatnya, beberapa konten diganti dengan elips dalam keluaran transkripsi berikut.

Sampel ini mencakup fitur opsional - Ringkasan panggilan generatif.


```
{
    "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": []
    },
    ...
}
```

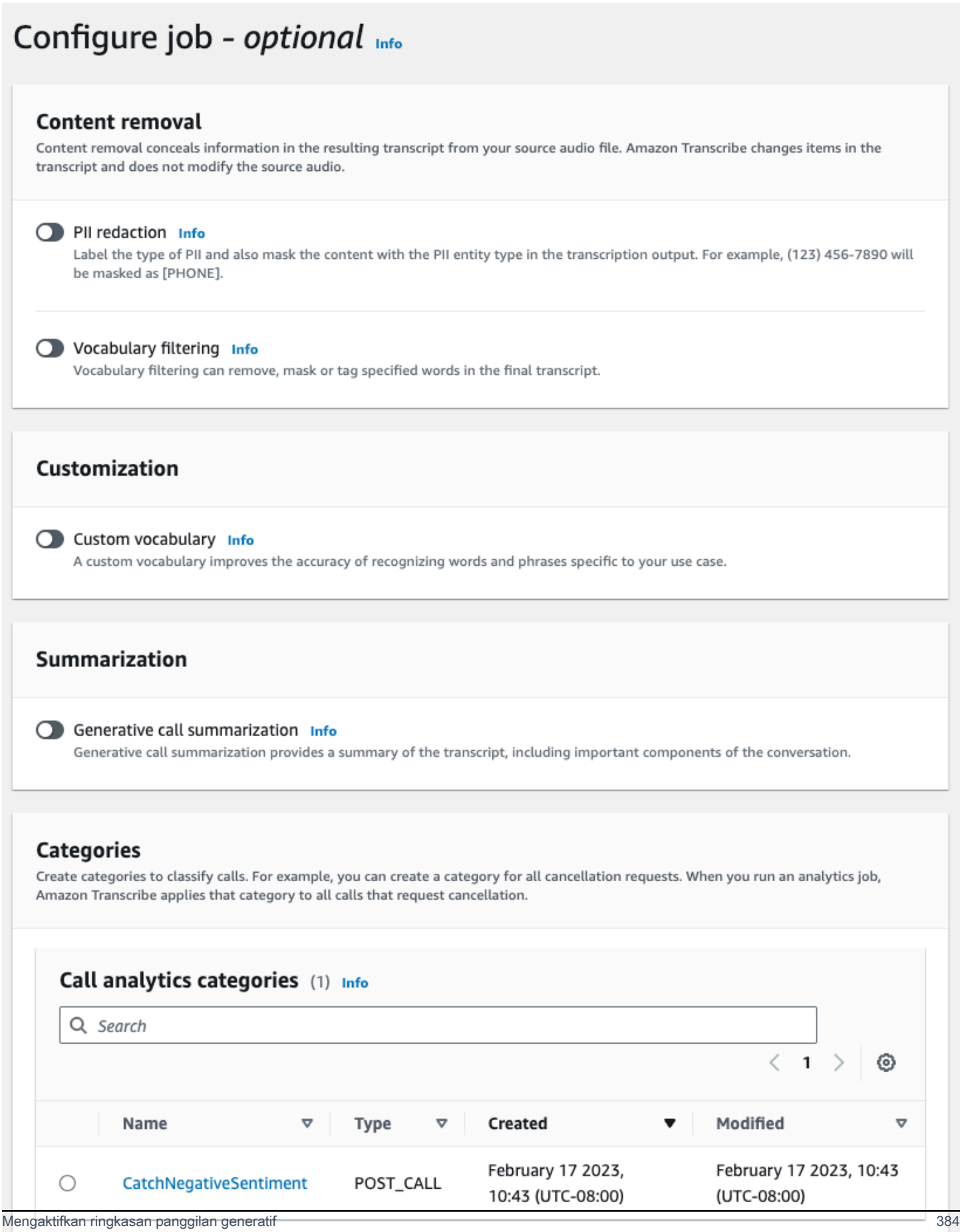
Mengaktifkan ringkasan panggilan generatif

> **Note**
>
> Didukung oleh Amazon Bedrock: AWS mengimplementasikan deteksi penyalahgunaan otomatis. Karena ringkasan pasca-kontak yang didukung oleh AI generatif dibangun di Amazon Bedrock, pengguna dapat memanfaatkan sepenuhnya kontrol yang diterapkan di Amazon Bedrock untuk menegakkan keselamatan, keamanan, dan penggunaan kecerdasan buatan (AI) yang bertanggung jawab.

Untuk menggunakan ringkasan panggilan generatif dengan pekerjaan analitik panggilan pos, lihat contoh berikut ini:

AWS Management Console

Di panel Ringkasan, aktifkan ringkasan panggilan Generatif untuk menerima ringkasan dalam output.


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
Search
1
Name
Type
Created
Modified
CatchNegativeSentiment
POST_CALL
February 17 2023, 10:43 (UTC-08:00)
February 17 2023, 10:43 (UTC-08:00)
If the above categories aren't relevant to your use case, you can create a new category. Create a new category.

AWS CLI

Contoh ini menggunakan start-call-analytics-jobperintah dan `Settings` parameter dengan `Summarization` sub-parameter. Untuk informasi selengkapnya, lihat `StartCallAnalyticsJob`.

```
aws transcribe start-call-analytics-job \
--region us-west-2 \
--call-analytics-job-name my-first-call-analytics-job \
--media MediaFileUri=s3://amzn-s3-demo-bucket/my-input-files/my-media-file.flac \
--output-location s3://amzn-s3-demo-bucket/my-output-files/ \
--data-access-role-arn arn:aws:iam::111122223333:role/ExampleRole \
--channel-definitions ChannelId=0,ParticipantRole=AGENT
 ChannelId=1,ParticipantRole=CUSTOMER
--settings '{"Summarization":{"GenerateAbstractiveSummary":true}}'
```

Berikut contoh lain menggunakan start-call-analytics-jobperintah, dan badan permintaan yang memungkinkan ringkasan untuk pekerjaan itu.

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

File my-call-analytics-job.json berisi badan permintaan berikut.

```
{
  "CallAnalyticsJobName": "my-first-call-analytics-job",
  "DataAccessRoleArn": "arn:aws:iam::111122223333:role/ExampleRole",
  "Media": {
    "MediaFileUri": "s3://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
  },
  "OutputLocation": "s3://amzn-s3-demo-bucket/my-output-files/",
  "ChannelDefinitions": [
    {
      "ChannelId": 0,
      "ParticipantRole": "AGENT"
    },
    {
```

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

AWS SDK untuk Python (Boto3)

Contoh ini menggunakan AWS SDK untuk Python (Boto3) untuk memulai Call Analytics dengan ringkasan diaktifkan menggunakan metode start_call_analytics_job. Untuk informasi selengkapnya, lihat `StartCallAnalyticsJob`.

Untuk contoh tambahan menggunakan AWS SDKs, termasuk contoh khusus fitur, skenario, dan lintas layanan, lihat bagian ini. Contoh kode untuk Amazon Transcribe menggunakan AWS SDKs

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

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

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

Analisis Panggilan Waktu Nyata

Real-time Call Analytics memberikan wawasan real-time yang dapat digunakan untuk mengatasi masalah dan mengurangi eskalasi saat terjadi.

Wawasan berikut tersedia dengan Analytics Panggilan real-time:

- Kategori acara yang menggunakan aturan untuk menandai kata kunci dan frasa tertentu; acara kategori dapat digunakan untuk membuat peringatan waktu nyata
- Deteksi masalah mengidentifikasi masalah yang diucapkan dalam setiap segmen audio
- Identifikasi PII (data sensitif) dalam transkrip teks Anda
- Redaksi PII (data sensitif) dari transkrip teks Anda
- Analisis sentimen untuk setiap segmen pidato

Selain real-time Call Analytics, juga Amazon Transcribe dapat melakukan analisis pasca-panggilan di aliran media Anda. Anda dapat menyertakan analitik pasca-panggilan dalam permintaan Analytics Panggilan real-time Anda menggunakan `PostCallAnalyticsSettings`parameter.

Wawasan waktu nyata

Bagian ini merinci wawasan yang tersedia untuk transkripsi Analisis Panggilan waktu nyata.

Kategori acara

Dengan menggunakan acara kategori, Anda dapat mencocokkan transkripsi berdasarkan kata kunci atau frasa yang tepat. Misalnya, jika Anda menyetel filter untuk frasa “Saya ingin berbicara dengan manajer”, Amazon Transcribe filter untuk frasa yang tepat itu.

Berikut adalah contoh output.

Untuk informasi selengkapnya tentang membuat kategori Analisis Panggilan real-time, lihatMembuat kategori untuk transkripsi waktu nyata.

> **Tip**
>
> Acara kategori memungkinkan Anda mengatur peringatan waktu nyata; lihat Membuat peringatan waktu nyata untuk kecocokan kategori untuk informasi selengkapnya.

Deteksi masalah

Deteksi masalah memberikan ringkasan singkat dari masalah yang terdeteksi dalam setiap segmen audio. Dengan menggunakan fitur deteksi masalah, Anda dapat:

- Mengurangi kebutuhan akan pencatatan manual selama dan setelah panggilan
- Meningkatkan efisiensi agen, memungkinkan mereka untuk merespon lebih cepat kepada pelanggan

> **Note**
>
> Deteksi masalah didukung dengan dialek bahasa Inggris ini: Australian (`en-AU`), British (`en-GB`), dan US (`en-US`).

Fitur deteksi masalah bekerja di semua industri dan sektor bisnis, dan berbasis konteks. Ini berfungsi out-of-the-box dan dengan demikian tidak mendukung penyesuaian, seperti pelatihan model atau kategori khusus.

Deteksi masalah dengan Analytics Panggilan real-time dilakukan pada setiap segmen audio lengkap.

Berikut adalah contoh output.

Identifikasi PII (data sensitif)

Label identifikasi data sensitif informasi identitas pribadi (PII) dalam transkrip teks. Parameter ini berguna untuk melindungi informasi pelanggan.

> **Note**
>
> Identifikasi PII real-time didukung dengan dialek bahasa Inggris ini: Australia (`en-AU`), Inggris (`en-GB`), AS (`en-US`) dan dengan dialek bahasa Spanyol (). `es-US`

Identifikasi PII dengan Analytics Panggilan real-time dilakukan pada setiap segmen audio lengkap.

Untuk melihat daftar PII yang diidentifikasi menggunakan fitur ini, atau untuk mempelajari lebih lanjut tentang identifikasi PII dengan Amazon Transcribe, lihat. Menyunting atau mengidentifikasi informasi yang dapat diidentifikasi secara pribadi

Berikut adalah contoh keluaran.

Redaksi PII (data sensitif)

Redaksi data sensitif menggantikan informasi identitas pribadi (PII) dalam transkrip teks Anda dengan jenis PII (misalnya,). `[NAME]` Parameter ini berguna untuk melindungi informasi pelanggan.

> **Note**
>
> Redaksi PII real-time didukung dengan dialek bahasa Inggris ini: Australia (`en-AU`), Inggris (), AS (`en-GBen-US`) dan dengan dialek bahasa Spanyol (). `es-US`

Redaksi PII dengan Analytics Panggilan real-time dilakukan pada setiap segmen audio lengkap.

Untuk melihat daftar PII yang disunting menggunakan fitur ini, atau untuk mempelajari lebih lanjut tentang redaksi dengan Amazon Transcribe, lihat. Menyunting atau mengidentifikasi informasi yang dapat diidentifikasi secara pribadi

Berikut adalah contoh keluaran.

Analisis sentimen

Analisis sentimen memperkirakan bagaimana perasaan pelanggan dan agen selama panggilan berlangsung. Metrik ini disediakan untuk setiap segmen pidato dan direpresentasikan sebagai nilai kualitatif (`positive`,, `neutral`mixed, ataunegative).

Dengan menggunakan parameter ini, Anda dapat mengevaluasi sentimen keseluruhan secara kualitatif untuk setiap peserta panggilan dan sentimen untuk setiap peserta selama setiap segmen pidato. Metrik ini dapat membantu mengidentifikasi apakah agen Anda dapat menyenangkan pelanggan yang kesal pada saat panggilan berakhir.

Analisis sentimen dengan Analytics Panggilan real-time dilakukan pada setiap segmen audio lengkap.

Analisis sentimen berfungsi out-of-the-box dan karenanya tidak mendukung penyesuaian, seperti pelatihan model atau kategori khusus.

Berikut adalah contoh output.

Membuat kategori untuk transkripsi waktu nyata

Real-time Call Analytics mendukung pembuatan kategori khusus, yang dapat Anda gunakan untuk menyesuaikan analisis transkrip Anda agar paling sesuai dengan kebutuhan bisnis spesifik Anda.

Anda dapat membuat kategori sebanyak yang Anda suka untuk mencakup berbagai skenario yang berbeda. Untuk setiap kategori yang Anda buat, Anda harus membuat antara 1 dan 20 aturan. Transkripsi Analytics Panggilan Real-time hanya mendukung aturan yang menggunakan `TranscriptFilter`(kecocokan kata kunci). Untuk detail lebih lanjut tentang penggunaan aturan dengan `CreateCallAnalyticsCategory`operasi, lihat Kriteria aturan untuk kategori Analisis Panggilan real-time bagian.

Jika konten di media Anda cocok dengan semua aturan yang Anda tentukan dalam kategori tertentu, beri Amazon Transcribe label output Anda dengan kategori tersebut. Lihat keluaran acara kategori untuk contoh kecocokan kategori dalam format keluaran JSON.

Berikut adalah beberapa contoh dari apa yang dapat Anda lakukan dengan kategori khusus:

- Identifikasi masalah yang memerlukan perhatian segera dengan menandai dan melacak kumpulan kata kunci tertentu

- Pantau kepatuhan, seperti agen yang berbicara (atau menghilangkan) frasa tertentu
- Tandai kata dan frasa tertentu secara real time; Anda kemudian dapat mengatur kecocokan kategori Anda untuk mengatur peringatan langsung. Misalnya, jika Anda membuat kategori Analisis Panggilan real-time untuk pelanggan yang mengatakan “berbicara dengan manajer”, Anda dapat mengatur peringatan peristiwa untuk kecocokan kategori waktu nyata ini yang memberi tahu manajer yang bertugas.

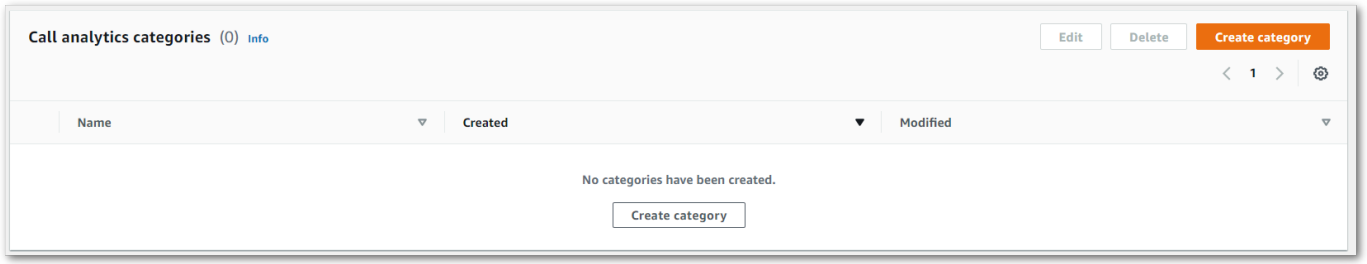
Kategori pasca-panggilan versus waktu nyata

Saat membuat kategori baru, Anda dapat menentukan apakah Anda ingin kategori tersebut dibuat sebagai kategori pasca-panggilan (`POST_CALL`) atau sebagai kategori real-time (`REAL_TIME`). Jika Anda tidak menentukan opsi, kategori Anda dibuat sebagai kategori pasca-panggilan secara default. Pertandingan kategori waktu nyata dapat digunakan untuk membuat peringatan waktu nyata. Untuk informasi selengkapnya, lihat Membuat peringatan waktu nyata untuk kecocokan kategori.

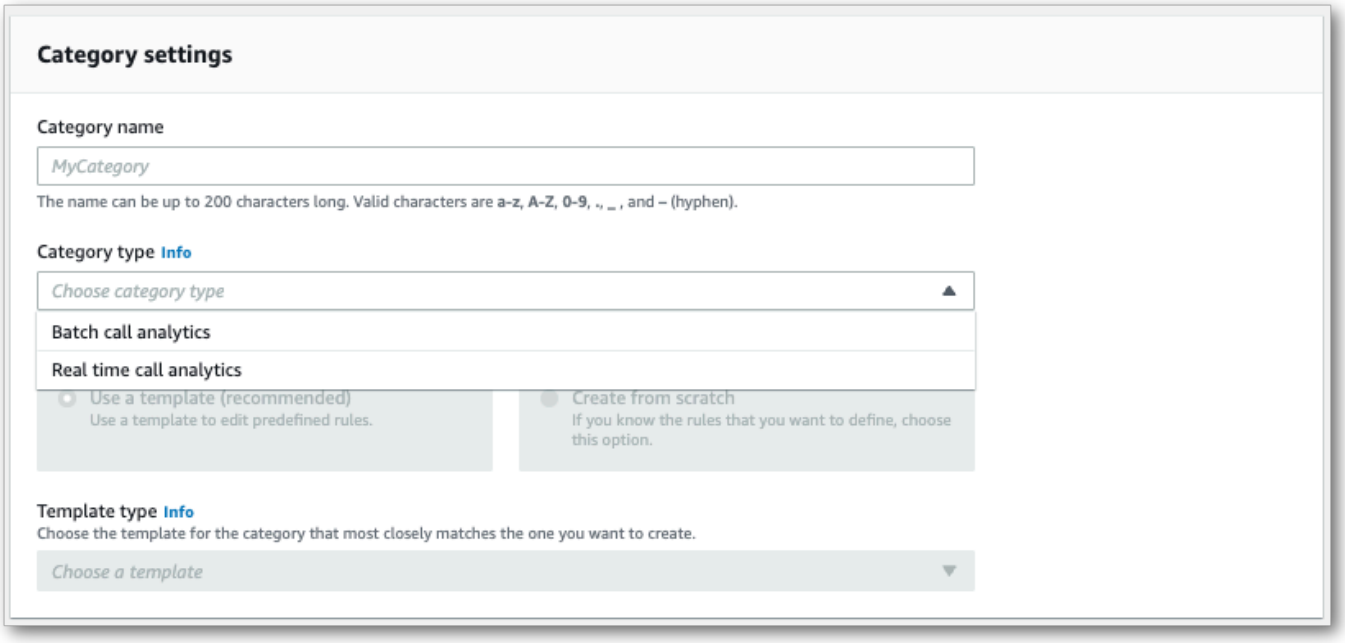
Untuk membuat kategori baru untuk Analytics Panggilan real-time, Anda dapat menggunakan AWS Management ConsoleAWS CLI,, atau AWS SDKs; lihat contoh berikut:

AWS Management Console

1. Di panel navigasi, di bawah Amazon Transcribe, pilih Amazon Transcribe Call Analytics.
2. Pilih kategori analisis Panggilan, yang akan membawa Anda ke halaman kategori analisis Panggilan. Pilih tombol Buat kategori.

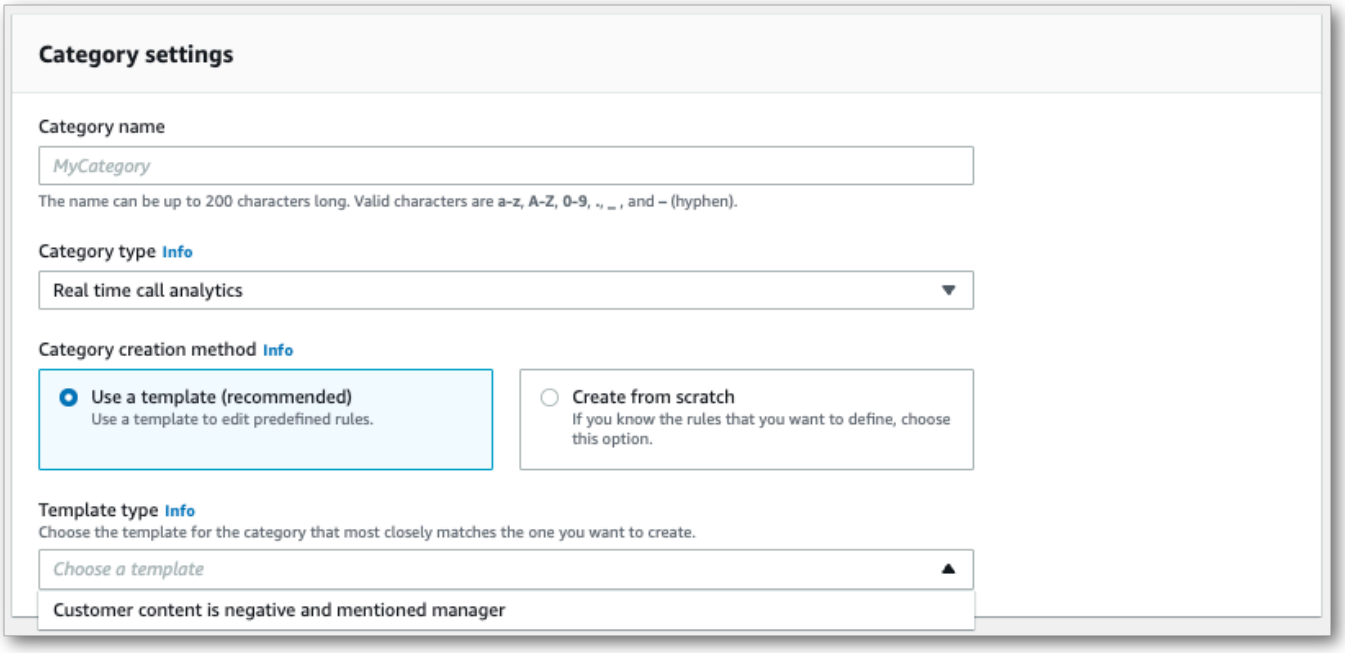


3. Anda sekarang berada di halaman kategori Buat. Masukkan nama untuk kategori Anda, lalu pilih 'Analisis panggilan waktu nyata' di menu tarik-turun tipe Kategori.

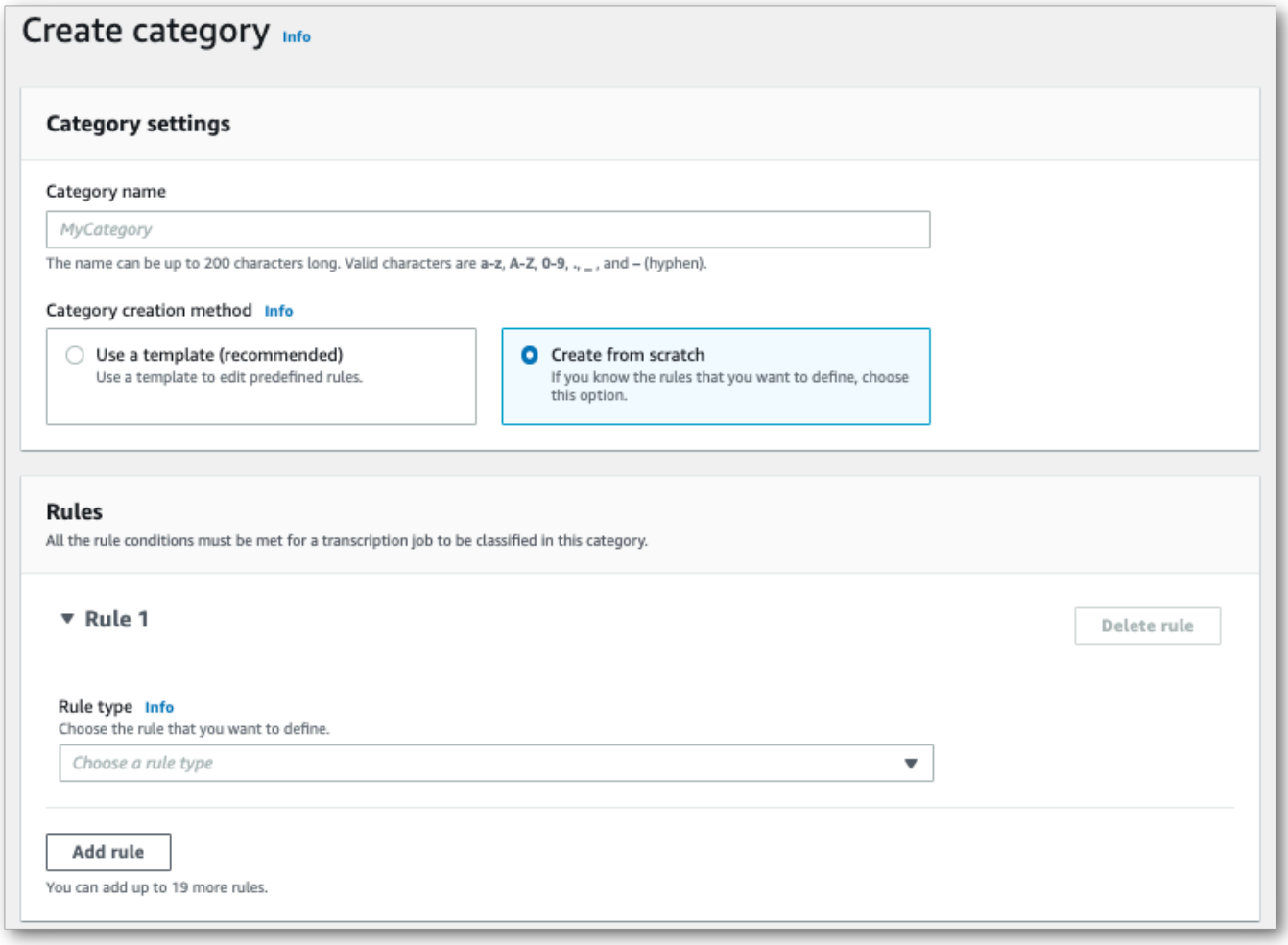


4. Anda dapat memilih template untuk membuat kategori Anda atau Anda dapat membuatnya dari awal.

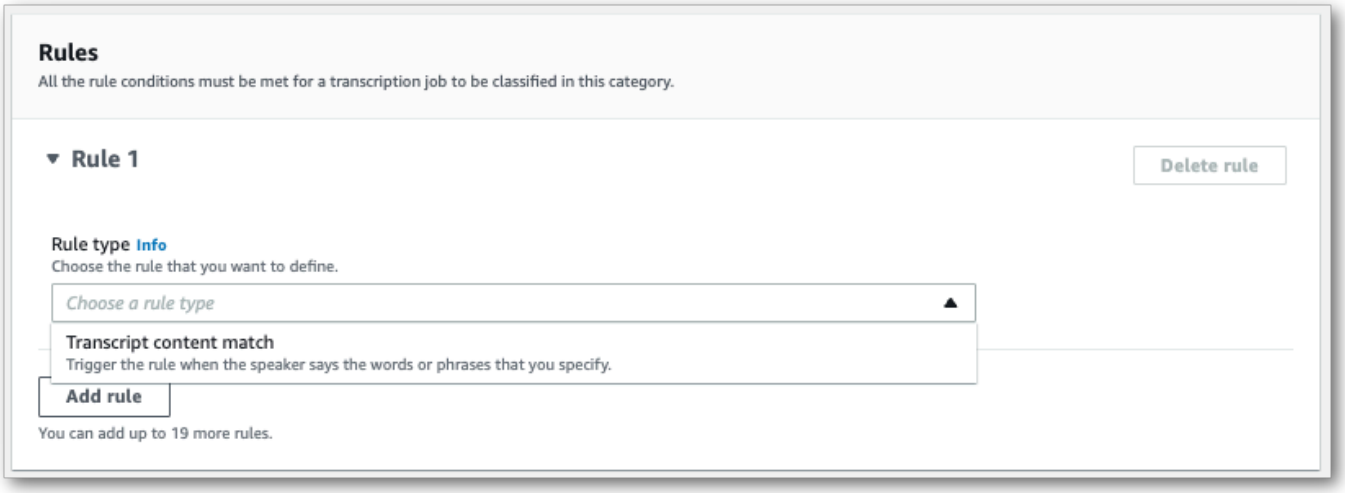
   Jika menggunakan templat: pilih Gunakan templat (disarankan), pilih templat yang Anda inginkan, lalu pilih Buat kategori.



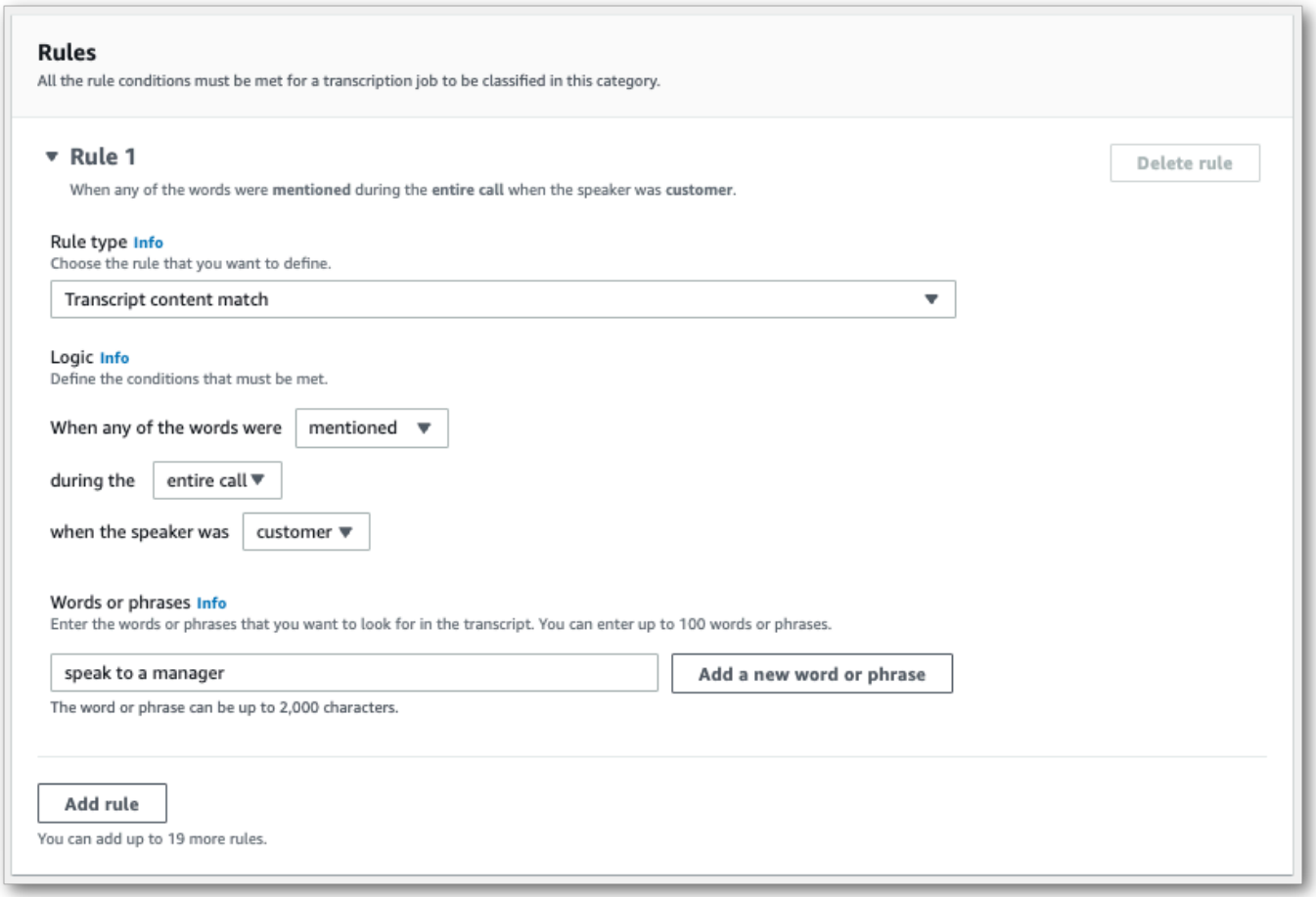
5. Jika membuat kategori kustom: pilih Buat dari awal.



6. Tambahkan aturan ke kategori Anda menggunakan menu tarik-turun. Anda dapat menambahkan hingga 20 aturan per kategori. Dengan transkripsi Analisis Panggilan real-time, Anda hanya dapat menyertakan aturan yang melibatkan pencocokan konten transkrip. Setiap pertandingan ditandai secara real time.



7. Berikut adalah contoh kategori dengan satu aturan: pelanggan yang mengatakan “berbicara dengan manajer” kapan saja dalam panggilan.



8. Setelah selesai menambahkan aturan ke kategori, pilih Buat kategori.

AWS CLI

Contoh ini menggunakan create-call-analytics-categoryperintah. Lihat informasi selengkapnya di `CreateCallAnalyticsCategory`, `CategoryProperties`, dan `Rule`.

Contoh berikut membuat kategori dengan aturan:

- Pelanggan mengucapkan frasa “berbicara dengan manajer” kapan saja dalam panggilan.

Contoh ini menggunakan create-call-analytics-categoryperintah, dan badan permintaan yang menambahkan aturan ke kategori Anda.

```
aws transcribe create-call-analytics-category \
--cli-input-json file://filepath/my-first-analytics-category.json
```

File my-first-analytics-category.json berisi badan permintaan berikut.

```
{
  "CategoryName": "my-new-real-time-category",
  "InputType": "REAL_TIME",
  "Rules": [
        {
            "TranscriptFilter": {
                "Negate": false,
                "Targets": [
                    "speak to the manager"
                ],
                "TranscriptFilterType": "EXACT"
            }
        }
    ]
}
```

AWS SDK untuk Python (Boto3)

Contoh ini menggunakan AWS SDK untuk Python (Boto3) untuk membuat kategori menggunakan `CategoryName` dan `Rules` argumen untuk metode create_call_analytics_category. Lihat informasi selengkapnya di `CreateCallAnalyticsCategory`, `CategoryProperties`, dan `Rule`.

Untuk contoh tambahan menggunakan AWS SDKs, termasuk contoh khusus fitur, skenario, dan lintas layanan, lihat bagian ini. Contoh kode untuk Amazon Transcribe menggunakan AWS SDKs

Contoh berikut membuat kategori dengan aturan:

- Pelanggan mengucapkan frasa “berbicara dengan manajer” kapan saja dalam panggilan.

```
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)
```

Kriteria aturan untuk kategori Analisis Panggilan real-time

Bagian ini menguraikan jenis `REAL_TIME` aturan kustom yang dapat Anda buat menggunakan operasi `CreateCallAnalyticsCategory`API.

Deteksi masalah terjadi secara otomatis, jadi Anda tidak perlu membuat aturan atau kategori apa pun untuk menandai masalah.

Perhatikan bahwa hanya kecocokan kata kunci yang didukung untuk transkripsi Analisis Panggilan real-time. Jika Anda ingin membuat kategori yang mencakup interupsi, keheningan, atau sentimen, lihat. Kriteria aturan untuk kategori analitik pasca-panggilan

Pencocokan kata kunci

Aturan menggunakan kata kunci (tipe `TranscriptFilter`data) dirancang untuk mencocokkan:

- Kata atau frasa khusus yang diucapkan oleh agen, pelanggan, atau keduanya
- Kata atau frasa khusus yang tidak diucapkan oleh agen, pelanggan, atau keduanya
- Kata atau frasa khusus yang terjadi dalam kerangka waktu tertentu

Berikut adalah contoh parameter yang tersedia dengan `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
}
```

Lihat `CreateCallAnalyticsCategory`dan `TranscriptFilter`untuk informasi lebih lanjut tentang parameter ini dan nilai valid yang terkait dengan masing-masing parameter.

Analisis pasca-panggilan dengan transkripsi waktu nyata

Analisis pasca-panggilan adalah fitur opsional yang tersedia dengan transkripsi Call Analytics waktu nyata. Selain wawasan analitik real-time standar, analitik pasca-panggilan memberi Anda hal-hal berikut:

- Item tindakan: Daftar item tindakan apa pun yang diidentifikasi dalam panggilan
- Interupsi: Mengukur jika dan ketika satu peserta memotong kalimat tengah peserta lainnya
- Masalah: Menyediakan masalah yang diidentifikasi dalam panggilan
- Kenyaringan: Mengukur volume di mana setiap peserta berbicara

- Waktu non-bicara: Mengukur periode waktu yang tidak mengandung ucapan
- Hasil: Memberikan hasil, atau resolusi, yang diidentifikasi dalam panggilan
- Kecepatan bicara: Mengukur kecepatan di mana kedua peserta berbicara
- Waktu bicara: Mengukur jumlah waktu (dalam milidetik) setiap peserta berbicara selama panggilan

Saat diaktifkan, analitik pasca-panggilan dari aliran audio menghasilkan transkrip yang mirip dengan analitik pasca-panggilan dari file audio dan menyimpannya di Amazon S3 bucket yang ditentukan. `OutputLocation` Selain itu, analitik pasca-panggilan merekam aliran audio Anda dan menyimpannya sebagai file audio (`WAV`format) dalam Amazon S3 bucket yang sama. Jika Anda mengaktifkan redaksi, transkrip yang disunting dan file audio yang disunting juga disimpan di bucket yang ditentukan. Amazon S3 Mengaktifkan analitik pasca-panggilan dengan aliran audio Anda menghasilkan antara dua dan empat file, seperti yang dijelaskan di sini:

- Jika redaksi tidak diaktifkan, file output Anda adalah:
  1. Transkrip yang belum disunting
  2. File audio yang belum disunting
- Jika redaksi diaktifkan tanpa opsi (`redacted`) yang tidak disunting, file keluaran Anda adalah:
  1. Transkrip yang disunting
  2. File audio yang disunting
- Jika redaksi diaktifkan dengan opsi (`redacted_and_unredacted`) yang tidak disunting, file keluaran Anda adalah:
  1. Transkrip yang disunting
  2. File audio yang disunting
  3. Transkrip yang belum disunting
  4. File audio yang belum disunting

Perhatikan bahwa jika Anda mengaktifkan analitik pasca-panggilan (`PostCallAnalyticsSettings`) dengan permintaan Anda, dan Anda menggunakan `FLAC` atau `OPUS-OGG` media, Anda tidak mendapatkan `loudnessScore` transkrip Anda dan tidak ada rekaman audio streaming Anda yang dibuat. Transcribe mungkin juga tidak dapat menyediakan analitik pasca-panggilan untuk streaming audio yang berjalan lama yang berlangsung lebih dari 90 menit.

Untuk informasi selengkapnya tentang wawasan yang tersedia dengan analitik pasca-panggilan untuk aliran audio, lihat bagian wawasan analitik pasca-panggilan.

> **Tip**
>
> Jika Anda mengaktifkan analitik pasca-panggilan dengan permintaan Analytics Panggilan real-time, semua `REAL-TIME` kategori `POST_CALL` dan kategori Anda diterapkan ke transkripsi analitik pasca-panggilan Anda.

Mengaktifkan analitik pasca-panggilan

Untuk mengaktifkan analitik pasca-panggilan, Anda harus menyertakan `PostCallAnalyticsSettings`parameter dalam permintaan Analytics Panggilan real-time Anda. Parameter berikut harus disertakan saat `PostCallAnalyticsSettings` diaktifkan:

- `OutputLocation`: Amazon S3 Ember tempat Anda ingin transkrip pasca-panggilan Anda disimpan.
- `DataAccessRoleArn`: Nama Sumber Daya Amazon (ARN) dari Amazon S3 peran yang memiliki izin untuk mengakses bucket yang ditentukan. Amazon S3 Perhatikan bahwa Anda juga harus menggunakan kebijakan Trust untuk analitik real-time.

Jika Anda menginginkan versi transkrip yang disunting, Anda dapat menyertakan `ContentRedactionOutput` atau `ContentRedactionType` dalam permintaan Anda. Untuk informasi selengkapnya tentang parameter ini, lihat `StartCallAnalyticsStreamTranscription`di Referensi API.

Untuk memulai transkripsi Analytics Panggilan real-time dengan analitik pasca-panggilan diaktifkan, Anda dapat menggunakan AWS Management Console(hanya demo), HTTP/2, atau. WebSockets Sebagai contoh, lihat Memulai transkripsi Analisis Panggilan waktu-nyata.

> **Important**
>
> Saat ini, AWS Management Console satu-satunya menawarkan demo untuk Analytics Panggilan real-time dengan contoh audio yang dimuat sebelumnya. Jika Anda ingin menggunakan audio Anda sendiri, Anda harus menggunakan API (HTTP/2, WebSockets, atau SDK).

Contoh keluaran analitik pasca-panggilan

Transkrip pasca-panggilan ditampilkan dalam turn-by-turn format berdasarkan segmen. Mereka termasuk karakteristik panggilan, sentimen, ringkasan panggilan, deteksi masalah, dan (opsional) redaksi PII. Jika salah satu kategori pasca-panggilan Anda cocok dengan konten audio, ini juga ada dalam output Anda.

Untuk meningkatkan akurasi dan lebih lanjut menyesuaikan transkrip Anda dengan kasus penggunaan Anda, seperti menyertakan istilah khusus industri, tambahkan kosakata khusus atau model bahasa khusus ke permintaan Analisis Panggilan Anda. Untuk menutupi, menghapus, atau menandai kata-kata yang tidak Anda inginkan dalam hasil transkripsi Anda, seperti kata-kata kotor, tambahkan pemfilteran kosakata.

Berikut adalah contoh keluaran analitik pasca-panggilan yang dikompilasi:

```
{
    "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
        }
    },
    ...
}
```

Memulai transkripsi Analisis Panggilan waktu-nyata

Sebelum memulai transkripsi Analisis Panggilan real-time, Anda harus membuat semua kategori yang Amazon Transcribe ingin Anda cocokkan dalam panggilan Anda.

> **Note**
>
> Transkrip Call Analytics tidak dapat dicocokkan secara surut dengan kategori baru. Hanya kategori yang Anda buat sebelum memulai transkripsi Call Analytics yang dapat diterapkan ke output transkripsi tersebut.

Jika Anda telah membuat satu atau beberapa kategori, dan audio Anda cocok dengan semua aturan dalam setidaknya satu kategori Anda, Amazon Transcribe tandai output Anda dengan kategori yang cocok. Jika Anda memilih untuk tidak menggunakan kategori, atau jika audio Anda tidak cocok dengan aturan yang ditentukan dalam kategori Anda, transkrip Anda tidak ditandai.

Untuk menyertakan analitik pasca-panggilan dengan transkripsi Analytics Panggilan real-time Anda, Anda harus menyediakan Amazon S3 bucket dalam permintaan menggunakan parameter. `OutputLocation` Anda juga harus menyertakan a `DataAccessRoleArn` yang memiliki izin menulis ke bucket yang ditentukan. Transkrip terpisah diproduksi dan disimpan dalam bucket yang ditentukan setelah menyelesaikan sesi streaming Call Analytics waktu nyata Anda.

Dengan Analytics Panggilan real-time, Anda juga memiliki opsi untuk membuat peringatan kategori waktu nyata; lihat Membuat peringatan waktu nyata untuk kecocokan kategori untuk instruksi.

Untuk memulai transkripsi Analytics Panggilan real-time, Anda dapat menggunakan AWS Management Console, HTTP/2, atau WebSockets; lihat contoh berikut:

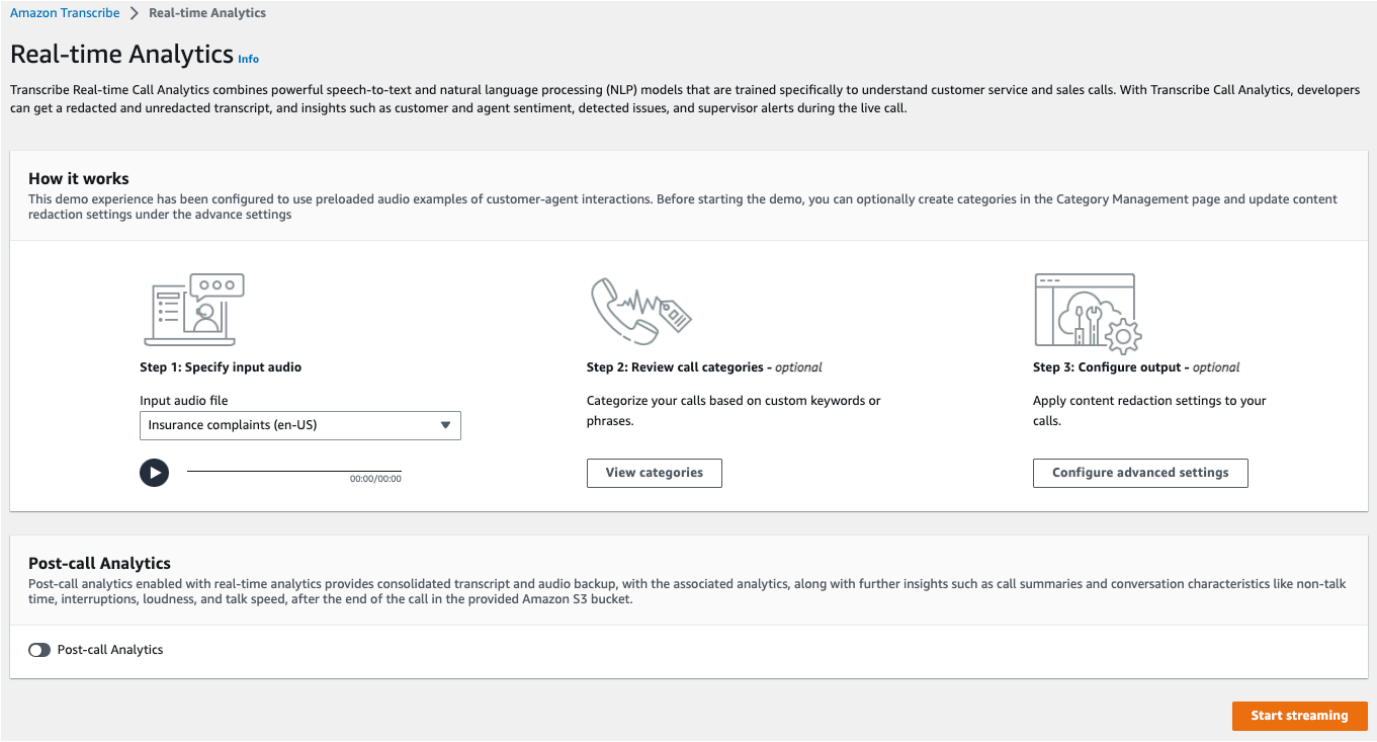
> **Important**
>
> Saat ini, AWS Management Console satu-satunya menawarkan demo untuk Analytics Panggilan real-time dengan contoh audio yang dimuat sebelumnya. Jika Anda ingin menggunakan audio Anda sendiri, Anda harus menggunakan API (HTTP/2, WebSockets, atau SDK).

AWS Management Console

Gunakan prosedur berikut untuk memulai permintaan Call Analytics. Panggilan yang cocok dengan semua karakteristik yang ditentukan oleh kategori diberi label dengan kategori tersebut.

> **Note**
>
> Hanya demo yang tersedia di AWS Management Console. Untuk memulai transkripsi analitik real-time khusus, Anda harus menggunakan API.

1. Di panel navigasi, di bawah Amazon Transcribe Call Analytics, pilih Analisis panggilan real-time.



2. Untuk Langkah 1: Tentukan audio input, pilih file uji demo dari menu tarik-turun.

Step 1: Specify input audio

Input audio file

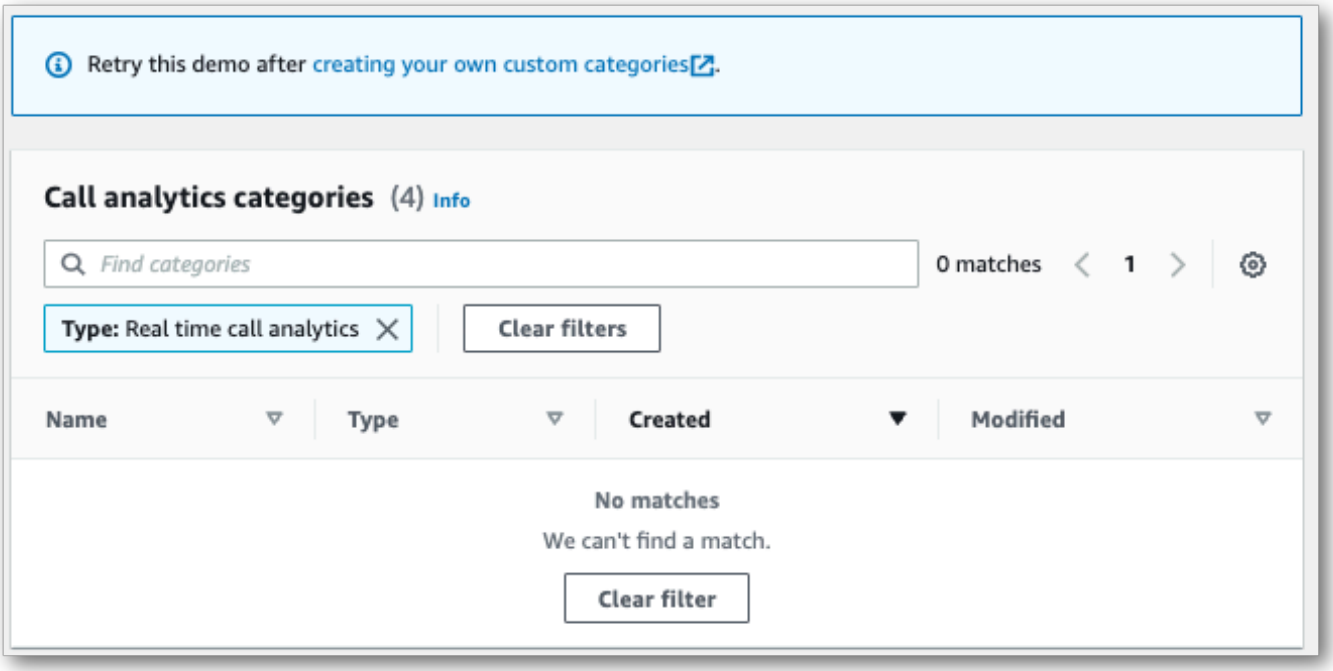
Insurance complaints (en-US)

Insurance complaints (en-US)

Hospitality complaints (en-US)

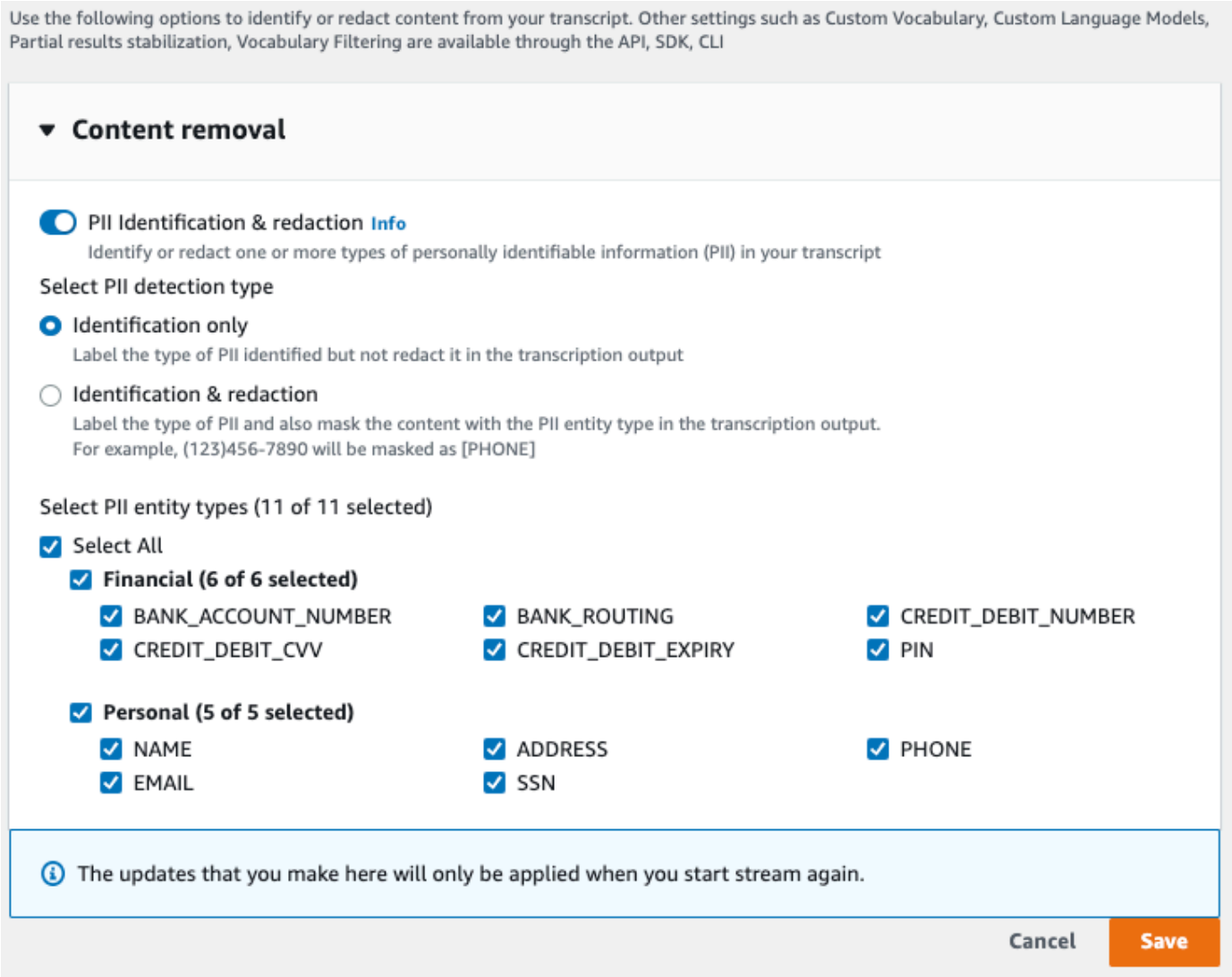
3. Untuk Langkah 2: Tinjau kategori panggilan, Anda memiliki opsi untuk meninjau kategori Analisis Panggilan real-time yang sebelumnya Anda buat. Semua kategori Analisis Panggilan real-time diterapkan ke transkripsi Anda.

   Memilih kategori Tampilan membuka panel baru yang menampilkan kategori Analytics Panggilan real-time Anda yang ada dan menyediakan tautan untuk membuat yang baru.



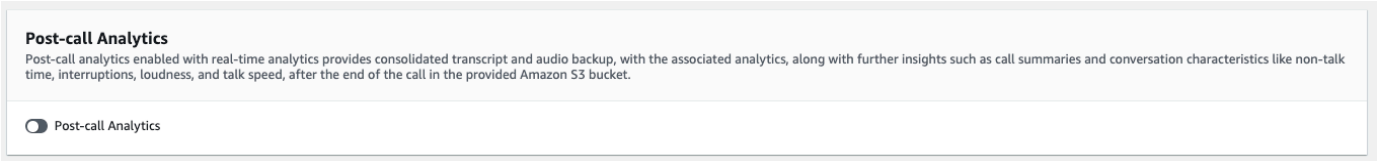
4. Untuk Langkah 3: Konfigurasikan input dan output, Anda memiliki opsi untuk menerapkan pengaturan tambahan.

   Memilih Konfigurasi pengaturan lanjutan membuka panel baru tempat Anda dapat menentukan pengaturan redaksi konten.

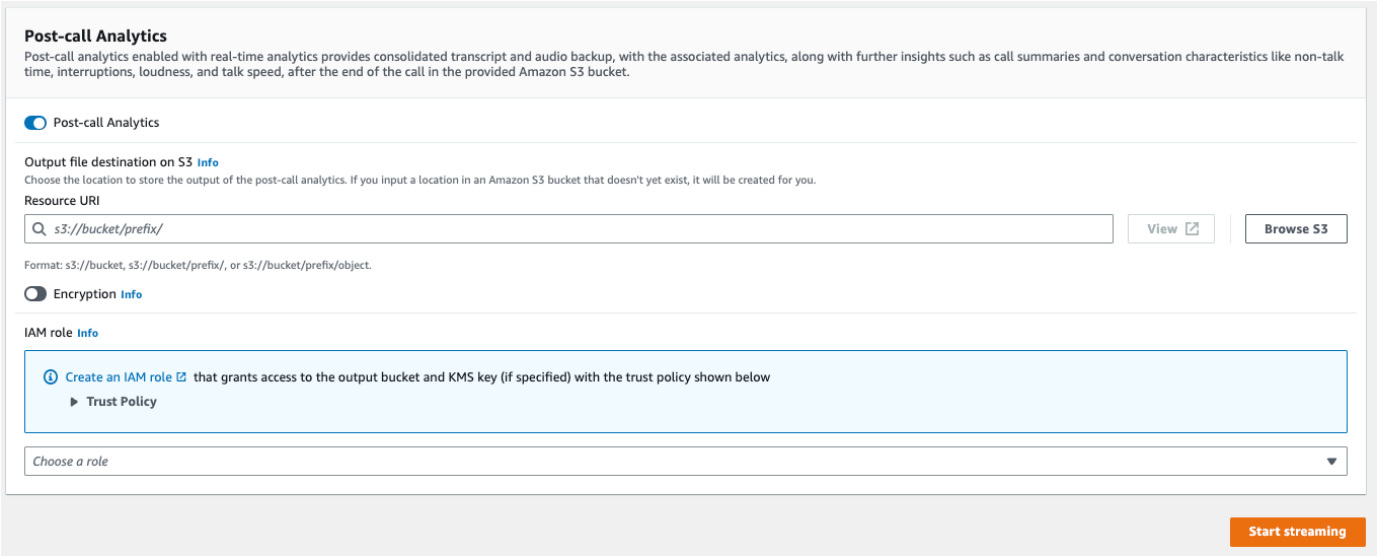


Setelah Anda membuat semua pilihan Anda, pilih Simpan untuk kembali ke halaman utama.

5. Untuk menerapkan analitik tambahan, Anda dapat beralih di Analytics Pasca-Panggilan. Ini memberi Anda analitik yang sama dengan transkripsi analitik pasca-panggilan, termasuk interupsi, kenyaringan, waktu non-bicara, kecepatan bicara, waktu bicara, masalah, item tindakan, dan hasil. Output analitik pasca-panggilan disimpan dalam file terpisah dari transkrip Analytics Panggilan real-time Anda.



Jika Anda menerapkan analitik pasca-panggilan, Anda harus menentukan tujuan file Amazon S3 keluaran dan IAM peran. Anda dapat memilih untuk mengenkripsi output Anda secara opsional.



6. Pilih Mulai streaming.

Aliran HTTP/2

Contoh ini membuat permintaan HTTP/2 dengan Call Analytics diaktifkan. Untuk informasi lebih lanjut tentang menggunakan streaming HTTP/2 dengan Amazon Transcribe, lihat. Menyiapkan aliran HTTP/2 Untuk detail selengkapnya tentang parameter dan header khusus untuk Amazon Transcribe, lihat `StartCallAnalyticsStreamTranscription`.

Contoh ini mencakup analitik pasca-panggilan. Jika Anda tidak ingin analitik pasca-panggilan, hapus `PostCallAnalyticsSettings` bagian dari permintaan.

Perhatikan bahwa peristiwa konfigurasi yang ditampilkan dalam contoh berikut harus diteruskan sebagai peristiwa pertama dalam aliran.

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

```
{
    "AudioStream": {
        "AudioEvent": {
            "AudioChunk": blob
        },
        "ConfigurationEvent": {
            "ChannelDefinitions": [
                {
                    "ChannelId": 0,
                    "ParticipantRole": "AGENT"
                },
                {
                     "ChannelId": 1,
                    "ParticipantRole": "CUSTOMER"
                }
            ],
            "PostCallAnalyticsSettings": {
                "OutputLocation": "s3://amzn-s3-demo-bucket/my-output-files/",
                "DataAccessRoleArn": "arn:aws:iam::111122223333:role/ExampleRole"
            }
        }
    }
}
```

Definisi parameter dapat ditemukan di Referensi API; parameter yang umum untuk semua operasi AWS API tercantum di bagian Parameter Umum.

WebSocket aliran

Contoh ini membuat URL presigned yang menggunakan Call Analytics dalam WebSocket aliran. Jeda baris telah ditambahkan untuk keterbacaan. Untuk informasi selengkapnya tentang penggunaan WebSocket stream dengan Amazon Transcribe, lihatMenyiapkan WebSocket aliran. Untuk detail lebih lanjut tentang parameter, lihat `StartCallAnalyticsStreamTranscription`.

Contoh ini mencakup analitik pasca-panggilan. Jika Anda tidak ingin analitik pasca-panggilan, hapus `PostCallAnalyticsSettings` bagian dari permintaan.

Perhatikan bahwa peristiwa konfigurasi yang ditampilkan dalam contoh berikut harus diteruskan sebagai peristiwa pertama dalam aliran.

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

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

{
    "AudioStream": {
        "AudioEvent": {
            "AudioChunk": blob
        },
        "ConfigurationEvent": {
            "ChannelDefinitions": [
                {
                    "ChannelId": 0,
                    "ParticipantRole": "AGENT"
                },
                {
                     "ChannelId": 1,
                    "ParticipantRole": "CUSTOMER"
                }
            ],
            "PostCallAnalyticsSettings": {
                "OutputLocation": "s3://amzn-s3-demo-bucket/my-output-files/",
                "DataAccessRoleArn": "arn:aws:iam::111122223333:role/ExampleRole"
            }
        }
    }
}
```

Definisi parameter dapat ditemukan di Referensi API; parameter yang umum untuk semua operasi AWS API tercantum di bagian Parameter Umum.

> **Tip**
>
> HTTP/2 di atas dan WebSocket contoh termasuk analitik pasca-panggilan. Jika Anda tidak ingin analitik pasca-panggilan, hapus `PostCallAnalyticsSettings` bagian dari permintaan.
> Jika Anda mengaktifkan`PostCallAnalyticsSettings`, Anda harus mengirim acara konfigurasi sebagai acara pertama. Acara konfigurasi Anda mencakup pengaturan untuk `ChannelDenifitions` dan`PostStreamAnalyticsSettings`, seperti yang ditunjukkan pada contoh sebelumnya.
> Data biner diteruskan sebagai pesan biner dengan `content-type application/octet-stream` dan peristiwa konfigurasi diteruskan sebagai pesan teks dengan`content-type application/json`.
> Untuk informasi selengkapnya, lihat Menyiapkan transkripsi streaming.

Membuat peringatan waktu nyata untuk kecocokan kategori

Untuk mengatur peringatan waktu nyata, Anda harus terlebih dahulu membuat `TranscriptFilterType`kategori dengan `REAL_TIME` bendera. Bendera ini memungkinkan kategori Anda diterapkan ke transkripsi Analisis Panggilan real-time.

Untuk petunjuk tentang membuat kategori baru, lihatMembuat kategori untuk transkripsi waktu nyata.

Saat Anda memulai transkripsi Analisis Panggilan real-time, semua kategori yang memiliki `REAL_TIME` tanda secara otomatis diterapkan ke output transkripsi Anda di tingkat segmen. Jika `TranscriptFilterType` kecocokan terjadi, itu muncul di bawah `CategoryEvent` bagian transkrip Anda. Anda kemudian dapat menggunakan parameter ini dan sub-parameternya, `MatchedCategories` dan`MatchedDetails`, untuk mengatur peringatan real-time kustom.

Berikut adalah contoh output transkripsi Call Analytics real-time untuk `CategoryEvent` kecocokan:

```
"CategoryEvent": {
    "MatchedCategories": [ "shipping-complaint" ],
    "MatchedDetails": {
        "my package never arrived" : {
            "TimestampRanges": [
                {
                    "BeginOffsetMillis": 19010,
                    "EndOffsetMillis": 22690
```

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

Contoh sebelumnya mewakili kecocokan teks yang tepat dengan pidato "paket saya tidak pernah tiba," yang mewakili aturan dalam kategori 'pengiriman-keluhan'.

Anda dapat mengatur peringatan waktu nyata untuk menyertakan kombinasi parameter yang tercantum. Misalnya, Anda dapat menyetel peringatan agar hanya menyertakan frasa yang cocok (`MatchedDetails`) atau hanya nama kategori (`MatchedCategories`). Atau Anda dapat mengatur peringatan Anda untuk menyertakan semua parameter.

Cara Anda mengatur peringatan waktu nyata tergantung pada antarmuka organisasi dan jenis peringatan yang Anda inginkan. Misalnya, Anda dapat mengatur `CategoryEvent` kecocokan untuk mengirim pemberitahuan pop-up, email, teks, atau peringatan lain yang dapat diterima sistem Anda.

Output Analisis Panggilan Waktu Nyata

Transkrip Analisis Panggilan Real-time ditampilkan dalam turn-by-turn format berdasarkan segmen. Mereka termasuk peristiwa kategori, deteksi masalah, sentimen, dan identifikasi dan redaksi PII. Acara kategori memungkinkan Anda mengatur peringatan waktu nyata; lihat Membuat peringatan waktu nyata untuk kecocokan kategori untuk informasi selengkapnya.

Untuk meningkatkan akurasi dan lebih lanjut menyesuaikan transkrip Anda dengan kasus penggunaan Anda, seperti menyertakan istilah khusus industri, tambahkan kosakata khusus atau model bahasa khusus ke permintaan Analisis Panggilan Anda. Untuk menutupi, menghapus, atau menandai kata-kata yang tidak Anda inginkan dalam hasil transkripsi Anda, seperti kata-kata kotor, tambahkan pemfilteran kosakata.

Bagian berikut menunjukkan contoh keluaran JSON untuk transkripsi Call Analytics waktu nyata.

Kategori acara

Inilah yang terlihat seperti kecocokan kategori dalam keluaran transkripsi Anda. Contoh ini menunjukkan bahwa audio dari stempel waktu 19010 milidetik ke stempel waktu 22690 milidetik cocok dengan kategori 'keluhan jaringan'. Dalam hal ini, kategori 'keluhan jaringan' khusus mengharuskan pelanggan mengatakan "masalah jaringan" (pencocokan kata yang tepat).

```
"CategoryEvent": {
```

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

Deteksi masalah

Inilah yang terlihat seperti kecocokan deteksi masalah dalam keluaran transkripsi Anda. Contoh ini menunjukkan bahwa teks dari karakter 26 ke karakter 62 menjelaskan masalah.

```
"UtteranceEvent": {
    ...
    "Transcript": "Wang Xiulan I'm tired of the network issues my phone is having.",
    ...
    "IssuesDetected": [
        {
            "CharacterOffsets": {
                "BeginOffsetChar": 26,
                "EndOffsetChar": 62
            }
        }
    ]
},
```

Sentimen

Inilah yang terlihat seperti analisis sentimen dalam keluaran transkripsi Anda.

```
"UtteranceEvent": {
    ...
    "Sentiment": "NEGATIVE",
    "Items": [{
```

```
        ...
```

Identifikasi PII

Inilah yang terlihat seperti identifikasi PII dalam keluaran transkripsi Anda.

```
"Entities": [
    {
        "Content": "Wang Xiulan",
        "Category": "PII",
        "Type": "NAME",
        "BeginOffsetMillis": 7999375,
        "EndOffsetMillis": 199375,
        "Confidence": 0.9989
    }
],
```

Redaksi PII

Inilah yang tampak seperti redaksi PII dalam keluaran transkripsi Anda.

```
"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
        }
    ]
},
```

Output Analisis Panggilan real-time yang dikompilasi

Untuk singkatnya, beberapa konten diganti dengan elips dalam keluaran transkripsi berikut.

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

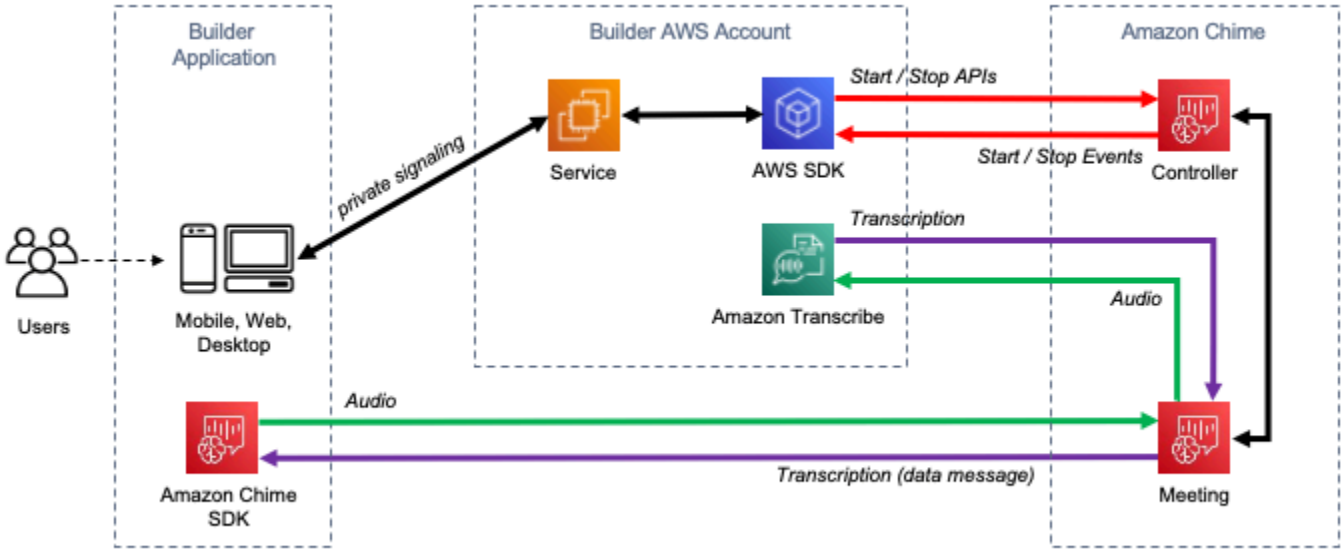
Mentranskripsikan Amazon Chime panggilan Anda secara real time

Amazon Transcribe terintegrasi dengan Amazon Chime SDK, memfasilitasi transkripsi panggilan Anda secara real-time. Amazon Chime

Saat Anda meminta transkripsi menggunakan Amazon Chime SDK API, Amazon Chime mulailah mengalirkan audio ke Amazon Transcribe dan terus melakukannya selama panggilan berlangsung.

Amazon Chime SDK menggunakan algoritma 'pembicara aktif' untuk memilih dua pembicara aktif teratas, dan kemudian mengirimkan audio mereka Amazon Transcribe sebagai dua saluran terpisah melalui satu aliran. Peserta rapat menerima transkripsi yang dikaitkan dengan pengguna melalui pesan data Amazon Chime SDK. Anda dapat melihat contoh pengiriman di Panduan Pengembang Amazon Chime SDK.

Aliran data Amazon Chime transkripsi digambarkan dalam diagram berikut:



Untuk informasi tambahan dan petunjuk terperinci tentang cara mengatur Amazon Chime transkripsi waktu nyata, lihat Menggunakan transkripsi langsung Amazon Chime SDK di Panduan Pengembang Amazon Chime SDK. Untuk operasi API, lihat Referensi API Amazon Chime SDK.

Menyelam lebih dalam dengan Blog AWS Machine Learning

Untuk mempelajari lebih lanjut tentang meningkatkan akurasi dengan transkripsi waktu nyata, lihat:

- Amazon Chime Rapat SDK sekarang mendukung transkripsi langsung dengan dan Amazon TranscribeAmazon Transcribe Medical
- Amazon Chime SDK untuk Solusi Telemedicine

Contoh kode untuk Amazon Transcribe menggunakan AWS SDKs

Contoh kode berikut menunjukkan cara menggunakan Amazon Transcribe dengan AWS perangkat pengembangan perangkat lunak (SDK).

Tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Skenario adalah contoh kode yang menunjukkan kepada Anda bagaimana menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan atau dikombinasikan dengan yang lain Layanan AWS.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Contoh kode

- Contoh dasar untuk Amazon Transcribe menggunakan AWS SDKs
  - Tindakan untuk Amazon Transcribe menggunakan AWS SDKs
    - Gunakan CreateVocabulary dengan AWS SDK atau CLI
    - Gunakan DeleteMedicalTranscriptionJob dengan AWS SDK atau CLI
    - Gunakan DeleteTranscriptionJob dengan AWS SDK atau CLI
    - Gunakan DeleteVocabulary dengan AWS SDK atau CLI
    - Gunakan GetTranscriptionJob dengan AWS SDK atau CLI
    - Gunakan GetVocabulary dengan AWS SDK atau CLI
    - Gunakan ListMedicalTranscriptionJobs dengan AWS SDK atau CLI
    - Gunakan ListTranscriptionJobs dengan AWS SDK atau CLI
    - Gunakan ListVocabularies dengan AWS SDK atau CLI
    - Gunakan StartMedicalTranscriptionJob dengan AWS SDK atau CLI
    - Gunakan StartTranscriptionJob dengan AWS SDK atau CLI
    - Gunakan UpdateVocabulary dengan AWS SDK atau CLI
- Skenario untuk Amazon Transcribe menggunakan AWS SDKs

- Membangun aplikasi streaming Amazon Transcribe
- Mengonversi teks menjadi ucapan dan kembali ke teks menggunakan AWS SDK
- Membuat dan menyempurnakan kosakata kustom Amazon Transcribe menggunakan SDK AWS
- Transcribe audio dan dapatkan data pekerjaan dengan Amazon Transcribe menggunakan SDK AWS

Contoh dasar untuk Amazon Transcribe menggunakan AWS SDKs

Contoh kode berikut menunjukkan cara menggunakan dasar-dasar Amazon Transcribe with. AWS SDKs

Contoh

- Tindakan untuk Amazon Transcribe menggunakan AWS SDKs
  - Gunakan CreateVocabulary dengan AWS SDK atau CLI
  - Gunakan DeleteMedicalTranscriptionJob dengan AWS SDK atau CLI
  - Gunakan DeleteTranscriptionJob dengan AWS SDK atau CLI
  - Gunakan DeleteVocabulary dengan AWS SDK atau CLI
  - Gunakan GetTranscriptionJob dengan AWS SDK atau CLI
  - Gunakan GetVocabulary dengan AWS SDK atau CLI
  - Gunakan ListMedicalTranscriptionJobs dengan AWS SDK atau CLI
  - Gunakan ListTranscriptionJobs dengan AWS SDK atau CLI
  - Gunakan ListVocabularies dengan AWS SDK atau CLI
  - Gunakan StartMedicalTranscriptionJob dengan AWS SDK atau CLI
  - Gunakan StartTranscriptionJob dengan AWS SDK atau CLI
  - Gunakan UpdateVocabulary dengan AWS SDK atau CLI

Tindakan untuk Amazon Transcribe menggunakan AWS SDKs

Contoh kode berikut menunjukkan cara melakukan tindakan Amazon Transcribe individual dengan. AWS SDKs Setiap contoh menyertakan tautan ke GitHub, di mana Anda dapat menemukan instruksi untuk mengatur dan menjalankan kode.

Kutipan ini memanggil Amazon Transcribe API dan merupakan kutipan kode dari program yang lebih besar yang harus dijalankan dalam konteks. Anda dapat melihat tindakan dalam konteks diSkenario untuk Amazon Transcribe menggunakan AWS SDKs .

Contoh berikut hanya mencakup tindakan yang paling umum digunakan. Untuk daftar lengkapnya, lihat Referensi Amazon Transcribe API.

Contoh

- Gunakan CreateVocabulary dengan AWS SDK atau CLI
- Gunakan DeleteMedicalTranscriptionJob dengan AWS SDK atau CLI
- Gunakan DeleteTranscriptionJob dengan AWS SDK atau CLI
- Gunakan DeleteVocabulary dengan AWS SDK atau CLI
- Gunakan GetTranscriptionJob dengan AWS SDK atau CLI
- Gunakan GetVocabulary dengan AWS SDK atau CLI
- Gunakan ListMedicalTranscriptionJobs dengan AWS SDK atau CLI
- Gunakan ListTranscriptionJobs dengan AWS SDK atau CLI
- Gunakan ListVocabularies dengan AWS SDK atau CLI
- Gunakan StartMedicalTranscriptionJob dengan AWS SDK atau CLI
- Gunakan StartTranscriptionJob dengan AWS SDK atau CLI
- Gunakan UpdateVocabulary dengan AWS SDK atau CLI

Gunakan **`CreateVocabulary`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`CreateVocabulary`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Membuat dan menyempurnakan kosakata khusus

.NET

SDK untuk .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
    /// <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;
    }
```

- Untuk detail API, lihat CreateVocabularydi Referensi AWS SDK untuk .NET API.

CLI

AWS CLI

Untuk membuat kosakata khusus

`create-vocabulary`Contoh berikut membuat kosakata kustom. Untuk membuat kosakata khusus, Anda harus membuat file teks dengan semua istilah yang ingin Anda transkripsikan dengan lebih akurat. Untuk vocabulary-file-uri, tentukan URI Amazon Simple Storage Service (Amazon S3) dari file teks tersebut. Untuk kode bahasa, tentukan kode bahasa yang sesuai dengan bahasa kosakata kustom Anda. Untuk nama kosakata, tentukan apa yang ingin Anda sebut kosakata khusus Anda.

```
aws transcribe create-vocabulary \
    --language-code language-code \
    --vocabulary-name cli-vocab-example \
    --vocabulary-file-uri s3://amzn-s3-demo-bucket/Amazon-S3-prefix/the-text-file-for-the-custom-vocabulary.txt
```

Output:

```
{
    "VocabularyName": "cli-vocab-example",
    "LanguageCode": "language-code",
    "VocabularyState": "PENDING"
}
```

Untuk informasi selengkapnya, lihat Kosakata Khusus di Panduan Pengembang Amazon Transcribe.

- Untuk detail API, lihat CreateVocabularydi Referensi AWS CLI Perintah.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
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
```

- Untuk detail API, lihat CreateVocabularydi AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`DeleteMedicalTranscriptionJob`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`DeleteMedicalTranscriptionJob`.

.NET

SDK untuk .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
    /// <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;
    }
```

- Untuk detail API, lihat DeleteMedicalTranscriptionJobdi Referensi AWS SDK untuk .NET API.

CLI

AWS CLI

Untuk menghapus pekerjaan transkripsi medis

`delete-medical-transcription-job`Contoh berikut menghapus pekerjaan transkripsi medis.

```
aws transcribe delete-medical-transcription-job \
    --medical-transcription-job-name medical-transcription-job-name
```

Perintah ini tidak menghasilkan output.

Untuk informasi selengkapnya, lihat DeleteMedicalTranscriptionJobdi Panduan Pengembang Amazon Transcribe.

- Untuk detail API, lihat DeleteMedicalTranscriptionJobdi Referensi AWS CLI Perintah.

JavaScript

SDK untuk JavaScript (v3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

Buat klien.

```
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 };
```

Hapus pekerjaan transkripsi medis.

```
// 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_transciption_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();
```

- Untuk informasi selengkapnya, lihat AWS SDK untuk JavaScript Panduan Developer.
- Untuk detail API, lihat DeleteMedicalTranscriptionJobdi Referensi AWS SDK untuk JavaScript API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`DeleteTranscriptionJob`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`DeleteTranscriptionJob`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Membuat dan menyempurnakan kosakata khusus

.NET

SDK untuk .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
    /// <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;
    }
```

- Untuk detail API, lihat DeleteTranscriptionJobdi Referensi AWS SDK untuk .NET API.

CLI

AWS CLI

Untuk menghapus salah satu pekerjaan transkripsi Anda

`delete-transcription-job`Contoh berikut menghapus salah satu pekerjaan transkripsi Anda.

```
aws transcribe delete-transcription-job \
    --transcription-job-name your-transcription-job
```

Perintah ini tidak menghasilkan output.

Untuk informasi selengkapnya, lihat DeleteTranscriptionJobdi Panduan Pengembang Amazon Transcribe.

- Untuk detail API, lihat DeleteTranscriptionJobdi Referensi AWS CLI Perintah.

JavaScript

SDK untuk JavaScript (v3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

Hapus pekerjaan transkripsi.

```
// 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, 'transciption_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();
```

Buat klien.

```
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 };
```

- Untuk informasi selengkapnya, lihat AWS SDK untuk JavaScript Panduan Developer.
- Untuk detail API, lihat DeleteTranscriptionJobdi Referensi AWS SDK untuk JavaScript API.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
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
```

- Untuk detail API, lihat DeleteTranscriptionJobdi AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`DeleteVocabulary`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`DeleteVocabulary`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Membuat dan menyempurnakan kosakata khusus

.NET

SDK untuk .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
    /// <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;
    }
```

- Untuk detail API, lihat DeleteVocabularydi Referensi AWS SDK untuk .NET API.

CLI

AWS CLI

Untuk menghapus kosakata khusus

`delete-vocabulary`Contoh berikut menghapus kosakata kustom.

```
aws transcribe delete-vocabulary \
    --vocabulary-name vocabulary-name
```

Perintah ini tidak menghasilkan output.

Untuk informasi selengkapnya, lihat Kosakata Khusus di Panduan Pengembang Amazon Transcribe.

- Untuk detail API, lihat DeleteVocabularydi Referensi AWS CLI Perintah.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
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
```

- Untuk detail API, lihat DeleteVocabularydi AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`GetTranscriptionJob`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`GetTranscriptionJob`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Membuat dan menyempurnakan kosakata khusus
- Transcribe audio dan dapatkan data pekerjaan

.NET

SDK untuk .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
    /// <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;
    }
```

- Untuk detail API, lihat GetTranscriptionJobdi Referensi AWS SDK untuk .NET API.

CLI

AWS CLI

Untuk mendapatkan informasi tentang pekerjaan transkripsi tertentu

`get-transcription-job`Contoh berikut mendapatkan informasi tentang pekerjaan transkripsi tertentu. Untuk mengakses hasil transkripsi, gunakan TranscriptFileUri parameter. Gunakan MediaFileUri parameter untuk melihat file audio mana yang Anda transkripsikan dengan pekerjaan ini. Anda dapat menggunakan objek Pengaturan untuk melihat fitur opsional yang telah Anda aktifkan dalam pekerjaan transkripsi.

```
aws transcribe get-transcription-job \
    --transcription-job-name your-transcription-job
```

Output:

```
{
    "TranscriptionJob": {
```

```
        "TranscriptionJobName": "your-transcription-job",
        "TranscriptionJobStatus": "COMPLETED",
        "LanguageCode": "language-code",
        "MediaSampleRateHertz": 48000,
        "MediaFormat": "mp4",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.file-
extension"
        },
        "Transcript": {
            "TranscriptFileUri": "https://Amazon-S3-file-location-of-
transcription-output"
        },
        "StartTime": "2020-09-18T22:27:23.970000+00:00",
        "CreationTime": "2020-09-18T22:27:23.948000+00:00",
        "CompletionTime": "2020-09-18T22:28:21.197000+00:00",
        "Settings": {
            "ChannelIdentification": false,
            "ShowAlternatives": false
        },
        "IdentifyLanguage": true,
        "IdentifiedLanguageScore": 0.8672199249267578
    }
}
```

Untuk informasi selengkapnya, lihat Memulai (Antarmuka Baris AWS Perintah) di Panduan Pengembang Amazon Transcribe.

- Untuk detail API, lihat GetTranscriptionJobdi Referensi AWS CLI Perintah.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
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
```

- Untuk detail API, lihat GetTranscriptionJobdi AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`GetVocabulary`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`GetVocabulary`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Membuat dan menyempurnakan kosakata khusus

.NET

SDK untuk .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
/// <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;
}
```

- Untuk detail API, lihat GetVocabularydi Referensi AWS SDK untuk .NET API.

CLI

AWS CLI

Untuk mendapatkan informasi tentang kosakata khusus

`get-vocabulary`Contoh berikut mendapatkan informasi tentang kosakata kustom yang dibuat sebelumnya.

```
aws transcribe get-vocabulary \
    --vocabulary-name cli-vocab-1
```

Output:

```
{
    "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"
}
```

Untuk informasi selengkapnya, lihat Kosakata Khusus di Panduan Pengembang Amazon Transcribe.

- Untuk detail API, lihat GetVocabularydi Referensi AWS CLI Perintah.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
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
```

- Untuk detail API, lihat GetVocabularydi AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`ListMedicalTranscriptionJobs`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`ListMedicalTranscriptionJobs`.

.NET

SDK untuk .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
   /// <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;
    }
```

- Untuk detail API, lihat ListMedicalTranscriptionJobsdi Referensi AWS SDK untuk .NET API.

CLI

AWS CLI

Untuk membuat daftar pekerjaan transkripsi medis Anda

`list-medical-transcription-jobs`Contoh berikut mencantumkan pekerjaan transkripsi medis yang terkait dengan AWS akun dan Wilayah Anda. Untuk mendapatkan informasi lebih lanjut tentang pekerjaan transkripsi tertentu, salin nilai MedicalTranscriptionJobName parameter dalam output transkripsi, dan tentukan nilai itu untuk `MedicalTranscriptionJobName` opsi perintah. `get-medical-transcription-job` Untuk melihat lebih banyak pekerjaan transkripsi Anda, salin nilai NextToken parameter, jalankan `list-medical-transcription-jobs` perintah lagi, dan tentukan nilai itu di `--next-token` opsi.

```
aws transcribe list-medical-transcription-jobs
```

Output:

```
{
    "NextToken": "3/PblzkiGhzjER3KHuQt2fmbPLF7cDYafjFMEoGn44ON/
gsuUSTIkGyanvRE6WMXFd/ZTEc2EZj+P9eii/
z1O2FDYli6RLI0WoRX4RwMisVrh9G0Kie0Y8ikBCdtqlZB10Wa9McC+ebOl
+LaDtZPC4u6ttoHLRlEfzqstHXSgapXg3tEBtm9piIaPB6MOM5BB6t86+qtmocTR/
qrteHZBBudhTfbCwhsxaqujHiiUvFdm3BQbKKWIW06yV9b+4f38oD2lVIan
+vfUs3gBYAl5VTDmXXzQPBQOHPjtwmFI+IWX15nSUjWuN3TUylHgPWzDaYT8qBtu0Z+3UG4V6b
+K2CC0XszXg5rBq9hYgNzy4XoFh/6s5DoSnzq49Q9xHgHdT2yBADFmvFK7myZBsj75+2vQZOSVpWUPy3WT/32zFAc
+mFYfUjtTZ8n/jq7aQEjQ42A
+X/7K6JgOcdVPtEg8PlDr5kgYYG3q3OmYXX37U3FZuJmnTI63VtIXsNnOU5eGoYObtpk00Nq9UkzgSJxqj84ZD5n
+S0EGy9ZUYBJRRcGeYUM3Q4DbSJfUwSAqcFdLIWZdp8qIREMQIBWy7BLwSdyqsQo2vRrd53hm5aWM7SVf6pPq6X/
IXR5+1eUOOD8/coaTT4ES2DerbV6RkV4o0VT1d0SdVX/
MmtkNG8nYj8PqU07w7988quh1ZP6D80veJS1q73tUUR9MjnGernW2tAnvnLNhdefBcD
+sZVfYq3iBMFY7wTy1P1G6NqW9GrYDYoX3tTPWlD7phpbVSyKrh/
```

```
PdYrps5UxnsGoA1b7L/FfAXDfUoGrGUB4N3JsPYXX9D++g+6gV1qBBs/
WfF934aKqfD6UTggm/zV3GAOWiBpfvAZRvEb924i6yGHyMC7y54O1ZAwSBupmI
+FFd13CaPO4kN1vJlth6aM5vUPXg4BpyUhtbRhwD/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"
        }
    ]
}
```

Untuk informasi selengkapnya, lihat https://docs.aws.amazon.com/transcribe/ latest/dg/batch - med-transcription.html> di Panduan Pengembang Amazon Transcribe.

- Untuk detail API, lihat ListMedicalTranscriptionJobsdi Referensi AWS CLI Perintah.

JavaScript

SDK untuk JavaScript (v3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

Buat klien.

```
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 };
```

Daftar pekerjaan transkripsi medis.

```
// 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();
```

- Untuk informasi selengkapnya, lihat AWS SDK untuk JavaScript Panduan Developer.
- Untuk detail API, lihat ListMedicalTranscriptionJobsdi Referensi AWS SDK untuk JavaScript API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`ListTranscriptionJobs`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`ListTranscriptionJobs`.

.NET

SDK untuk .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
   /// <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;
   }
```

- Untuk detail API, lihat ListTranscriptionJobsdi Referensi AWS SDK untuk .NET API.

CLI

AWS CLI

Untuk membuat daftar pekerjaan transkripsi Anda

`list-transcription-jobs`Contoh berikut mencantumkan pekerjaan transkripsi yang terkait dengan AWS akun dan Wilayah Anda.

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

Output:

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

Untuk informasi selengkapnya, lihat Memulai (Antarmuka Baris AWS Perintah) di Panduan Pengembang Amazon Transcribe.

- Untuk detail API, lihat ListTranscriptionJobsdi Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
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("--------------");
               });
```

```
        }
    }
```

- Untuk detail API, lihat ListTranscriptionJobsdi Referensi AWS SDK for Java 2.x API.

JavaScript

SDK untuk JavaScript (v3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

Daftar pekerjaan transkripsi.

```
// 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();
```

Buat klien.

```
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 };
```

- Untuk informasi selengkapnya, lihat AWS SDK untuk JavaScript Panduan Developer.
- Untuk detail API, lihat ListTranscriptionJobsdi Referensi AWS SDK untuk JavaScript API.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
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
```

- Untuk detail API, lihat ListTranscriptionJobsdi AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`ListVocabularies`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`ListVocabularies`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Membuat dan menyempurnakan kosakata khusus

.NET

SDK untuk .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
    /// <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;
    }
```

- Untuk detail API, lihat ListVocabulariesdi Referensi AWS SDK untuk .NET API.

CLI

AWS CLI

Untuk membuat daftar kosakata kustom Anda

`list-vocabularies`Contoh berikut mencantumkan kosakata kustom yang terkait dengan AWS akun dan Wilayah Anda.

```
aws transcribe list-vocabularies
```

Output:

```
{
```

```
    "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"
        }
    ]
}
```


Untuk informasi selengkapnya, lihat Kosakata Khusus di Panduan Pengembang Amazon Transcribe.

- Untuk detail API, lihat ListVocabulariesdi Referensi AWS CLI Perintah.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
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
```

- Untuk detail API, lihat ListVocabulariesdi AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`StartMedicalTranscriptionJob`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`StartMedicalTranscriptionJob`.

.NET

SDK untuk .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
    /// <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;
    }
```

- Untuk detail API, lihat StartMedicalTranscriptionJobdi Referensi AWS SDK untuk .NET API.

CLI

AWS CLI

Contoh 1: Untuk mentranskripsikan dikte medis yang disimpan sebagai file audio

`start-medical-transcription-job`Contoh berikut mentranskripsikan file audio. Anda menentukan lokasi output transkripsi dalam `OutputBucketName` parameter.

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

```
    --cli-input-json file://myfile.json
```

Isi dari `myfile.json`:

```
{
    "MedicalTranscriptionJobName": "simple-dictation-medical-transcription-job",
    "LanguageCode": "language-code",
    "Specialty": "PRIMARYCARE",
    "Type": "DICTATION",
    "OutputBucketName":"amzn-s3-demo-bucket",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
    }
}
```

Output:

```
{
    "MedicalTranscriptionJob": {
        "MedicalTranscriptionJobName": "simple-dictation-medical-transcription-
job",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "language-code",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
        },
        "StartTime": "2020-09-20T00:35:22.256000+00:00",
        "CreationTime": "2020-09-20T00:35:22.218000+00:00",
        "Specialty": "PRIMARYCARE",
        "Type": "DICTATION"
    }
}
```

Untuk informasi selengkapnya, lihat Ikhtisar Transkripsi Batch di Panduan Pengembang Amazon Transcribe.

Contoh 2: Untuk mentranskripsikan dialog klinis-pasien yang disimpan sebagai file audio

`start-medical-transcription-job`Contoh berikut mentranskripsikan file audio yang berisi dialog klinis-pasien. Anda menentukan lokasi output transkripsi dalam OutputBucketName parameter.

```
aws transcribe start-medical-transcription-job \
    --cli-input-json file://mysecondfile.json
```

Isi dari `mysecondfile.json`:

```
{
    "MedicalTranscriptionJobName": "simple-dictation-medical-transcription-job",
    "LanguageCode": "language-code",
    "Specialty": "PRIMARYCARE",
    "Type": "CONVERSATION",
    "OutputBucketName":"amzn-s3-demo-bucket",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
    }
}
```

Output:

```
{
    "MedicalTranscriptionJob": {
        "MedicalTranscriptionJobName": "simple-conversation-medical-
transcription-job",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "language-code",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
        },
        "StartTime": "2020-09-20T23:19:49.965000+00:00",
        "CreationTime": "2020-09-20T23:19:49.941000+00:00",
        "Specialty": "PRIMARYCARE",
        "Type": "CONVERSATION"
    }
}
```

Untuk informasi selengkapnya, lihat Ikhtisar Transkripsi Batch di Panduan Pengembang Amazon Transcribe.

Contoh 3: Untuk mentranskripsikan file audio multisaluran dari dialog dokter-pasien

`start-medical-transcription-job`Contoh berikut mentranskripsikan audio dari setiap saluran dalam file audio dan menggabungkan transkripsi terpisah dari setiap

saluran menjadi satu output transkripsi. Anda menentukan lokasi output transkripsi dalam `OutputBucketName` parameter.

```
aws transcribe start-medical-transcription-job \
    --cli-input-json file://mythirdfile.json
```

Isi dari `mythirdfile.json`:

```
{
    "MedicalTranscriptionJobName": "multichannel-conversation-medical-
transcription-job",
    "LanguageCode": "language-code",
    "Specialty": "PRIMARYCARE",
    "Type": "CONVERSATION",
    "OutputBucketName":"amzn-s3-demo-bucket",
        "Media": {
          "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
        },
        "Settings":{
          "ChannelIdentification": true
        }
}
```

Output:

```
{
    "MedicalTranscriptionJob": {
        "MedicalTranscriptionJobName": "multichannel-conversation-medical-
transcription-job",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "language-code",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
        },
        "StartTime": "2020-09-20T23:46:44.081000+00:00",
        "CreationTime": "2020-09-20T23:46:44.053000+00:00",
        "Settings": {
            "ChannelIdentification": true
        },
        "Specialty": "PRIMARYCARE",
        "Type": "CONVERSATION"
    }
```

```
}
```

Untuk informasi selengkapnya, lihat Identifikasi Saluran di Panduan Pengembang Amazon Transcribe.

Contoh 4: Untuk mentranskripsikan file audio dari dialog klinis-pasien dan mengidentifikasi pembicara dalam output transkripsi

`start-medical-transcription-job`Contoh berikut mentranskripsikan file audio dan memberi label ucapan setiap pembicara dalam output transkripsi. Anda menentukan lokasi output transkripsi dalam `OutputBucketName` parameter.

```
aws transcribe start-medical-transcription-job \
    --cli-input-json file://myfourthfile.json
```

Isi dari `myfourthfile.json`:

```
{
    "MedicalTranscriptionJobName": "speaker-id-conversation-medical-
transcription-job",
    "LanguageCode": "language-code",
    "Specialty": "PRIMARYCARE",
    "Type": "CONVERSATION",
    "OutputBucketName":"amzn-s3-demo-bucket",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
        },
    "Settings":{
        "ShowSpeakerLabels": true,
        "MaxSpeakerLabels": 2
        }
}
```

Output:

```
{
    "MedicalTranscriptionJob": {
        "MedicalTranscriptionJobName": "speaker-id-conversation-medical-
transcription-job",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "language-code",
```

```
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
        },
        "StartTime": "2020-09-21T18:43:37.265000+00:00",
        "CreationTime": "2020-09-21T18:43:37.157000+00:00",
        "Settings": {
            "ShowSpeakerLabels": true,
            "MaxSpeakerLabels": 2
        },
        "Specialty": "PRIMARYCARE",
        "Type": "CONVERSATION"
    }
}
```

Untuk informasi selengkapnya, lihat Mengidentifikasi Pembicara di Panduan Pengembang Amazon Transcribe.

Contoh 5: Untuk mentranskripsikan percakapan medis yang disimpan sebagai file audio dengan hingga dua alternatif transkripsi

`start-medical-transcription-job`Contoh berikut membuat hingga dua transkripsi alternatif dari satu file audio. Setiap transkripsi memiliki tingkat kepercayaan yang terkait dengannya. Secara default, Amazon Transcribe mengembalikan transkripsi dengan tingkat kepercayaan tertinggi. Anda dapat menentukan bahwa Amazon Transcribe mengembalikan transkripsi tambahan dengan tingkat kepercayaan yang lebih rendah. Anda menentukan lokasi output transkripsi dalam `OutputBucketName` parameter.

```
aws transcribe start-medical-transcription-job \
    --cli-input-json file://myfifthfile.json
```

Isi dari `myfifthfile.json`:

```
{
    "MedicalTranscriptionJobName": "alternatives-conversation-medical-
transcription-job",
    "LanguageCode": "language-code",
    "Specialty": "PRIMARYCARE",
    "Type": "CONVERSATION",
    "OutputBucketName":"amzn-s3-demo-bucket",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
```

```
    },
    "Settings":{
        "ShowAlternatives": true,
        "MaxAlternatives": 2
    }
}
```

Output:

```
{
    "MedicalTranscriptionJob": {
        "MedicalTranscriptionJobName": "alternatives-conversation-medical-
transcription-job",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "language-code",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
        },
        "StartTime": "2020-09-21T19:09:18.199000+00:00",
        "CreationTime": "2020-09-21T19:09:18.171000+00:00",
        "Settings": {
            "ShowAlternatives": true,
            "MaxAlternatives": 2
        },
        "Specialty": "PRIMARYCARE",
        "Type": "CONVERSATION"
    }
}
```

Untuk informasi selengkapnya, lihat Transkripsi Alternatif di Panduan Pengembang Amazon Transcribe.

Contoh 6: Untuk mentranskripsikan file audio dikte medis dengan hingga dua transkripsi alternatif

`start-medical-transcription-job`Contoh berikut mentranskripsikan file audio dan menggunakan filter kosakata untuk menutupi kata-kata yang tidak diinginkan. Anda menentukan lokasi output transkripsi dalam OutputBucketName parameter.

```
aws transcribe start-medical-transcription-job \
    --cli-input-json file://mysixthfile.json
```

Isi dari `mysixthfile.json`:

```
{
    "MedicalTranscriptionJobName": "alternatives-conversation-medical-
transcription-job",
    "LanguageCode": "language-code",
    "Specialty": "PRIMARYCARE",
    "Type": "DICTATION",
    "OutputBucketName":"amzn-s3-demo-bucket",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
    },
    "Settings":{
          "ShowAlternatives": true,
          "MaxAlternatives": 2
    }
}
```

Output:

```
{
    "MedicalTranscriptionJob": {
        "MedicalTranscriptionJobName": "alternatives-dictation-medical-
transcription-job",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "language-code",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
        },
        "StartTime": "2020-09-21T21:01:14.592000+00:00",
        "CreationTime": "2020-09-21T21:01:14.569000+00:00",
        "Settings": {
            "ShowAlternatives": true,
            "MaxAlternatives": 2
        },
        "Specialty": "PRIMARYCARE",
        "Type": "DICTATION"
    }
}
```

Untuk informasi selengkapnya, lihat Transkripsi Alternatif di Panduan Pengembang Amazon Transcribe.

Contoh 7: Untuk mentranskripsikan file audio dikte medis dengan akurasi yang meningkat dengan menggunakan kosakata khusus

`start-medical-transcription-job`Contoh berikut mentranskripsikan file audio dan menggunakan kosakata khusus medis yang telah Anda buat sebelumnya untuk meningkatkan akurasi transkripsi. Anda menentukan lokasi output transkripsi dalam `OutputBucketName` parameter.

```
aws transcribe start-transcription-job \
    --cli-input-json file://myseventhfile.json
```

Isi dari `mysixthfile.json`:

```
{
    "MedicalTranscriptionJobName": "vocabulary-dictation-medical-transcription-
job",
    "LanguageCode": "language-code",
    "Specialty": "PRIMARYCARE",
    "Type": "DICTATION",
    "OutputBucketName":"amzn-s3-demo-bucket",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
    },
    "Settings":{
        "VocabularyName": "cli-medical-vocab-1"
    }
}
```

Output:

```
{
    "MedicalTranscriptionJob": {
        "MedicalTranscriptionJobName": "vocabulary-dictation-medical-
transcription-job",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "language-code",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.extension"
        },
        "StartTime": "2020-09-21T21:17:27.045000+00:00",
        "CreationTime": "2020-09-21T21:17:27.016000+00:00",
        "Settings": {
```

```
            "VocabularyName": "cli-medical-vocab-1"
        },
        "Specialty": "PRIMARYCARE",
        "Type": "DICTATION"
    }
}
```

Untuk informasi selengkapnya, lihat Kosakata Kustom Medis di Panduan Pengembang Amazon Transcribe.

- Untuk detail API, lihat StartMedicalTranscriptionJobdi Referensi AWS CLI Perintah.

JavaScript

SDK untuk JavaScript (v3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

Buat klien.

```
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 };
```

Mulai pekerjaan transkripsi medis.

```
// 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();
```

- Untuk informasi selengkapnya, lihat AWS SDK untuk JavaScript Panduan Developer.
- Untuk detail API, lihat StartMedicalTranscriptionJobdi Referensi AWS SDK untuk JavaScript API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`StartTranscriptionJob`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`StartTranscriptionJob`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Membuat dan menyempurnakan kosakata khusus
- Transcribe audio dan dapatkan data pekerjaan

.NET

SDK untuk .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
    /// <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;
    }
```

- Untuk detail API, lihat StartTranscriptionJobdi Referensi AWS SDK untuk .NET API.

CLI

AWS CLI

Contoh 1: Untuk mentranskripsikan file audio

`start-transcription-job`Contoh berikut mentranskripsikan file audio Anda.

```
aws transcribe start-transcription-job \
    --cli-input-json file://myfile.json
```

Isi dari `myfile.json`:

```
{
    "TranscriptionJobName": "cli-simple-transcription-job",
    "LanguageCode": "the-language-of-your-transcription-job",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/Amazon-S3-prefix/your-media-
file-name.file-extension"
    }
}
```

Untuk informasi selengkapnya, lihat Memulai (Antarmuka Baris AWS Perintah) di Panduan Pengembang Amazon Transcribe.

Contoh 2: Untuk mentranskripsikan file audio multi-saluran

`start-transcription-job`Contoh berikut mentranskripsikan file audio multi-saluran Anda.

```
aws transcribe start-transcription-job \
    --cli-input-json file://mysecondfile.json
```

Isi dari `mysecondfile.json`:

```
{
    "TranscriptionJobName": "cli-channelid-job",
    "LanguageCode": "the-language-of-your-transcription-job",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/Amazon-S3-prefix/your-media-
file-name.file-extension"
    },
    "Settings":{
        "ChannelIdentification":true
    }
}
```

Output:

```
{
    "TranscriptionJob": {
        "TranscriptionJobName": "cli-channelid-job",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "the-language-of-your-transcription-job",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/Amazon-S3-prefix/your-
media-file-name.file-extension"
        },
        "StartTime": "2020-09-17T16:07:56.817000+00:00",
        "CreationTime": "2020-09-17T16:07:56.784000+00:00",
        "Settings": {
            "ChannelIdentification": true
        }
    }
}
```

Untuk informasi selengkapnya, lihat Mentranskripsikan Audio Multi-Saluran di Panduan Pengembang Amazon Transcribe.

Contoh 3: Untuk mentranskripsikan file audio dan mengidentifikasi speaker yang berbeda

`start-transcription-job`Contoh berikut mentranskripsikan file audio Anda dan mengidentifikasi speaker dalam output transkripsi.

```
aws transcribe start-transcription-job \
    --cli-input-json file://mythirdfile.json
```

Isi dari `mythirdfile.json`:

```
{
    "TranscriptionJobName": "cli-speakerid-job",
    "LanguageCode": "the-language-of-your-transcription-job",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/Amazon-S3-prefix/your-media-
file-name.file-extension"
    },
    "Settings":{
    "ShowSpeakerLabels": true,
    "MaxSpeakerLabels": 2
    }
}
```

Output:

```
{
    "TranscriptionJob": {
        "TranscriptionJobName": "cli-speakerid-job",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "the-language-of-your-transcription-job",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/Amazon-S3-prefix/your-
media-file-name.file-extension"
        },
        "StartTime": "2020-09-17T16:22:59.696000+00:00",
        "CreationTime": "2020-09-17T16:22:59.676000+00:00",
        "Settings": {
            "ShowSpeakerLabels": true,
            "MaxSpeakerLabels": 2
        }
    }
}
```

Untuk informasi selengkapnya, lihat Mengidentifikasi Pembicara di Panduan Pengembang Amazon Transcribe.

Contoh 4: Untuk mentranskripsikan file audio dan menutupi kata-kata yang tidak diinginkan dalam output transkripsi

`start-transcription-job`Contoh berikut mentranskripsikan file audio Anda dan menggunakan filter kosakata yang sebelumnya Anda buat untuk menutupi kata-kata yang tidak diinginkan.

```
aws transcribe start-transcription-job \
    --cli-input-json file://myfourthfile.json
```

Isi dari `myfourthfile.json`:

```
{
    "TranscriptionJobName": "cli-filter-mask-job",
    "LanguageCode": "the-language-of-your-transcription-job",
    "Media": {
          "MediaFileUri": "s3://amzn-s3-demo-bucket/Amazon-S3-prefix/your-media-
file-name.file-extension"
    },
    "Settings":{
        "VocabularyFilterName": "your-vocabulary-filter",
        "VocabularyFilterMethod": "mask"
    }
}
```

Output:

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

Untuk informasi selengkapnya, lihat Memfilter Transkripsi di Panduan Pengembang Amazon Transcribe.

Contoh 5: Untuk mentranskripsikan file audio dan menghapus kata-kata yang tidak diinginkan dalam output transkripsi

`start-transcription-job`Contoh berikut mentranskripsikan file audio Anda dan menggunakan filter kosakata yang sebelumnya Anda buat untuk menutupi kata-kata yang tidak diinginkan.

```
aws transcribe start-transcription-job \
    --cli-input-json file://myfifthfile.json
```

Isi dari `myfifthfile.json`:

```
{
    "TranscriptionJobName": "cli-filter-remove-job",
    "LanguageCode": "the-language-of-your-transcription-job",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/Amazon-S3-prefix/your-media-
file-name.file-extension"
    },
    "Settings":{
        "VocabularyFilterName": "your-vocabulary-filter",
        "VocabularyFilterMethod": "remove"
    }
}
```

Output:

```
{
    "TranscriptionJob": {
        "TranscriptionJobName": "cli-filter-remove-job",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "the-language-of-your-transcription-job",
        "Media": {
```

```
            "MediaFileUri": "s3://amzn-s3-demo-bucket/Amazon-S3-prefix/your-
media-file-name.file-extension"
        },
        "StartTime": "2020-09-18T16:36:18.568000+00:00",
        "CreationTime": "2020-09-18T16:36:18.547000+00:00",
        "Settings": {
            "VocabularyFilterName": "your-vocabulary-filter",
            "VocabularyFilterMethod": "remove"
        }
    }
}
```

Untuk informasi selengkapnya, lihat Memfilter Transkripsi di Panduan Pengembang Amazon Transcribe.

Contoh 6: Untuk mentranskripsikan file audio dengan akurasi yang meningkat menggunakan kosakata khusus

`start-transcription-job`Contoh berikut mentranskripsikan file audio Anda dan menggunakan filter kosakata yang sebelumnya Anda buat untuk menutupi kata-kata yang tidak diinginkan.

```
aws transcribe start-transcription-job \
    --cli-input-json file://mysixthfile.json
```

Isi dari `mysixthfile.json`:

```
{
    "TranscriptionJobName": "cli-vocab-job",
    "LanguageCode": "the-language-of-your-transcription-job",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/Amazon-S3-prefix/your-media-
file-name.file-extension"
    },
    "Settings":{
        "VocabularyName": "your-vocabulary"
    }
}
```

Output:

```
{
```

```
    "TranscriptionJob": {
        "TranscriptionJobName": "cli-vocab-job",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "the-language-of-your-transcription-job",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/Amazon-S3-prefix/your-
media-file-name.file-extension"
        },
        "StartTime": "2020-09-18T16:36:18.568000+00:00",
        "CreationTime": "2020-09-18T16:36:18.547000+00:00",
        "Settings": {
            "VocabularyName": "your-vocabulary"
        }
    }
}
```

Untuk informasi selengkapnya, lihat Memfilter Transkripsi di Panduan Pengembang Amazon Transcribe.

Contoh 7: Untuk mengidentifikasi bahasa file audio dan menuliskannya

`start-transcription-job`Contoh berikut mentranskripsikan file audio Anda dan menggunakan filter kosakata yang sebelumnya Anda buat untuk menutupi kata-kata yang tidak diinginkan.

```
aws transcribe start-transcription-job \
    --cli-input-json file://myseventhfile.json
```

Isi dari `myseventhfile.json`:

```
{
    "TranscriptionJobName": "cli-identify-language-transcription-job",
    "IdentifyLanguage": true,
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/Amazon-S3-prefix/your-media-
file-name.file-extension"
    }
}
```

Output:

```
{
```

```
    "TranscriptionJob": {
        "TranscriptionJobName": "cli-identify-language-transcription-job",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/Amazon-S3-prefix/your-
media-file-name.file-extension"
        },
        "StartTime": "2020-09-18T22:27:23.970000+00:00",
        "CreationTime": "2020-09-18T22:27:23.948000+00:00",
        "IdentifyLanguage": true
    }
}
```

Untuk informasi selengkapnya, lihat Mengidentifikasi Bahasa di Panduan Pengembang Amazon Transcribe.

Contoh 8: Untuk mentranskripsikan file audio dengan informasi yang dapat diidentifikasi secara pribadi disunting

`start-transcription-job`Contoh berikut mentranskripsikan file audio Anda dan menyunting informasi identitas pribadi apa pun dalam keluaran transkripsi.

```
aws transcribe start-transcription-job \
    --cli-input-json file://myeighthfile.json
```

Isi dari `myeigthfile.json`:

```
{
    "TranscriptionJobName": "cli-redaction-job",
    "LanguageCode": "language-code",
    "Media": {
        "MediaFileUri": "s3://Amazon-S3-Prefix/your-media-file.file-extension"
    },
    "ContentRedaction": {
        "RedactionOutput":"redacted",
        "RedactionType":"PII"
    }
}
```

Output:

```
{
```

```
    "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"
        }
    }
}
```

Untuk informasi selengkapnya, lihat Redaksi Konten Otomatis di Panduan Pengembang Amazon Transcribe.

Contoh 9: Untuk menghasilkan transkrip dengan informasi identitas pribadi (PII) yang disunting dan transkrip yang tidak disunting

`start-transcription-job`Contoh berikut menghasilkan dua transkripsi file audio Anda, satu dengan informasi yang dapat diidentifikasi secara pribadi disunting, dan yang lainnya tanpa redaksi apa pun.

```
aws transcribe start-transcription-job \
    --cli-input-json file://myninthfile.json
```

Isi dari `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"
    }
```

```
}
```

Output:

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

Untuk informasi selengkapnya, lihat Redaksi Konten Otomatis di Panduan Pengembang Amazon Transcribe.

Contoh 10: Untuk menggunakan model bahasa kustom yang sebelumnya Anda buat untuk mentranskripsikan file audio.

`start-transcription-job`Contoh berikut mentranskripsikan file audio Anda dengan model bahasa khusus yang telah Anda buat sebelumnya.

```
aws transcribe start-transcription-job \
    --cli-input-json file://mytenthfile.json
```

Isi dari `mytenthfile.json`:

```
{
    "TranscriptionJobName": "cli-clm-2-job-1",
    "LanguageCode": "language-code",
    "Media": {
        "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.file-extension"
    },
```

```
    "ModelSettings": {
        "LanguageModelName":"cli-clm-2"
    }
}
```

Output:

```
{
    "TranscriptionJob": {
        "TranscriptionJobName": "cli-clm-2-job-1",
        "TranscriptionJobStatus": "IN_PROGRESS",
        "LanguageCode": "language-code",
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/your-audio-file.file-
extension"
        },
        "StartTime": "2020-09-28T17:56:01.835000+00:00",
        "CreationTime": "2020-09-28T17:56:01.801000+00:00",
        "ModelSettings": {
            "LanguageModelName": "cli-clm-2"
        }
    }
}
```

Untuk informasi selengkapnya, lihat Meningkatkan Akurasi Transkripsi Khusus Domain dengan Model Bahasa Khusus di Panduan Pengembang Amazon Transcribe.

- Untuk detail API, lihat StartTranscriptionJobdi Referensi AWS CLI Perintah.

JavaScript

SDK untuk JavaScript (v3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

Mulai pekerjaan transkripsi.

```
// 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();
```

Buat klien.

```
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 };
```

- Untuk informasi selengkapnya, lihat AWS SDK untuk JavaScript Panduan Developer.
- Untuk detail API, lihat StartTranscriptionJobdi Referensi AWS SDK untuk JavaScript API.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
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
```

- Untuk detail API, lihat StartTranscriptionJobdi AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`UpdateVocabulary`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`UpdateVocabulary`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Membuat dan menyempurnakan kosakata khusus

.NET

SDK untuk .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
    /// <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;
    }
```

- Untuk detail API, lihat UpdateVocabularydi Referensi AWS SDK untuk .NET API.

CLI

AWS CLI

Untuk memperbarui kosakata khusus dengan istilah baru.

`update-vocabulary`Contoh berikut menimpa istilah yang digunakan untuk membuat kosakata khusus dengan yang baru yang Anda berikan. Prasyarat: untuk mengganti istilah dalam kosakata khusus, Anda memerlukan file dengan istilah baru.

```
aws transcribe update-vocabulary \
    --vocabulary-file-uri s3://amzn-s3-demo-bucket/Amazon-S3-Prefix/custom-
vocabulary.txt \
    --vocabulary-name custom-vocabulary \
    --language-code language-code
```

Output:

```
{
    "VocabularyName": "custom-vocabulary",
    "LanguageCode": "language",
    "VocabularyState": "PENDING"
}
```

Untuk informasi selengkapnya, lihat Kosakata Khusus di Panduan Pengembang Amazon Transcribe.

- Untuk detail API, lihat UpdateVocabularydi Referensi AWS CLI Perintah.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
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
```

- Untuk detail API, lihat UpdateVocabularydi AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Skenario untuk Amazon Transcribe menggunakan AWS SDKs

Contoh kode berikut menunjukkan cara mengimplementasikan skenario umum di Amazon Transcribe with. AWS SDKs Skenario ini menunjukkan kepada Anda cara menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam Amazon Transcribe atau digabungkan dengan yang lain. Layanan AWS Setiap skenario menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode.

Skenario menargetkan tingkat pengalaman menengah untuk membantu Anda memahami tindakan layanan dalam konteks.

Contoh

- Membangun aplikasi streaming Amazon Transcribe
- Mengonversi teks menjadi ucapan dan kembali ke teks menggunakan AWS SDK
- Membuat dan menyempurnakan kosakata kustom Amazon Transcribe menggunakan SDK AWS
- Transcribe audio dan dapatkan data pekerjaan dengan Amazon Transcribe menggunakan SDK AWS

Membangun aplikasi streaming Amazon Transcribe

Contoh kode berikut menunjukkan cara membuat aplikasi yang merekam, mentranskripsikan, dan menerjemahkan audio langsung secara real-time, dan mengirim email hasilnya.

JavaScript

SDK untuk JavaScript (v3)

Menunjukkan cara menggunakan Amazon Transcribe untuk membuat aplikasi yang merekam, menyalin, dan menerjemahkan audio langsung secara real-time, dan mengirim email hasilnya menggunakan Amazon Simple Email Service (Amazon SES).

Untuk kode sumber lengkap dan instruksi tentang cara mengatur dan menjalankan, lihat contoh lengkapnya di GitHub.

Layanan yang digunakan dalam contoh ini

- Amazon Comprehend
- Amazon SES
- Amazon Transcribe
- Amazon Translate

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Mengonversi teks menjadi ucapan dan kembali ke teks menggunakan AWS SDK

Contoh kode berikut ini menunjukkan cara:

- Menggunakan Amazon Polly untuk mensintesis file input teks biasa (UTF-8) ke file audio.
- Mengunggah file audio ke bucket Amazon S3.
- Menggunakan Amazon Transcribe untuk mengonversi file audio menjadi teks.
- Tampilkan teks.

Rust

SDK untuk Rust

Gunakan Amazon Polly untuk mensintesis file input teks biasa (UTF-8) menjadi file audio, unggah file audio ke bucket Amazon S3, gunakan Amazon Transcribe untuk mengonversi file audio tersebut menjadi teks, dan tampilkan teksnya.

Untuk kode sumber lengkap dan instruksi tentang cara mengatur dan menjalankan, lihat contoh lengkapnya di GitHub.

Layanan yang digunakan dalam contoh ini

- Amazon Polly
- Amazon S3
- Amazon Transcribe

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Membuat dan menyempurnakan kosakata kustom Amazon Transcribe menggunakan SDK AWS

Contoh kode berikut ini menunjukkan cara:

- Unggah file audio ke Amazon S3.
- Jalankan pekerjaan Amazon Transcribe untuk menyalin file dan mendapatkan hasilnya.
- Buat dan perbaiki kosakata khusus untuk meningkatkan akurasi transkripsi.
- Jalankan pekerjaan dengan kosakata khusus dan dapatkan hasilnya.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

Transcribe file audio yang berisi pembacaan Jabberwocky oleh Lewis Carroll. Mulailah dengan membuat fungsi yang membungkus tindakan Amazon Transcribe.

```
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
```

Panggil fungsi pembungkus untuk mentranskripsikan audio tanpa kosakata khusus dan kemudian dengan versi berbeda dari kosakata khusus untuk melihat hasil yang lebih baik.

```
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!")
```

- Untuk detail API, lihat topik berikut ini adalah Referensi API SDK untuk Python (Boto3)AWS .
  - CreateVocabulary
  - DeleteTranscriptionJob
  - DeleteVocabulary
  - GetTranscriptionJob
  - GetVocabulary
  - ListVocabularies
  - StartTranscriptionJob
  - UpdateVocabulary

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Transcribe audio dan dapatkan data pekerjaan dengan Amazon Transcribe menggunakan SDK AWS

Contoh kode berikut ini menunjukkan cara:

- Mulai pekerjaan transkripsi dengan Amazon Transcribe.
- Tunggu hingga tugas selesai.

- Dapatkan URI tempat transkrip disimpan.

Untuk informasi selengkapnya, lihat Memulai Amazon Transcribe.

Java

SDK untuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

Mentranskripsikan file PCM.

```
/**
 * 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();
        }
    }
}
```

Mentranskripsikan audio streaming dari mikrofon komputer Anda.

```
public class TranscribeStreamingDemoApp {
    private static final Region REGION = Region.US_EAST_1;
    private static TranscribeStreamingAsyncClient client;

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

        client = TranscribeStreamingAsyncClient.builder()
                .credentialsProvider(getCredentials())
                .region(REGION)
```

```
                .build();

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

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

    private static InputStream getStreamFromMic() throws LineUnavailableException
{

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

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

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

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

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

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

```
    private static StartStreamTranscriptionResponseHandler getResponseHandler() {
        return StartStreamTranscriptionResponseHandler.builder()
                .onResponse(r -> {
                    System.out.println("Received Initial response");
                })
                .onError(e -> {
                    System.out.println(e.getMessage());
                    StringWriter sw = new StringWriter();
                    e.printStackTrace(new PrintWriter(sw));
                    System.out.println("Error Occurred: " + sw);
                })
                .onComplete(() -> {
                    System.out.println("=== All records stream successfully
 ===");
                })
                .subscriber(event -> {
                    List<Result> results = ((TranscriptEvent)
 event).transcript().results();
                    if (results.size() > 0) {
                        if (!
results.get(0).alternatives().get(0).transcript().isEmpty()) {

 System.out.println(results.get(0).alternatives().get(0).transcript());
                        }
                    }
                })
                .build();
    }


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

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

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

            if (currentSubscription == null) {
                currentSubscription = new SubscriptionImpl(s, inputStream);
```

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

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

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

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

            demand.getAndAdd(n);

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

```
                    subscriber.onError(e);
                }
            });
        }

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

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

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

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

            return audioBuffer;
        }

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

- Untuk detail API, lihat topik berikut di Referensi API AWS SDK for Java 2.x .
  - GetTranscriptionJob
  - StartTranscriptionJob

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
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()
```

- Untuk detail API, lihat topik berikut ini adalah Referensi API SDK untuk Python (Boto3)AWS .
  - GetTranscriptionJob
  - StartTranscriptionJob

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Keamanan di Amazon Transcribe

Keamanan cloud di AWS adalah prioritas tertinggi. Sebagai AWS pelanggan, Anda mendapat manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan.

Keamanan adalah tanggung jawab bersama antara Anda AWS dan Anda. Model tanggung jawab bersama menjelaskan hal ini sebagai keamanan cloud dan keamanan dalam cloud:

- Keamanan cloud: AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan AWS layanan di AWS Cloud. AWS juga memberi Anda layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga secara teratur menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari Program AWS Kepatuhan Program AWS Kepatuhan . Untuk mempelajari tentang program kepatuhan yang berlaku Amazon Transcribe, lihat AWS Layanan dalam Lingkup oleh AWS Layanan Program Kepatuhan .
- Keamanan di cloud: Tanggung jawab Anda ditentukan oleh AWS layanan yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain termasuk sensitivitas data Anda, persyaratan perusahaan Anda, serta hukum dan regulasi yang berlaku.

Dokumentasi ini akan membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan Amazon Transcribe. Topik berikut menunjukkan cara mengonfigurasi Amazon Transcribe untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga belajar cara menggunakan AWS layanan lain untuk memantau dan mengamankan Amazon Transcribe sumber daya Anda.

Topik

- Identity and Access Management untuk Amazon Transcribe
- Perlindungan data di Amazon Transcribe
- Pemantauan Amazon Transcribe
- Validasi kepatuhan untuk Amazon Transcribe
- Ketahanan di Amazon Transcribe
- Keamanan infrastruktur di Amazon Transcribe
- Analisis dan manajemen kerentanan di Amazon Transcribe
- Praktik terbaik keamanan untuk Amazon Transcribe

Identity and Access Management untuk Amazon Transcribe

AWS Identity and Access Management (IAM) adalah Layanan AWS yang membantu administrator mengontrol akses ke AWS sumber daya dengan aman. Administrator IAM mengontrol siapa yang dapat diautentikasi (masuk) dan diberi wewenang (memiliki izin) untuk menggunakan sumber daya. Amazon Transcribe IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Topik

- Audiens
- Mengautentikasi dengan identitas
- Mengelola akses menggunakan kebijakan
- Bagaimana Amazon Transcribe bekerja dengan IAM
- Pencegahan "confused deputy" lintas layanan
- Amazon Transcribe contoh kebijakan berbasis identitas
- Memecahkan masalah Amazon Transcribe identitas dan akses

Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda, tergantung pada pekerjaan yang Anda lakukan. Amazon Transcribe

Pengguna layanan — Jika Anda menggunakan Amazon Transcribe layanan untuk melakukan pekerjaan Anda, administrator Anda memberi Anda kredensi dan izin yang Anda butuhkan. Saat Anda menggunakan lebih banyak Amazon Transcribe fitur untuk melakukan pekerjaan Anda, Anda mungkin memerlukan izin tambahan. Memahami cara mengelola akses dapat membantu Anda meminta izin yang tepat dari administrator Anda. Jika Anda tidak dapat mengakses fitur di Amazon Transcribe, lihat Memecahkan masalah Amazon Transcribe identitas dan akses.

Administrator layanan — Jika Anda bertanggung jawab atas Amazon Transcribe sumber daya di perusahaan Anda, Anda mungkin memiliki akses penuh ke Amazon Transcribe. Tugas Anda adalah menentukan Amazon Transcribe fitur dan sumber daya mana yang harus diakses pengguna layanan Anda. Kemudian, Anda harus mengirimkan permintaan kepada administrator IAM untuk mengubah izin pengguna layanan Anda. Tinjau informasi di halaman ini untuk memahami konsep dasar IAM. Untuk mempelajari lebih lanjut tentang bagaimana perusahaan Anda dapat menggunakan IAM Amazon Transcribe, lihatBagaimana Amazon Transcribe bekerja dengan IAM.

Administrator IAM – Jika Anda adalah administrator IAM, Anda mungkin ingin belajar dengan lebih detail tentang cara Anda menulis kebijakan untuk mengelola akses ke Amazon Transcribe. Untuk melihat contoh kebijakan Amazon Transcribe berbasis identitas yang dapat Anda gunakan di IAM, lihat. Amazon Transcribe contoh kebijakan berbasis identitas

Mengautentikasi dengan identitas

Otentikasi adalah cara Anda masuk AWS menggunakan kredensi identitas Anda. Anda harus diautentikasi (masuk ke AWS) sebagai Pengguna root akun AWS, sebagai pengguna IAM, atau dengan mengasumsikan peran IAM.

Anda dapat masuk AWS sebagai identitas federasi dengan menggunakan kredensil yang disediakan melalui sumber identitas. AWS IAM Identity Center Pengguna (IAM Identity Center), autentikasi masuk tunggal perusahaan Anda, dan kredensi Google atau Facebook Anda adalah contoh identitas federasi. Saat Anda masuk sebagai identitas terfederasi, administrator Anda sebelumnya menyiapkan federasi identitas menggunakan peran IAM. Ketika Anda mengakses AWS dengan menggunakan federasi, Anda secara tidak langsung mengambil peran.

Bergantung pada jenis pengguna Anda, Anda dapat masuk ke AWS Management Console atau portal AWS akses. Untuk informasi selengkapnya tentang masuk AWS, lihat Cara masuk ke Panduan AWS Sign-In Pengguna Anda Akun AWS.

Jika Anda mengakses AWS secara terprogram, AWS sediakan kit pengembangan perangkat lunak (SDK) dan antarmuka baris perintah (CLI) untuk menandatangani permintaan Anda secara kriptografis dengan menggunakan kredensil Anda. Jika Anda tidak menggunakan AWS alat, Anda harus menandatangani permintaan sendiri. Guna mengetahui informasi selengkapnya tentang penggunaan metode yang disarankan untuk menandatangani permintaan sendiri, lihat AWS Signature Version 4 untuk permintaan API dalam Panduan Pengguna IAM.

Apa pun metode autentikasi yang digunakan, Anda mungkin diminta untuk menyediakan informasi keamanan tambahan. Misalnya, AWS merekomendasikan agar Anda menggunakan otentikasi multi-faktor (MFA) untuk meningkatkan keamanan akun Anda. Untuk mempelajari selengkapnya, lihat Autentikasi multi-faktor dalam Panduan Pengguna AWS IAM Identity Center dan Autentikasi multi-faktor AWS di IAM dalam Panduan Pengguna IAM.

Akun AWS pengguna root

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya di akun. Identitas ini disebut pengguna Akun AWS root dan diakses dengan masuk dengan alamat email dan kata sandi yang Anda gunakan

untuk membuat akun. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari. Lindungi kredensial pengguna root Anda dan gunakan kredensial tersebut untuk melakukan tugas yang hanya dapat dilakukan pengguna root. Untuk daftar lengkap tugas yang mengharuskan Anda masuk sebagai pengguna root, lihat Tugas yang memerlukan kredensial pengguna root dalam Panduan Pengguna IAM.

Identitas gabungan

Sebagai praktik terbaik, mewajibkan pengguna manusia, termasuk pengguna yang memerlukan akses administrator, untuk menggunakan federasi dengan penyedia identitas untuk mengakses Layanan AWS dengan menggunakan kredensi sementara.

Identitas federasi adalah pengguna dari direktori pengguna perusahaan Anda, penyedia identitas web, direktori Pusat Identitas AWS Directory Service, atau pengguna mana pun yang mengakses Layanan AWS dengan menggunakan kredensil yang disediakan melalui sumber identitas. Ketika identitas federasi mengakses Akun AWS, mereka mengambil peran, dan peran memberikan kredensi sementara.

Untuk manajemen akses terpusat, kami sarankan Anda menggunakan AWS IAM Identity Center. Anda dapat membuat pengguna dan grup di Pusat Identitas IAM, atau Anda dapat menghubungkan dan menyinkronkan ke sekumpulan pengguna dan grup di sumber identitas Anda sendiri untuk digunakan di semua aplikasi Akun AWS dan aplikasi Anda. Untuk informasi tentang Pusat Identitas IAM, lihat Apakah itu Pusat Identitas IAM? dalam Panduan Pengguna AWS IAM Identity Center .

Pengguna dan grup IAM

Pengguna IAM adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus untuk satu orang atau aplikasi. Jika memungkinkan, kami merekomendasikan untuk mengandalkan kredensial sementara, bukan membuat pengguna IAM yang memiliki kredensial jangka panjang seperti kata sandi dan kunci akses. Namun, jika Anda memiliki kasus penggunaan tertentu yang memerlukan kredensial jangka panjang dengan pengguna IAM, kami merekomendasikan Anda merotasi kunci akses. Untuk informasi selengkapnya, lihat Merotasi kunci akses secara teratur untuk kasus penggunaan yang memerlukan kredensial jangka panjang dalam Panduan Pengguna IAM.

Grup IAM adalah identitas yang menentukan sekumpulan pengguna IAM. Anda tidak dapat masuk sebagai grup. Anda dapat menggunakan grup untuk menentukan izin bagi beberapa pengguna sekaligus. Grup mempermudah manajemen izin untuk sejumlah besar pengguna sekaligus. Misalnya, Anda dapat meminta kelompok untuk menyebutkan IAMAdmins dan memberikan izin kepada grup tersebut untuk mengelola sumber daya IAM.

Pengguna berbeda dari peran. Pengguna secara unik terkait dengan satu orang atau aplikasi, tetapi peran dimaksudkan untuk dapat digunakan oleh siapa pun yang membutuhkannya. Pengguna memiliki kredensial jangka panjang permanen, tetapi peran memberikan kredensial sementara. Untuk mempelajari selengkapnya, lihat Kasus penggunaan untuk pengguna IAM dalam Panduan Pengguna IAM.

Peran IAM

Peran IAM adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus. Peran ini mirip dengan pengguna IAM, tetapi tidak terkait dengan orang tertentu. Untuk mengambil peran IAM sementara AWS Management Console, Anda dapat beralih dari pengguna ke peran IAM (konsol). Anda dapat mengambil peran dengan memanggil operasi AWS CLI atau AWS API atau dengan menggunakan URL kustom. Untuk informasi selengkapnya tentang cara menggunakan peran, lihat Metode untuk mengambil peran dalam Panduan Pengguna IAM.

Peran IAM dengan kredensial sementara berguna dalam situasi berikut:

- Akses pengguna terfederasi – Untuk menetapkan izin ke identitas terfederasi, Anda membuat peran dan menentukan izin untuk peran tersebut. Ketika identitas terfederasi mengautentikasi, identitas tersebut terhubung dengan peran dan diberi izin yang ditentukan oleh peran. Untuk informasi tentang peran untuk federasi, lihat Buat peran untuk penyedia identitas pihak ketiga dalam Panduan Pengguna IAM. Jika menggunakan Pusat Identitas IAM, Anda harus mengonfigurasi set izin. Untuk mengontrol apa yang dapat diakses identitas Anda setelah identitas tersebut diautentikasi, Pusat Identitas IAM akan mengorelasikan set izin ke peran dalam IAM. Untuk informasi tentang set izin, lihat Set izin dalam Panduan Pengguna AWS IAM Identity Center .
- Izin pengguna IAM sementara – Pengguna atau peran IAM dapat mengambil peran IAM guna mendapatkan berbagai izin secara sementara untuk tugas tertentu.
- Akses lintas akun – Anda dapat menggunakan peran IAM untuk mengizinkan seseorang (prinsipal tepercaya) di akun lain untuk mengakses sumber daya di akun Anda. Peran adalah cara utama untuk memberikan akses lintas akun. Namun, dengan beberapa Layanan AWS, Anda dapat melampirkan kebijakan secara langsung ke sumber daya (alih-alih menggunakan peran sebagai proxy). Untuk mempelajari perbedaan antara peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Akses sumber daya lintas akun di IAM dalam Panduan Pengguna IAM.
- Akses lintas layanan — Beberapa Layanan AWS menggunakan fitur lain Layanan AWS. Misalnya, saat Anda melakukan panggilan dalam suatu layanan, biasanya layanan tersebut menjalankan aplikasi di Amazon EC2 atau menyimpan objek di Amazon S3. Sebuah layanan mungkin

melakukannya menggunakan izin prinsipal yang memanggil, menggunakan peran layanan, atau peran terkait layanan.

- Sesi akses teruskan (FAS) — Saat Anda menggunakan pengguna atau peran IAM untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian menginisiasi tindakan lain di layanan yang berbeda. FAS menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Permintaan FAS hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan ketika mengajukan permintaan FAS, lihat Sesi akses maju.
- Peran layanan – Peran layanan adalah peran IAM yang dijalankan oleh layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat Buat sebuah peran untuk mendelegasikan izin ke Layanan AWS dalam Panduan pengguna IAM.
- Peran terkait layanan — Peran terkait layanan adalah jenis peran layanan yang ditautkan ke peran layanan. Layanan AWS Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.

- Aplikasi yang berjalan di Amazon EC2 — Anda dapat menggunakan peran IAM untuk mengelola kredensi sementara untuk aplikasi yang berjalan pada EC2 instance dan membuat AWS CLI atau AWS permintaan API. Ini lebih baik untuk menyimpan kunci akses dalam EC2 instance. Untuk menetapkan AWS peran ke EC2 instance dan membuatnya tersedia untuk semua aplikasinya, Anda membuat profil instance yang dilampirkan ke instance. Profil instance berisi peran dan memungkinkan program yang berjalan pada EC2 instance untuk mendapatkan kredensi sementara. Untuk informasi selengkapnya, lihat Menggunakan peran IAM untuk memberikan izin ke aplikasi yang berjalan di EC2 instans Amazon di Panduan Pengguna IAM.

Mengelola akses menggunakan kebijakan

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke AWS identitas atau sumber daya. Kebijakan adalah objek AWS yang, ketika dikaitkan dengan identitas atau sumber daya, menentukan izinnya. AWS mengevaluasi kebijakan ini ketika prinsipal (pengguna, pengguna root, atau sesi peran) membuat permintaan. Izin dalam kebijakan menentukan apakah permintaan

diizinkan atau ditolak. Sebagian besar kebijakan disimpan AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang struktur dan isi dokumen kebijakan JSON, lihat Gambaran umum kebijakan JSON dalam Panduan Pengguna IAM.

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, principal dapat melakukan tindakan pada suatu sumber daya, dan dalam suatu syarat.

Secara default, pengguna dan peran tidak memiliki izin. Untuk memberikan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat mengambil peran.

Kebijakan IAM mendefinisikan izin untuk suatu tindakan terlepas dari metode yang Anda gunakan untuk melakukan operasinya. Misalnya, anggaplah Anda memiliki kebijakan yang mengizinkan tindakan `iam:GetRole`. Pengguna dengan kebijakan tersebut bisa mendapatkan informasi peran dari AWS Management Console, API AWS CLI, atau AWS API.

Kebijakan berbasis identitas

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Tentukan izin IAM kustom dengan kebijakan terkelola pelanggan dalam Panduan Pengguna IAM.

Kebijakan berbasis identitas dapat dikategorikan lebih lanjut sebagai kebijakan inline atau kebijakan yang dikelola. Kebijakan inline disematkan langsung ke satu pengguna, grup, atau peran. Kebijakan terkelola adalah kebijakan mandiri yang dapat Anda lampirkan ke beberapa pengguna, grup, dan peran dalam. Akun AWS Kebijakan AWS terkelola mencakup kebijakan terkelola dan kebijakan yang dikelola pelanggan. Untuk mempelajari cara memilih antara kebijakan yang dikelola atau kebijakan inline, lihat Pilih antara kebijakan yang dikelola dan kebijakan inline dalam Panduan Pengguna IAM.

Kebijakan berbasis sumber daya

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya,

administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya tempat kebijakan dilampirkan, kebijakan menentukan tindakan apa yang dapat dilakukan oleh prinsipal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus menentukan prinsipal dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. Layanan AWS

Kebijakan berbasis sumber daya merupakan kebijakan inline yang terletak di layanan tersebut. Anda tidak dapat menggunakan kebijakan AWS terkelola dari IAM dalam kebijakan berbasis sumber daya.

Daftar kontrol akses (ACLs)

Access control lists (ACLs) mengontrol prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACLs mirip dengan kebijakan berbasis sumber daya, meskipun mereka tidak menggunakan format dokumen kebijakan JSON.

Amazon S3, AWS WAF, dan Amazon VPC adalah contoh layanan yang mendukung. ACLs Untuk mempelajari selengkapnya ACLs, lihat Ringkasan daftar kontrol akses (ACL) di Panduan Pengembang Layanan Penyimpanan Sederhana Amazon.

Jenis-jenis kebijakan lain

AWS mendukung jenis kebijakan tambahan yang kurang umum. Jenis-jenis kebijakan ini dapat mengatur izin maksimum yang diberikan kepada Anda oleh jenis kebijakan yang lebih umum.

- Batasan izin – Batasan izin adalah fitur lanjutan tempat Anda mengatur izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas ke entitas IAM (pengguna IAM atau peran IAM). Anda dapat menetapkan batasan izin untuk suatu entitas. Izin yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas milik entitas dan batasan izinnya. Kebijakan berbasis sumber daya yang menentukan pengguna atau peran dalam bidang `Principal` tidak dibatasi oleh batasan izin. Penolakan eksplisit dalam salah satu kebijakan ini akan menggantikan pemberian izin. Untuk informasi selengkapnya tentang batasan izin, lihat Batasan izin untuk entitas IAM dalam Panduan Pengguna IAM.
- Kebijakan kontrol layanan (SCPs) — SCPs adalah kebijakan JSON yang menentukan izin maksimum untuk organisasi atau unit organisasi (OU) di. AWS Organizations AWS Organizations adalah layanan untuk mengelompokkan dan mengelola secara terpusat beberapa Akun AWS yang dimiliki bisnis Anda. Jika Anda mengaktifkan semua fitur dalam organisasi, Anda dapat menerapkan kebijakan kontrol layanan (SCPs) ke salah satu atau semua akun Anda. SCP membatasi izin untuk entitas di akun anggota, termasuk masing-masing. Pengguna root akun AWS

Untuk informasi selengkapnya tentang Organizations dan SCPs, lihat Kebijakan kontrol layanan di Panduan AWS Organizations Pengguna.

- Kebijakan kontrol sumber daya (RCPs) — RCPs adalah kebijakan JSON yang dapat Anda gunakan untuk menetapkan izin maksimum yang tersedia untuk sumber daya di akun Anda tanpa memperbarui kebijakan IAM yang dilampirkan ke setiap sumber daya yang Anda miliki. RCP membatasi izin untuk sumber daya di akun anggota dan dapat memengaruhi izin efektif untuk identitas, termasuk Pengguna root akun AWS, terlepas dari apakah itu milik organisasi Anda. Untuk informasi selengkapnya tentang Organizations dan RCPs, termasuk daftar dukungan Layanan AWS tersebut RCPs, lihat Kebijakan kontrol sumber daya (RCPs) di Panduan AWS Organizations Pengguna.
- Kebijakan sesi – Kebijakan sesi adalah kebijakan lanjutan yang Anda berikan sebagai parameter ketika Anda membuat sesi sementara secara programatis untuk peran atau pengguna terfederasi. Izin sesi yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas pengguna atau peran dan kebijakan sesi. Izin juga bisa datang dari kebijakan berbasis sumber daya. Penolakan eksplisit dalam salah satu kebijakan ini akan menggantikan pemberian izin. Untuk informasi selengkapnya, lihat Kebijakan sesi dalam Panduan Pengguna IAM.

Berbagai jenis kebijakan

Ketika beberapa jenis kebijakan berlaku pada suatu permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan apakah akan mengizinkan permintaan saat beberapa jenis kebijakan terlibat, lihat Logika evaluasi kebijakan di Panduan Pengguna IAM.

Bagaimana Amazon Transcribe bekerja dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses Amazon Transcribe, pelajari fitur IAM yang tersedia untuk digunakan. Amazon Transcribe

IAM fitur yang dapat Anda gunakan dengan Amazon Transcribe

IAM fitur	Amazon Transcribe dukungan
Kebijakan berbasis identitas	Ya
Kebijakan berbasis sumber daya	Tidak
Tindakan kebijakan	Ya

IAM fitur	Amazon Transcribe dukungan
Sumber daya kebijakan	Ya
kunci-kunci persyaratan kebijakan (spesifik layanan)	Ya
ACLs	Tidak
ABAC (tanda dalam kebijakan)	Parsial
Kredensial sementara	Ya
Izin principal	Ya
Peran layanan	Ya
Peran terkait layanan	Tidak

Untuk mendapatkan tampilan tingkat tinggi tentang cara Amazon Transcribe dan AWS layanan lain bekerja dengan sebagian besar IAM fitur, lihat AWS layanan yang berfungsi IAM di Panduan IAM Pengguna.

Kebijakan berbasis identitas untuk Amazon Transcribe

Mendukung kebijakan berbasis identitas: Ya

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Tentukan izin IAM kustom dengan kebijakan terkelola pelanggan dalam Panduan Pengguna IAM.

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan secara spesifik apakah tindakan dan sumber daya diizinkan atau ditolak, serta kondisi yang menjadi dasar dikabulkan atau ditolaknya tindakan tersebut. Anda tidak dapat menentukan secara spesifik prinsipal dalam sebuah kebijakan berbasis identitas karena prinsipal berlaku bagi pengguna atau peran yang melekat kepadanya. Untuk mempelajari semua elemen yang dapat Anda gunakan dalam kebijakan JSON, lihat Referensi elemen kebijakan JSON IAM dalam Panduan Pengguna IAM.

Contoh kebijakan berbasis identitas untuk Amazon Transcribe

Untuk melihat contoh kebijakan Amazon Transcribe berbasis identitas, lihat. Amazon Transcribe contoh kebijakan berbasis identitas

Kebijakan berbasis sumber daya dalam Amazon Transcribe

Mendukung kebijakan berbasis sumber daya: Tidak

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya tempat kebijakan dilampirkan, kebijakan menentukan tindakan apa yang dapat dilakukan oleh prinsipal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus menentukan prinsipal dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. Layanan AWS

Untuk mengaktifkan akses lintas akun, Anda dapat menentukan secara spesifik seluruh akun atau entitas IAM di akun lain sebagai prinsipal dalam kebijakan berbasis sumber daya. Menambahkan prinsipal akun silang ke kebijakan berbasis sumber daya hanya setengah dari membangun hubungan kepercayaan. Ketika prinsipal dan sumber daya berbeda Akun AWS, administrator IAM di akun tepercaya juga harus memberikan izin entitas utama (pengguna atau peran) untuk mengakses sumber daya. Mereka memberikan izin dengan melampirkan kebijakan berbasis identitas kepada entitas. Namun, jika kebijakan berbasis sumber daya memberikan akses ke principal dalam akun yang sama, tidak diperlukan kebijakan berbasis identitas tambahan. Untuk informasi selengkapnya, lihat Akses sumber daya lintas akun di IAM dalam Panduan Pengguna IAM.

Tindakan kebijakan untuk Amazon Transcribe

Mendukung tindakan kebijakan: Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, di mana utama dapat melakukan tindakan pada sumber daya, dan dalam kondisi apa.

Elemen `Action` dari kebijakan JSON menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Tindakan kebijakan biasanya memiliki

nama yang sama dengan operasi AWS API terkait. Ada beberapa pengecualian, misalnya tindakan hanya izin yang tidak memiliki operasi API yang cocok. Ada juga beberapa operasi yang memerlukan beberapa tindakan dalam suatu kebijakan. Tindakan tambahan ini disebut tindakan dependen.

Sertakan tindakan dalam kebijakan untuk memberikan izin untuk melakukan operasi terkait.

Untuk melihat daftar Amazon Transcribe tindakan, lihat Tindakan yang ditentukan oleh Amazon Transcribe dalam Referensi Otorisasi Layanan.

Tindakan kebijakan Amazon Transcribe menggunakan `transcribe` awalan sebelum tindakan. Untuk menetapkan secara spesifik beberapa tindakan dalam satu pernyataan, pisahkan tindakan tersebut dengan koma.

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

Anda juga dapat menentukan beberapa tindakan menggunakan wildcard (*). Sebagai contoh, untuk menentukan semua tindakan yang dimulai dengan kata `List`, sertakan tindakan berikut:

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

Untuk melihat contoh kebijakan Amazon Transcribe berbasis identitas, lihat. Amazon Transcribe contoh kebijakan berbasis identitas

Sumber daya kebijakan untuk Amazon Transcribe

Mendukung sumber daya kebijakan: Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, di mana utama dapat melakukan tindakan pada sumber daya, dan dalam kondisi apa.

Elemen kebijakan JSON `Resource` menentukan objek yang menjadi target penerapan tindakan. Pernyataan harus menyertakan elemen `Resource` atau `NotResource`. Praktik terbaiknya, tentukan

sumber daya menggunakan Amazon Resource Name (ARN). Anda dapat melakukan ini untuk tindakan yang mendukung jenis sumber daya tertentu, yang dikenal sebagai izin tingkat sumber daya.

Untuk tindakan yang tidak mendukung izin di tingkat sumber daya, misalnya operasi pencantuman, gunakan wildcard (*) untuk menunjukkan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

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

Untuk melihat daftar jenis sumber daya dan jenis Amazon Transcribe sumber daya ARNs, lihat Sumber daya yang ditentukan oleh Amazon Transcribe dalam Referensi Otorisasi Layanan. Untuk mempelajari tindakan yang dapat menentukan ARN setiap sumber daya, lihat Tindakan yang ditentukan oleh Amazon Transcribe.

Untuk melihat contoh kebijakan Amazon Transcribe berbasis identitas, lihat. Amazon Transcribe contoh kebijakan berbasis identitas

Kunci kondisi kebijakan untuk Amazon Transcribe

Mendukung kunci kondisi kebijakan khusus layanan: Yes

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, di mana utama dapat melakukan tindakan pada sumber daya, dan dalam kondisi apa.

Elemen `Condition` (atau blok `Condition`) akan memungkinkan Anda menentukan kondisi yang menjadi dasar suatu pernyataan berlaku. Elemen `Condition` bersifat opsional. Anda dapat membuat ekspresi bersyarat yang menggunakan operator kondisi, misalnya sama dengan atau kurang dari, untuk mencocokkan kondisi dalam kebijakan dengan nilai-nilai yang diminta.

Jika Anda menentukan beberapa elemen `Condition` dalam sebuah pernyataan, atau beberapa kunci dalam elemen `Condition` tunggal, maka AWS akan mengevaluasinya menggunakan operasi `AND` logis. Jika Anda menentukan beberapa nilai untuk satu kunci kondisi, AWS mengevaluasi kondisi menggunakan `OR` operasi logis. Semua kondisi harus dipenuhi sebelum izin pernyataan diberikan.

Anda juga dapat menggunakan variabel placeholder saat menentukan kondisi. Sebagai contoh, Anda dapat memberikan izin kepada pengguna IAM untuk mengakses sumber daya hanya jika

izin tersebut mempunyai tanda yang sesuai dengan nama pengguna IAM mereka. Untuk informasi selengkapnya, lihat Elemen kebijakan IAM: variabel dan tanda dalam Panduan Pengguna IAM.

AWS mendukung kunci kondisi global dan kunci kondisi khusus layanan. Untuk melihat semua kunci kondisi AWS global, lihat kunci konteks kondisi AWS global di Panduan Pengguna IAM.

Untuk melihat daftar kunci Amazon Transcribe kondisi, lihat Kunci kondisi untuk Amazon Transcribe dalam Referensi Otorisasi Layanan. Untuk mempelajari tindakan dan sumber daya yang dapat Anda gunakan kunci kondisi, lihat Tindakan yang ditentukan oleh Amazon Transcribe.

Untuk melihat contoh kebijakan Amazon Transcribe berbasis identitas, lihat. Amazon Transcribe contoh kebijakan berbasis identitas

ACLs di Amazon Transcribe

Mendukung ACLs: Tidak

Access control lists (ACLs) mengontrol prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACLs mirip dengan kebijakan berbasis sumber daya, meskipun mereka tidak menggunakan format dokumen kebijakan JSON.

ABAC dengan Amazon Transcribe

Mendukung ABAC (tag dalam kebijakan): Sebagian

Kontrol akses berbasis atribut (ABAC) adalah strategi otorisasi yang menentukan izin berdasarkan atribut. Dalam AWS, atribut ini disebut tag. Anda dapat melampirkan tag ke entitas IAM (pengguna atau peran) dan ke banyak AWS sumber daya. Penandaan ke entitas dan sumber daya adalah langkah pertama dari ABAC. Kemudian rancanglah kebijakan ABAC untuk mengizinkan operasi ketika tanda milik prinsipal cocok dengan tanda yang ada di sumber daya yang ingin diakses.

ABAC sangat berguna di lingkungan yang berkembang dengan cepat dan berguna di situasi saat manajemen kebijakan menjadi rumit.

Untuk mengendalikan akses berdasarkan tanda, berikan informasi tentang tanda di elemen kondisi dari kebijakan menggunakan kunci kondisi `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`*, atau `aws:TagKeys`.

Jika sebuah layanan mendukung ketiga kunci kondisi untuk setiap jenis sumber daya, nilainya adalah Ya untuk layanan tersebut. Jika suatu layanan mendukung ketiga kunci kondisi untuk hanya beberapa jenis sumber daya, nilainya adalah Parsial.

Untuk informasi selengkapnya tentang ABAC, lihat Tentukan izin dengan otorisasi ABAC dalam Panduan Pengguna IAM. Untuk melihat tutorial yang menguraikan langkah-langkah pengaturan ABAC, lihat Menggunakan kontrol akses berbasis atribut (ABAC) dalam Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang menandai Amazon Transcribe sumber daya, lihatPenandaan pada sumber daya . Untuk informasi lebih rinci tentang kontrol akses berbasis tag, lihat Mengontrol akses ke AWS sumber daya menggunakan tag.

Menggunakan kredensial sementara dengan Amazon Transcribe

Mendukung kredensial sementara: Ya

Beberapa Layanan AWS tidak berfungsi saat Anda masuk menggunakan kredensi sementara. Untuk informasi tambahan, termasuk yang Layanan AWS bekerja dengan kredensi sementara, lihat Layanan AWS yang bekerja dengan IAM di Panduan Pengguna IAM.

Anda menggunakan kredensi sementara jika Anda masuk AWS Management Console menggunakan metode apa pun kecuali nama pengguna dan kata sandi. Misalnya, ketika Anda mengakses AWS menggunakan tautan masuk tunggal (SSO) perusahaan Anda, proses tersebut secara otomatis membuat kredensil sementara. Anda juga akan secara otomatis membuat kredensial sementara ketika Anda masuk ke konsol sebagai seorang pengguna lalu beralih peran. Untuk informasi selengkapnya tentang peralihan peran, lihat Beralih dari pengguna ke peran IAM (konsol) dalam Panduan Pengguna IAM.

Anda dapat membuat kredenal sementara secara manual menggunakan API AWS CLI atau AWS . Anda kemudian dapat menggunakan kredensi sementara tersebut untuk mengakses. AWS AWS merekomendasikan agar Anda secara dinamis menghasilkan kredensi sementara alih-alih menggunakan kunci akses jangka panjang. Untuk informasi lebih lanjut, lihat Kredensial keamanan sementara di IAM.

Izin principal lintas layanan untuk Amazon Transcribe

Mendukung sesi akses maju (FAS): Ya

Saat Anda menggunakan pengguna atau peran IAM untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian menginisiasi tindakan lain di layanan yang berbeda. FAS menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Permintaan FAS hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk

menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan ketika mengajukan permintaan FAS, lihat Sesi akses maju.

Peran layanan untuk Amazon Transcribe

Mendukung peran layanan: Ya

Peran layanan adalah peran IAM yang diambil oleh sebuah layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat Buat sebuah peran untuk mendelegasikan izin ke Layanan AWS dalam Panduan pengguna IAM.

> **Warning**
>
> Mengubah izin untuk peran layanan dapat merusak Amazon Transcribe fungsionalitas. Edit peran layanan hanya jika Amazon Transcribe memberikan panduan untuk melakukannya.

Peran terkait untuk Amazon Transcribe

Mendukung peran terkait layanan: Tidak

Peran terkait layanan adalah jenis peran layanan yang ditautkan ke. Layanan AWS Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.

Amazon Transcribe tidak mendukung peran terkait layanan.

Untuk detail tentang membuat atau mengelola peran terkait layanan untuk layanan lain, lihat AWS layanan yang berfungsi dengannya. IAM Cari layanan dalam tabel yang memiliki `Yes` di kolom Peran terkait layanan. Pilih tautan Ya untuk melihat dokumentasi peran terkait layanan untuk layanan tersebut.

Pencegahan "confused deputy" lintas layanan

Deputi yang bingung adalah entitas (layanan atau akun) yang dipaksa oleh entitas yang berbeda untuk melakukan suatu tindakan. Jenis peniruan identitas ini dapat terjadi lintas akun dan lintas layanan.

Untuk mencegah kebingungan deputi, AWS sediakan alat yang membantu Anda melindungi data Anda untuk semua layanan menggunakan prinsip layanan yang telah diberikan akses ke sumber daya di Anda. Akun AWS Bagian ini berfokus pada pencegahan wakil kebingungan lintas layanan khusus untuk Amazon Transcribe; namun, Anda dapat mempelajari lebih lanjut tentang topik ini di bagian masalah wakil yang membingungkan dari Panduan IAM Pengguna.

Untuk membatasi izin yang IAM diberikan Amazon Transcribe untuk mengakses sumber daya Anda, sebaiknya gunakan kunci konteks kondisi global `aws:SourceArn`dan `aws:SourceAccount`dalam kebijakan sumber daya Anda.

Jika Anda menggunakan kedua kunci konteks kondisi global ini, dan `aws:SourceArn` nilainya berisi Akun AWS ID, `aws:SourceAccount` nilai dan Akun AWS in `aws:SourceArn` harus menggunakan Akun AWS ID yang sama saat digunakan dalam pernyataan kebijakan yang sama.

Jika Anda ingin hanya satu sumber daya yang akan dikaitkan dengan akses lintas layanan, gunakan `aws:SourceArn`. Jika Anda ingin mengaitkan sumber daya apa pun Akun AWS dengan akses lintas layanan, gunakan`aws:SourceAccount`.

> **Note**
>
> Cara paling efektif untuk melindungi dari masalah wakil yang membingungkan adalah dengan menggunakan kunci konteks kondisi `aws:SourceArn` global dengan ARN penuh sumber daya. Jika Anda tidak mengetahui ARN lengkap, atau jika Anda menentukan beberapa sumber daya, gunakan kunci kondisi konteks `aws:SourceArn` global dengan wildcard (*) untuk bagian ARN yang tidak diketahui. Misalnya, `arn:aws:transcribe::123456789012:*`.

Untuk contoh kebijakan peran asumsi yang menunjukkan bagaimana Anda dapat mencegah masalah wakil yang membingungkan, lihatKebijakan pencegahan wakil yang membingungkan.

Amazon Transcribe contoh kebijakan berbasis identitas

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau mengubah sumber daya Amazon Transcribe . Mereka juga tidak dapat melakukan tugas dengan menggunakan AWS Management Console, AWS Command Line Interface (AWS CLI), atau AWS API. Untuk memberikan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat mengambil peran.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM menggunakan contoh dokumen kebijakan JSON ini, lihat Membuat kebijakan IAM (konsol) di Panduan Pengguna IAM.

Untuk detail tentang tindakan dan jenis sumber daya yang ditentukan oleh Amazon Transcribe, termasuk format ARNs untuk setiap jenis sumber daya, lihat Kunci tindakan, sumber daya, dan kondisi untuk Amazon Transcribe di Referensi Otorisasi Layanan.

Topik

- Praktik terbaik kebijakan
- Menggunakan AWS Management Console
- Izin diperlukan untuk peran IAM
- Izin diperlukan untuk kunci Amazon S3 enkripsi
- Mengizinkan pengguna melihat izin mereka sendiri
- AWS KMS kebijakan konteks enkripsi
- Kebijakan pencegahan wakil yang membingungkan
- Melihat pekerjaan transkripsi berdasarkan tag

Praktik terbaik kebijakan

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus Amazon Transcribe sumber daya di akun Anda. Tindakan ini membuat Akun AWS Anda dikenai biaya. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan AWS terkelola dan beralih ke izin hak istimewa paling sedikit — Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan AWS terkelola yang memberikan izin untuk banyak kasus penggunaan umum. Mereka tersedia di Anda Akun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola AWS pelanggan yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat Kebijakan yang dikelola AWS atau Kebijakan yang dikelola AWS untuk fungsi tugas dalam Panduan Pengguna IAM.
- Menerapkan izin dengan hak akses paling rendah – Ketika Anda menetapkan izin dengan kebijakan IAM, hanya berikan izin yang diperlukan untuk melakukan tugas. Anda melakukannya dengan mendefinisikan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, yang juga dikenal sebagai izin dengan hak akses paling rendah. Untuk informasi

selengkapnya tentang cara menggunakan IAM untuk mengajukan izin, lihat Kebijakan dan izin dalam IAM dalam Panduan Pengguna IAM.

- Gunakan kondisi dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan suatu kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Sebagai contoh, Anda dapat menulis kondisi kebijakan untuk menentukan bahwa semua permintaan harus dikirim menggunakan SSL. Anda juga dapat menggunakan ketentuan untuk memberikan akses ke tindakan layanan jika digunakan melalui yang spesifik Layanan AWS, seperti AWS CloudFormation. Untuk informasi selengkapnya, lihat Elemen kebijakan JSON IAM: Kondisi dalam Panduan Pengguna IAM.
- Gunakan IAM Access Analyzer untuk memvalidasi kebijakan IAM Anda untuk memastikan izin yang aman dan fungsional – IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan tersebut mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM. IAM Access Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat Validasi kebijakan dengan IAM Access Analyzer dalam Panduan Pengguna IAM.
- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Anda, Akun AWS aktifkan MFA untuk keamanan tambahan. Untuk meminta MFA ketika operasi API dipanggil, tambahkan kondisi MFA pada kebijakan Anda. Untuk informasi selengkapnya, lihat Amankan akses API dengan MFA dalam Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, lihat Praktik terbaik keamanan di IAM dalam Panduan Pengguna IAM.

Menggunakan AWS Management Console

Untuk mengakses konsol Amazon Transcribe, Anda harus memiliki set izin minimum. Izin ini harus memungkinkan Anda untuk membuat daftar dan melihat detail tentang Amazon Transcribe sumber daya di Anda Akun AWS. Jika Anda membuat kebijakan berbasis identitas yang lebih ketat daripada izin minimum yang diperlukan, konsol tidak akan berfungsi sebagaimana mestinya untuk entitas (pengguna atau peran) dengan kebijakan tersebut.

Anda tidak perlu mengizinkan izin konsol minimum untuk pengguna yang melakukan panggilan hanya ke AWS CLI atau AWS API. Sebagai gantinya, izinkan akses hanya ke tindakan yang sesuai dengan operasi API yang coba mereka lakukan.

Untuk memastikan bahwa entitas (pengguna dan peran) dapat menggunakan AWS Management Console, lampirkan salah satu kebijakan AWS-managed berikut kepada mereka.

- `AmazonTranscribeFullAccess`: Memberikan akses penuh untuk membuat, membaca, memperbarui, menghapus, dan menjalankan semua Amazon Transcribe sumber daya. Ini juga memungkinkan akses ke Amazon S3 ember `transcribe` dengan nama bucket.
- `AmazonTranscribeReadOnlyAccess`: Memberikan akses hanya-baca ke Amazon Transcribe sumber daya sehingga Anda bisa mendapatkan dan membuat daftar pekerjaan transkripsi dan kosakata khusus.

> **Note**
>
> Anda dapat meninjau kebijakan izin terkelola dengan masuk ke IAM AWS Management Console dan mencari berdasarkan nama kebijakan. Pencarian untuk “transkripsi” mengembalikan kedua kebijakan yang tercantum di atas (AmazonTranscribeReadOnlydan AmazonTranscribeFullAccess).

Anda juga dapat membuat IAM kebijakan kustom sendiri untuk mengizinkan izin tindakan Amazon Transcribe API. Anda dapat melampirkan kebijakan khusus ini ke entitas yang memerlukan izin tersebut.

Izin diperlukan untuk peran IAM

Jika Anda membuat IAM peran untuk dipanggil Amazon Transcribe, peran tersebut harus memiliki izin untuk mengakses Amazon S3 bucket. Jika berlaku, KMS key harus juga digunakan untuk mengenkripsi isi bucket. Lihat bagian berikut misalnya kebijakan.

Kebijakan kepercayaan

IAM Entitas yang Anda gunakan untuk membuat permintaan transkripsi Anda harus memiliki kebijakan kepercayaan yang memungkinkan Amazon Transcribe untuk mengambil peran itu. Gunakan kebijakan Amazon Transcribe kepercayaan berikut. Perhatikan bahwa jika Anda membuat permintaan Analytics Panggilan real-time dengan analitik pasca-panggilan diaktifkan, Anda harus menggunakan 'Kebijakan kepercayaan untuk Analitik Panggilan waktu'.

Kebijakan kepercayaan untuk 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:*"
        }
      }
    }
  ]
}
```

Kebijakan kepercayaan untuk Analisis Panggilan real-time

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "transcribe.streaming.amazonaws.com"
        ]
      },
      "Action": [
        "sts:AssumeRole"
      ],
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "111122223333"
        },
```

```
        "StringLike": {
          "aws:SourceArn": "arn:aws:transcribe:us-west-2:111122223333:*"
        }
      }
    }
  ]
}
```

Amazon S3 kebijakan bucket masukan

Kebijakan berikut memberikan izin IAM peran untuk mengakses file dari bucket input yang ditentukan.

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

Amazon S3 kebijakan bucket keluaran

Kebijakan berikut memberikan izin IAM peran untuk menulis file ke bucket keluaran yang ditentukan.

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

```
}
```

Izin diperlukan untuk kunci Amazon S3 enkripsi

Jika Anda menggunakan KMS key untuk mengenkripsi Amazon S3 bucket, sertakan yang berikut ini dalam KMS key kebijakan. Ini memberikan Amazon Transcribe akses ke isi ember. Untuk informasi selengkapnya tentang mengizinkan akses KMS keys, lihat Mengizinkan eksternal Akun AWS mengakses KMS key di Panduan AWS KMS Pengembang.

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

Mengizinkan pengguna melihat izin mereka sendiri

Contoh ini menunjukkan cara membuat kebijakan yang mengizinkan pengguna IAM melihat kebijakan inline dan terkelola yang dilampirkan ke identitas pengguna mereka. Kebijakan ini mencakup izin untuk menyelesaikan tindakan ini di konsol atau menggunakan API atau secara terprogram. AWS CLI AWS

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
```

```
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS KMS kebijakan konteks enkripsi

Kebijakan berikut memberikan izin “`ExampleRole`” IAM peran untuk menggunakan operasi AWS KMS Dekripsi dan Enkripsi untuk hal ini. KMS key Kebijakan ini hanya berfungsi untuk permintaan dengan setidaknya satu pasangan konteks enkripsi, dalam hal ini "`color:indigoBlue`”. Untuk informasi selengkapnya tentang konteks AWS KMS enkripsi, lihat[AWS KMS konteks enkripsi](#).

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

Kebijakan pencegahan wakil yang membingungkan

Berikut adalah contoh kebijakan peran asumsi yang menunjukkan bagaimana Anda dapat menggunakan `aws:SourceArn` dan `aws:SourceAccount` dengan Amazon Transcribe untuk mencegah masalah wakil yang membingungkan. Untuk informasi lebih lanjut tentang pencegahan wakil yang membingungkan, lihatPencegahan "confused deputy" lintas layanan.

```
{
  "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:*"
        }
      }
    }
  ]
}
```

Melihat pekerjaan transkripsi berdasarkan tag

Anda dapat menggunakan syarat dalam kebijakan berbasis identitas Anda untuk mengontrol akses ke sumber daya Amazon Transcribe berdasarkan tanda. Contoh ini menunjukkan cara Anda membuat kebijakan yang memungkinkan melihat pekerjaan transkripsi. Namun, izin hanya diberikan jika tag pekerjaan transkripsi `Owner` memiliki nilai nama pengguna tersebut. Kebijakan ini juga memberikan izin yang diperlukan untuk menyelesaikan tindakan ini menggunakan. AWS Management Console

Anda dapat melampirkan kebijakan ini ke IAM entitas di akun Anda. Jika peran bernama `test-role` mencoba melihat pekerjaan transkripsi, pekerjaan transkripsi harus diberi tag `Owner=test-role` atau `owner=test-role` (nama kunci kondisi tidak peka huruf besar/kecil), jika tidak maka akan ditolak aksesnya. Untuk informasi selengkapnya, lihat elemen kebijakan IAM JSON: Kondisi dalam Panduan IAM Pengguna.

Untuk informasi lebih lanjut tentang penandaan Amazon Transcribe, lihatPenandaan pada sumber daya .

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

Memecahkan masalah Amazon Transcribe identitas dan akses

Gunakan informasi berikut untuk mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui saat bekerja dengan Amazon Transcribe dan AWS Identity and Access Management (IAM).

Topik

- Saya tidak diotorisasi untuk melakukan tindakan di Amazon Transcribe
- Saya tidak berwenang untuk melakukan iam: PassRole
- Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses Amazon Transcribe sumber daya saya

Saya tidak diotorisasi untuk melakukan tindakan di Amazon Transcribe

Jika Anda menerima pesan kesalahan bahwa Anda tidak memiliki otorisasi untuk melakukan tindakan, kebijakan Anda harus diperbarui agar Anda dapat melakukan tindakan tersebut.

Contoh kesalahan berikut terjadi ketika pengguna IAM `mateojackson` mencoba menggunakan konsol untuk melihat detail tentang suatu sumber daya *`my-example-widget`* rekaan, tetapi tidak memiliki izin `transcribe:`*`GetWidget`* rekaan.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 transcribe:GetWidget on resource: my-example-widget
```

Dalam hal ini, kebijakan untuk pengguna `mateojackson` harus diperbarui untuk mengizinkan akses ke sumber daya *`my-example-widget`* dengan menggunakan tindakan `transcribe:`*`GetWidget`*.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya tidak berwenang untuk melakukan iam: PassRole

Jika Anda menerima kesalahan yang tidak diizinkan untuk melakukan `iam:PassRole` tindakan, kebijakan Anda harus diperbarui agar Anda dapat meneruskan peran Amazon Transcribe.

Beberapa Layanan AWS memungkinkan Anda untuk meneruskan peran yang ada ke layanan tersebut alih-alih membuat peran layanan baru atau peran terkait layanan. Untuk melakukannya, Anda harus memiliki izin untuk meneruskan peran ke layanan.

Contoh kesalahan berikut terjadi ketika pengguna IAM bernama `marymajor` mencoba menggunakan konsol tersebut untuk melakukan tindakan di Amazon Transcribe. Namun, tindakan tersebut memerlukan layanan untuk mendapatkan izin yang diberikan oleh peran layanan. Mary tidak memiliki izin untuk meneruskan peran tersebut pada layanan.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Dalam kasus ini, kebijakan Mary harus diperbarui agar dia mendapatkan izin untuk melakukan tindakan `iam:PassRole` tersebut.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses Amazon Transcribe sumber daya saya

Anda dapat membuat peran yang dapat digunakan pengguna di akun lain atau orang-orang di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa saja yang dipercaya untuk mengambil peran tersebut. Untuk layanan yang mendukung kebijakan berbasis sumber daya atau daftar kontrol akses (ACLs), Anda dapat menggunakan kebijakan tersebut untuk memberi orang akses ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa referensi berikut:

- Untuk mempelajari apakah Amazon Transcribe mendukung fitur-fitur ini, lihatBagaimana Amazon Transcribe bekerja dengan IAM.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda di seluruh sumber daya Akun AWS yang Anda miliki, lihat Menyediakan akses ke pengguna IAM di pengguna lain Akun AWS yang Anda miliki di Panduan Pengguna IAM.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda kepada pihak ketiga Akun AWS, lihat Menyediakan akses yang Akun AWS dimiliki oleh pihak ketiga dalam Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses melalui federasi identitas, lihat Menyediakan akses ke pengguna terautentikasi eksternal (federasi identitas) dalam Panduan Pengguna IAM.
- Untuk mempelajari perbedaan antara menggunakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Akses sumber daya lintas akun di IAM di Panduan Pengguna IAM.

Perlindungan data di Amazon Transcribe

Model tanggung jawab AWS bersama model berlaku untuk perlindungan data di Amazon Transcribe. Seperti yang dijelaskan dalam model AWS ini, bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk mempertahankan kendali atas konten yang di-host pada infrastruktur ini. Anda juga bertanggung jawab atas tugas-tugas konfigurasi dan manajemen keamanan untuk Layanan AWS yang Anda gunakan. Lihat informasi yang lebih lengkap tentang privasi data dalam Pertanyaan Umum Privasi Data. Lihat informasi tentang perlindungan data di Eropa di pos blog Model Tanggung Jawab Bersama dan GDPR AWS di Blog Keamanan AWS .

Untuk tujuan perlindungan data, kami menyarankan Anda melindungi Akun AWS kredensyal dan mengatur pengguna individu dengan AWS IAM Identity Center atau AWS Identity and Access Management (IAM). Dengan cara itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugasnya. Kami juga menyarankan supaya Anda mengamankan data dengan cara-cara berikut:

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk berkomunikasi dengan sumber daya. AWS Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Siapkan API dan pencatatan aktivitas pengguna dengan AWS CloudTrail. Untuk informasi tentang penggunaan CloudTrail jejak untuk menangkap AWS aktivitas, lihat Bekerja dengan CloudTrail jejak di AWS CloudTrail Panduan Pengguna.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default di dalamnya Layanan AWS.
- Gunakan layanan keamanan terkelola tingkat lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-3 saat mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Lihat informasi selengkapnya tentang titik akhir FIPS yang tersedia di Standar Pemrosesan Informasi Federal (FIPS) 140-3.

Kami sangat merekomendasikan agar Anda tidak pernah memasukkan informasi identifikasi yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam tanda atau bidang isian bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan Amazon Transcribe atau lainnya Layanan AWS menggunakan konsol, API AWS CLI, atau AWS SDKs. Data apa pun yang Anda masukkan ke dalam tanda atau bidang isian bebas yang digunakan untuk nama dapat digunakan untuk log

penagihan atau log diagnostik. Saat Anda memberikan URL ke server eksternal, kami sangat menganjurkan supaya Anda tidak menyertakan informasi kredensial di dalam URL untuk memvalidasi permintaan Anda ke server itu.

Privasi lalu lintas antar jaringan

Endpoint Amazon Virtual Private Cloud (Amazon VPC) for Amazon Transcribe adalah entitas logis dalam VPC yang memungkinkan konektivitas hanya untuk. Amazon Transcribe Amazon VPC merutekan permintaan ke Amazon Transcribe dan merutekan respons kembali ke VPC. Untuk informasi lebih lanjut, lihat AWS PrivateLink konsep. Untuk informasi tentang menggunakan Amazon VPC titik akhir dengan Amazon Transcribe lihatAmazon Transcribe dan antarmuka titik akhir VPC ()AWS PrivateLink.

Enkripsi data

Enkripsi data mengacu pada melindungi data saat dalam perjalanan dan saat istirahat. Anda dapat melindungi data Anda dengan menggunakan kunci Amazon S3-managed atau KMS keys diam, bersama Transport Layer Security (TLS) standar saat dalam perjalanan.

Enkripsi diam

Amazon Transcribe menggunakan Amazon S3 kunci default (SSE-S3) untuk enkripsi transkrip sisi server yang ditempatkan di bucket Anda. Amazon S3

Saat Anda menggunakan `StartTranscriptionJob`operasi, Anda dapat menentukan sendiri KMS key untuk mengenkripsi output dari pekerjaan transkripsi.

Amazon Transcribe menggunakan Amazon EBS volume yang dienkripsi dengan kunci default.

Enkripsi bergerak

Amazon Transcribe menggunakan TLS 1.2 dengan AWS sertifikat untuk mengenkripsi data dalam perjalanan. Ini termasuk transkripsi streaming.

Manajemen kunci

Amazon Transcribe bekerja dengan KMS keys untuk menyediakan enkripsi yang disempurnakan untuk data Anda. Dengan Amazon S3, Anda dapat mengenkripsi media input Anda saat membuat pekerjaan transkripsi. Integrasi dengan AWS KMS memungkinkan enkripsi output dari `StartTranscriptionJob`permintaan.

Jika Anda tidak menentukan a KMS key, output dari pekerjaan transkripsi dienkripsi dengan Amazon S3 kunci default (SSE-S3).

Untuk informasi selengkapnya AWS KMS, lihat Panduan AWS Key Management Service Pengembang.

Manajemen kunci menggunakan AWS Management Console

Untuk mengenkripsi output pekerjaan transkripsi Anda, Anda dapat memilih antara menggunakan KMS key untuk Akun AWS yang membuat permintaan, atau KMS key dari yang lain. Akun AWS

Jika Anda tidak menentukan a KMS key, output dari pekerjaan transkripsi dienkripsi dengan Amazon S3 kunci default (SSE-S3).

Untuk mengaktifkan enkripsi output:

1. Di bawah Data keluaran pilih Enkripsi.

2. Pilih apakah KMS key itu dari yang Akun AWS Anda gunakan saat ini atau dari yang berbeda Akun AWS. Jika Anda ingin menggunakan kunci dari saat ini Akun AWS, pilih kunci dari KMS key ID. Jika Anda menggunakan kunci dari yang berbeda Akun AWS, Anda harus memasukkan ARN kunci. Untuk menggunakan kunci dari yang berbeda Akun AWS, pemanggil harus memiliki `kms:Encrypt` izin untuk. KMS key Lihat Membuat kebijakan utama untuk informasi lebih lanjut.

Manajemen kunci menggunakan API

Untuk menggunakan enkripsi keluaran dengan API, Anda harus menentukan KMS key penggunaan `OutputEncryptionKMSKeyId` parameter `StartCallAnalyticsJob`, `StartMedicalTranscriptionJob`, atau `StartTranscriptionJob`operasi.

Jika menggunakan kunci yang terletak di saat ini Akun AWS, Anda dapat menentukan Anda KMS key dalam salah satu dari empat cara:

1. Gunakan KMS key ID itu sendiri. Misalnya, `1234abcd-12ab-34cd-56ef-1234567890ab`.
2. Gunakan alias untuk KMS key ID. Misalnya, `alias/ExampleAlias`.
3. Gunakan Nama Sumber Daya Amazon (ARN) untuk ID. KMS key Misalnya, `arn:aws:kms:region:account-ID:key/1234abcd-12ab-34cd-56ef-1234567890ab`.
4. Gunakan ARN untuk alias. KMS key Misalnya, `arn:aws:kms:region:account-ID:alias/ExampleAlias`.

Jika menggunakan kunci yang terletak di berbeda Akun AWS dari saat ini Akun AWS, Anda dapat menentukan Anda KMS key dalam salah satu dari dua cara:

1. Gunakan ARN untuk ID. KMS key Misalnya, `arn:aws:kms:region:account-ID:key/1234abcd-12ab-34cd-56ef-1234567890ab`.
2. Gunakan ARN untuk alias. KMS key Misalnya, `arn:aws:kms:region:account-ID:alias/ExampleAlias`.

Perhatikan bahwa entitas yang membuat permintaan harus memiliki izin untuk menggunakan yang ditentukan KMS key.

AWS KMS konteks enkripsi

AWS KMS konteks enkripsi adalah peta teks biasa, pasangan kunci non-rahasia: nilai. Peta ini mewakili data otentikasi tambahan, yang dikenal sebagai pasangan konteks enkripsi, yang menyediakan lapisan keamanan tambahan untuk data Anda. Amazon Transcribe memerlukan kunci enkripsi simetris untuk mengenkripsi output transkripsi ke dalam bucket yang ditentukan pelanggan. Amazon S3 Untuk mempelajari lebih lanjut, lihat Kunci asimetris di AWS KMS.

Saat membuat pasangan konteks enkripsi Anda, jangan sertakan informasi sensitif. Konteks enkripsi bukan rahasia — itu terlihat dalam teks biasa dalam CloudTrail log Anda (sehingga Anda dapat menggunakannya untuk mengidentifikasi dan mengkategorikan operasi kriptografi Anda).

Pasangan konteks enkripsi Anda dapat menyertakan karakter khusus, seperti garis bawah (_), tanda hubung (), garis miring (,-) dan / titik dua (\). :

> **Tip**
>
> Ini dapat berguna untuk menghubungkan nilai-nilai dalam pasangan konteks enkripsi Anda dengan data yang dienkripsi. Meskipun tidak diperlukan, kami sarankan Anda menggunakan

metadata non-sensitif yang terkait dengan konten terenkripsi Anda, seperti nama file, nilai header, atau bidang database yang tidak terenkripsi.

Untuk menggunakan enkripsi output dengan API, atur `KMSEncryptionContext` parameter dalam `StartTranscriptionJob`operasi. Untuk menyediakan konteks enkripsi untuk operasi enkripsi output, `OutputEncryptionKMSKeyId` parameter harus mereferensikan KMS key ID simetris.

Anda dapat menggunakan kunci AWS KMS kondisi dengan IAM kebijakan untuk mengontrol akses ke enkripsi simetris KMS key berdasarkan konteks enkripsi yang digunakan dalam permintaan operasi kriptografi. Untuk contoh kebijakan konteks enkripsi, lihatAWS KMS kebijakan konteks enkripsi.

Menggunakan konteks enkripsi adalah opsional, tetapi disarankan. Untuk informasi selengkapnya, lihat Konteks enkripsi.

Memilih untuk tidak menggunakan data Anda untuk perbaikan layanan

Secara default, Amazon Transcribe menyimpan dan menggunakan input suara yang telah diproses untuk mengembangkan layanan dan terus meningkatkan pengalaman Anda. Anda dapat memilih untuk tidak menggunakan konten Anda untuk mengembangkan dan meningkatkan Amazon Transcribe dengan menggunakan kebijakan AWS Organizations opt-out. Untuk informasi tentang cara memilih keluar, lihat kebijakan opt-out layanan AI.

Pemantauan Amazon Transcribe

Pemantauan adalah bagian penting dari menjaga keandalan, ketersediaan, dan kinerja Amazon Transcribe dan AWS solusi Anda yang lain. AWS menyediakan alat pemantauan berikut untuk menonton Amazon Transcribe, melaporkan ketika ada sesuatu yang salah, dan mengambil tindakan otomatis bila perlu:

- Amazon CloudWatchmemantau AWS sumber daya Anda dan aplikasi yang Anda jalankan AWS secara real time. Anda dapat mengumpulkan dan melacak metrik, membuat dasbor yang disesuaikan, dan mengatur alarm yang memberi tahu Anda atau mengambil tindakan saat metrik tertentu mencapai ambang batas yang ditentukan. Misalnya, Anda dapat CloudWatch melacak penggunaan CPU atau metrik lain pada Amazon EC2 instans Anda dan secara otomatis meluncurkan instance baru bila diperlukan.

- Amazon CloudWatch Logsdapat memantau, menyimpan, dan mengakses file log Anda dari Amazon EC2 instance, CloudTrail, dan sumber lainnya. CloudWatch Logs dapat memantau informasi dalam file log dan memberi tahu Anda ketika ambang batas tertentu terpenuhi. Anda juga dapat mengarsipkan data log dalam penyimpanan yang sangat durabel.
- AWS CloudTrailmenangkap panggilan API dan peristiwa terkait yang dibuat oleh atau atas nama Anda Akun AWS dan mengirimkan file log ke Amazon S3 bucket yang Anda tentukan. Anda dapat mengidentifikasi pengguna dan akun mana yang dipanggil AWS, alamat IP sumber dari mana panggilan dilakukan, dan kapan panggilan terjadi.

Untuk informasi selengkapnya, silakan lihat Panduan Pengguna Amazon CloudWatch.

Amazon EventBridgeadalah layanan tanpa server yang menggunakan peristiwa untuk menghubungkan komponen aplikasi bersama-sama, sehingga memudahkan Anda untuk membangun aplikasi berbasis peristiwa yang dapat diskalakan. EventBridge memberikan aliran data real-time dari aplikasi Anda sendiri, aplikasi Perangkat Lunak sebagai Layanan (SaaS), AWS dan layanan serta rute data tersebut ke target seperti. Lambda Anda dapat memantau peristiwa yang terjadi di layanan, dan membangun arsitektur berbasis peristiwa. Untuk informasi selengkapnya, silakan lihat Panduan Pengguna Amazon EventBridge.

Topik

- Pemantauan Amazon Transcribe dengan Amazon CloudWatch
- Pemantauan Amazon Transcribe dengan AWS CloudTrail
- Menggunakan Amazon EventBridge dengan Amazon Transcribe

Pemantauan Amazon Transcribe dengan Amazon CloudWatch

Anda dapat memantau Amazon Transcribe penggunaan CloudWatch, yang mengumpulkan data mentah dan memprosesnya menjadi metrik yang dapat dibaca, mendekati waktu nyata. Statistik ini disimpan untuk jangka waktu 15 bulan, sehingga Anda dapat mengakses informasi historis dan mendapatkan perspektif yang lebih baik tentang performa aplikasi atau layanan web Anda. Anda juga dapat mengatur alarm yang memperhatikan ambang batas tertentu dan mengirim notifikasi atau mengambil tindakan saat ambang batas tersebut terpenuhi. Untuk informasi selengkapnya, silakan lihat Panduan Pengguna CloudWatch .

Menggunakan Amazon CloudWatch metrik dan dimensi dengan Amazon Transcribe

Amazon Transcribe mendukung CloudWatch metrik dan dimensi, yang merupakan data yang dapat membantu Anda memantau kinerja. Kategori metrik yang didukung mencakup lalu lintas, kesalahan, transfer data, dan latensi yang terkait dengan pekerjaan transkripsi Anda. Metrik yang didukung terletak di namespace CloudWatch AWS/Transcribe.

> **Note**
>
> CloudWatch Metrik pemantauan tidak dipungut biaya dan tidak dihitung terhadap kuota CloudWatch layanan.

Untuk informasi selengkapnya tentang CloudWatch metrik, lihat Menggunakan Amazon CloudWatch metrik.

Pemantauan Amazon Transcribe dengan AWS CloudTrail

Amazon Transcribe terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil Amazon Transcribe oleh AWS Identity and Access Management (IAM) pengguna atau peran, atau oleh AWS layanan. CloudTrail menangkap semua panggilan API untuk Amazon Transcribe. Itu termasuk panggilan dari AWS Management Console dan panggilan kode ke Amazon Transcribe APIs, sebagai acara. Dengan membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara yang berkelanjutan, termasuk acara untuk Amazon Transcribe, ke Amazon S3 ember. Jika Anda tidak membuat jejak, Anda masih dapat melihat peristiwa terbaru di CloudTrail AWS Management Console dalam Sejarah acara. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat melihat setiap permintaan yang dibuat Amazon Transcribe, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk mempelajari selengkapnya CloudTrail, lihat Panduan AWS CloudTrail Pengguna.

Amazon Transcribe dan CloudTrail

CloudTrail diaktifkan pada Akun AWS saat Anda membuat akun. Ketika aktivitas terjadi Amazon Transcribe, aktivitas tersebut dicatat dalam suatu CloudTrail peristiwa bersama dengan Layanan AWS peristiwa lain dalam sejarah CloudTrail Peristiwa. Anda dapat melihat, mencari, dan mengunduh acara terbaru di situs Anda Akun AWS. Untuk informasi lain, lihat Melihat Peristiwa dengan Riwayat Peristiwa CloudTrail.

Untuk mendapatkan catatan acara yang sedang berlangsung di Anda Akun AWS, termasuk acara untuk Amazon Transcribe, buat jejak. Trail adalah konfigurasi yang memungkinkan CloudTrail untuk mengirimkan peristiwa sebagai file log ke Amazon S3 bucket tertentu. CloudTrail file log berisi satu atau lebih entri log. Peristiwa menunjukkan satu permintaan dari sumber mana pun. Ini mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. CloudTrail file log bukanlah jejak tumpukan yang diurutkan dari panggilan API publik, jadi file tersebut tidak muncul dalam urutan tertentu.

Secara default, saat Anda membuat jejak di AWS Management Console, jejak berlaku untuk semua Wilayah AWS. Jejak mencatat peristiwa dari semua Wilayah AWS di AWS partisi dan mengirimkan file log ke Amazon S3 bucket yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi lainnya Layanan AWS untuk menganalisis lebih lanjut dan menindaklanjuti data peristiwa yang dikumpulkan dalam CloudTrail log. Untuk informasi selengkapnya, lihat:

- Gambaran Umum untuk Membuat Jejak
- CloudTrail Layanan dan Integrasi yang Didukung
- Mengkonfigurasi Amazon SNS Pemberitahuan untuk CloudTrail
- Menerima File Log CloudTrail dari Beberapa Wilayah dan Menerima File Log CloudTrail dari Beberapa Akun

CloudTrail mencatat semua Amazon Transcribe tindakan, yang didokumentasikan dalam Referensi API. Misalnya, `CreateVocabulary`, `GetTranscriptionJob`, dan `StartTranscriptionJob`operasi menghasilkan entri dalam file CloudTrail log. Saat CloudTrail mencatat operasi Amazon Transcribe API, entri CloudTrail log menggunakan string kosong untuk informasi sensitif dalam parameter permintaan dan respons, seperti nilai Amazon S3 URI.

Setiap entri peristiwa atau log berisi informasi tentang entitas yang membuat permintaan tersebut. Informasi ini membantu Anda menentukan hal berikut:

- Apakah permintaan dibuat dengan root atau kredensi IAM pengguna
- Apakah permintaan dibuat dengan kredensi keamanan sementara untuk IAM peran atau pengguna gabungan
- Apakah permintaan dibuat oleh orang lain Layanan AWS

Untuk informasi lain, lihat Elemen userIdentity CloudTrail.

Anda juga dapat menggabungkan file Amazon Transcribe log dari beberapa Wilayah AWS dan beberapa Akun AWS ke dalam satu Amazon S3 bucket. Untuk informasi selengkapnya, lihat Menerima File CloudTrail Log dari Beberapa Wilayah dan Menerima File CloudTrail Log dari Beberapa Akun.

Contoh: entri file Amazon Transcribe log

Trail adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai file log ke Amazon S3 bucket tertentu. CloudTrail file log berisi satu atau lebih entri log. Peristiwa menunjukkan satu permintaan dari sumber mana pun. Ini mencakup informasi tentang tindakan yang diminta sebagai tanggal dan waktu tindakan, dan parameter permintaan. CloudTrail file log bukanlah jejak tumpukan yang diurutkan dari panggilan API publik, jadi file tersebut tidak muncul dalam urutan tertentu.

Panggilan ke operasi `StartTranscriptionJob`dan `GetTranscriptionJob`API membuat entri berikut.

> **Note**
>
> Saat CloudTrail mencatat operasi Amazon Transcribe API, entri CloudTrail log menggunakan string kosong untuk informasi sensitif dalam parameter permintaan dan respons, seperti nilai Amazon S3 URI.


```
{
    "Records": [
        {
            "eventVersion": "1.05",
            "userIdentity": {
                "type": "IAMUser",
                "principalId": "111122223333",
                "arn": "arn:aws:iam:us-west-2:111122223333:user/my-user-name",
                "accountId": "111122223333",
                "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
                "userName": "my-user-name"
            },
            "eventTime": "2022-03-07T15:03:45Z",
            "eventSource": "transcribe.amazonaws.com",
            "eventName": "StartTranscriptionJob",
            "awsRegion": "us-west-2",
            "sourceIPAddress": "127.0.0.1",
            "userAgent": "[]",
```

```
            "requestParameters": {
                "mediaFormat": "flac",
                "languageCode": "en-US",
                "transcriptionJobName": "my-first-transcription-job",
                "media": {
                    "mediaFileUri": ""
                }
            },
            "responseElements": {
                "transcriptionJob": {
                    "transcriptionJobStatus": "IN_PROGRESS",
                    "mediaFormat": "flac",
                    "creationTime": "2022-03-07T15:03:44.229000-08:00",
                    "transcriptionJobName": "my-first-transcription-job",
                    "languageCode": "en-US",
                    "media": {
                        "mediaFileUri": ""
                    }
                }
            },
            "requestID": "47B8E8D397DCE7A6",
            "eventID": "cdc4b7ed-e171-4cef-975a-ad829d4123e8",
            "eventType": "AwsApiCall",
            "recipientAccountId": "111122223333"
        },
        {
            "eventVersion": "1.05",
            "userIdentity": {
                "type": "IAMUser",
                "principalId": "111122223333",
                "arn": "arn:aws:iam:us-west-2:111122223333:user/my-user-name",
                "accountId": "111122223333",
                "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
                "userName": "my-user-name"
            },
            "eventTime": "2022-03-07T15:07:11Z",
            "eventSource": "transcribe.amazonaws.com",
            "eventName": "GetTranscriptionJob",
            "awsRegion": "us-west-2",
            "sourceIPAddress": "127.0.0.1",
            "userAgent": "[]",
            "requestParameters": {
                "transcriptionJobName": "my-first-transcription-job"
            },
```

```
            "responseElements": {
                "transcriptionJob": {
                    "settings": {

                    },
                    "transcriptionJobStatus": "COMPLETED",
                    "mediaFormat": "flac",
                    "creationTime": "2022-03-07T15:03:44.229000-08:00",
                    "transcriptionJobName": "my-first-transcription-job",
                    "languageCode": "en-US",
                    "media": {
                        "mediaFileUri": ""
                    },
                    "transcript": {
                        "transcriptFileUri": ""
                    }
                }
            },
            "requestID": "BD8798EACDD16751",
            "eventID": "607b9532-1423-41c7-b048-ec2641693c47",
            "eventType": "AwsApiCall",
            "recipientAccountId": "111122223333"
        }
    ]
}
```


Menggunakan Amazon EventBridge dengan Amazon Transcribe

Dengan Amazon EventBridge, Anda dapat menanggapi perubahan status dalam Amazon Transcribe pekerjaan Anda dengan memulai acara di tempat lain Layanan AWS. Saat pekerjaan transkripsi berubah status, EventBridge secara otomatis mengirimkan peristiwa ke aliran peristiwa. Anda membuat aturan yang menentukan peristiwa yang ingin Anda pantau dalam aliran peristiwa dan tindakan yang EventBridge harus diambil ketika peristiwa tersebut terjadi. Misalnya, merutekan acara ke layanan lain (atau target), yang kemudian dapat mengambil tindakan. Anda dapat, misalnya, mengonfigurasi aturan untuk merutekan acara ke AWS Lambda fungsi saat pekerjaan transkripsi berhasil diselesaikan. Untuk menentukan EventBridge aturan, lihat bagian berikut.

Anda dapat menerima pemberitahuan untuk acara melalui beberapa saluran, termasuk email, Pengembang Amazon Q dalam pemberitahuan obrolan aplikasi obrolan, atau pemberitahuan AWS Console Mobile Applicationpush. Anda juga dapat melihat notifikasi di Pusat Notifikasi Konsol. Jika Anda ingin mengatur notifikasi, Anda dapat menggunakannya Notifikasi Pengguna AWS. Notifikasi

Pengguna AWS mendukung agregasi, yang dapat mengurangi jumlah notifikasi yang Anda terima selama acara tertentu.

Mendefinisikan aturan EventBridge

Untuk menentukan EventBridge aturan, gunakan AWS Management Console. Saat Anda menentukan aturan, gunakan Amazon Transcribe sebagai nama layanan. Untuk contoh cara membuat EventBridge aturan, lihat Amazon EventBridge aturan.

Sebelum menggunakan EventBridge, perhatikan definisi berikut:

- Peristiwa —Sebuah peristiwa menunjukkan perubahan status salah satu pekerjaan transkripsi Anda. Misalnya, ketika `TranscriptionJobStatus` suatu pekerjaan berubah dari `IN_PROGRESS` ke`COMPLETED`.
- Target — Target adalah target lain Layanan AWS yang memproses suatu peristiwa. Misalnya, AWS Lambda atau Amazon Simple Notification Service (Amazon SNS). Target menerima peristiwa dalam format JSON.
- Aturan —Aturan cocok dengan peristiwa masuk yang EventBridge ingin Anda perhatikan dan merutekkannya ke target atau target untuk diproses. Jika aturan merutekan peristiwa ke beberapa target, semua target memproses peristiwa secara paralel. Aturan dapat menyesuaikan JSON yang dikirim ke target.

Amazon EventBridge peristiwa dipancarkan atas dasar upaya terbaik. Untuk informasi selengkapnya tentang membuat dan mengelola acara di EventBridge, lihat Amazon EventBridge acara di Panduan Amazon EventBridge Pengguna.

Berikut ini adalah contoh EventBridge aturan untuk Amazon Transcribe itu dimulai ketika status pekerjaan transkripsi berubah menjadi `COMPLETED` atau. `FAILED`

```
{
    "source": [
        "aws.transcribe"
    ],
    "detail-type": [
        "Transcribe Job State Change"
    ],
    "detail": {
        "TranscriptionJobStatus": [
            "COMPLETED",
```

```
            "FAILED"
        ]
    }
}
```

Aturan berisi bidang-bidang berikut:

- `source`—Sumber acara. Karena Amazon Transcribe, ini selaluaws.transcribe.
- `detail-type`—Pengidentifikasi untuk detail acara. Karena Amazon Transcribe, ini selaluTranscribe Job State Change.
- `detail`—Status pekerjaan baru dari pekerjaan transkripsi. Dalam contoh ini, aturan memulai peristiwa ketika status pekerjaan berubah menjadi `COMPLETED` atau`FAILED`.

Amazon Transcribe acara

Amazon EventBridge mencatat beberapa Amazon Transcribe peristiwa:

- Acara pekerjaan transkripsi
- Acara identifikasi bahasa
- Acara Panggilan Analytics
- Panggilan acara pasca-panggilan Analytics
- Acara kosakata

Semua peristiwa ini berisi bidang bersama berikut:

- `version`: Versi data acara. Nilai ini selalu `0`.
- `id`: Pengidentifikasi unik yang dihasilkan oleh EventBridge untuk acara tersebut.
- `detail-type`: Pengenal untuk rincian acara. Misalnya, `Transcribe Job State Change`.
- `source`: Sumber acara. Untuk Amazon Transcribe ini selaluaws.transcribe.
- `account`: Akun AWS ID akun yang menghasilkan panggilan API.
- `time`: Tanggal dan waktu acara disampaikan.
- `region`: Wilayah AWS Di mana permintaan dibuat.
- `resources`: Sumber daya yang digunakan oleh panggilan API. Karena Amazon Transcribe, bidang ini selalu kosong.

- `detail`: Detail tambahan tentang acara tersebut.
  - `FailureReason`: Bidang ini hadir jika status atau status berubah`FAILED`, dan menjelaskan alasan untuk `FAILED` status atau status.
  - Setiap jenis acara memiliki bidang unik tambahan yang ditampilkan di bawah`detail`. Bidang unik ini didefinisikan dalam bagian berikut setelah setiap contoh peristiwa.

Acara pekerjaan transkripsi

Ketika status pekerjaan berubah dari `IN_PROGRESS` ke `COMPLETED` atau`FAILED`, Amazon Transcribe menghasilkan peristiwa. Untuk mengidentifikasi pekerjaan yang mengubah status dan memulai acara di target Anda, gunakan `TranscriptionJobName` bidang acara. Sebuah Amazon Transcribe acara berisi informasi berikut. `FailureReason`Bidang ditambahkan di bawah `detail` jika status pekerjaan transkripsi Anda. `FAILED`

Perhatikan bahwa peristiwa ini hanya berlaku untuk operasi `StartTranscriptionJob`API.

```
{
    "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`: Nama unik yang Anda pilih untuk pekerjaan transkripsi Anda.
- `TranscriptionJobStatus` : Status pekerjaan transkripsi. Ini bisa `COMPLETED` atau`FAILED`.

Acara identifikasi bahasa

Saat Anda mengaktifkan identifikasi bahasa otomatis Amazon Transcribe , buat peristiwa saat status identifikasi bahasa adalah `COMPLETED` atau`FAILED`. Untuk mengidentifikasi pekerjaan yang mengubah status dan memulai acara di target Anda, gunakan `JobName` bidang acara. Sebuah

Amazon Transcribe acara berisi informasi berikut. `FailureReason`Bidang ditambahkan di bawah `detail` jika status identifikasi bahasa Anda`FAILED`.

Perhatikan bahwa peristiwa ini hanya berlaku untuk operasi `StartTranscriptionJob`API saat `LanguageIdSettings`parameter disertakan.

```
{
    "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`: Untuk pekerjaan transkripsi, nilai ini harus`TranscriptionJob`.
- `JobName`: Nama unik dari pekerjaan transkripsi Anda.
- `LanguageIdentificationStatus`: Status identifikasi bahasa dalam pekerjaan transkripsi. Ini bisa `COMPLETED` atau`FAILED`.

Acara Panggilan Analytics

Ketika status pekerjaan Call Analytics berubah dari `IN_PROGRESS` ke `COMPLETED` atau`FAILED`, Amazon Transcribe menghasilkan peristiwa. Untuk mengidentifikasi pekerjaan Analytics Panggilan yang mengubah status dan memulai acara di target Anda, gunakan `JobName` bidang acara. Sebuah Amazon Transcribe acara berisi informasi berikut. `FailureReason`Kolom ditambahkan di bawah `detail` jika status pekerjaan Call Analytics Anda`FAILED`.

Perhatikan bahwa peristiwa ini hanya berlaku untuk operasi `StartCallAnalyticsJob`API.

```
{
    "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`: Nama unik dari pekerjaan transkripsi Call Analytics Anda.
- `JobStatus`: Status pekerjaan transkripsi Call Analytics Anda. Ini bisa salah satu `COMPLETED` atau`FAILED`.
- `AnalyticsJobDetails`: Detail pekerjaan transkripsi Call Analytics Anda, termasuk informasi tentang fitur analitik yang dilewati.

Panggilan acara pasca-panggilan Analytics

Saat transkripsi analitik pasca-panggilan mengubah status dari `IN_PROGRESS` ke `COMPLETED` atau`FAILED`, Amazon Transcribe menghasilkan peristiwa. Untuk mengidentifikasi pekerjaan pasca-panggilan Analytics Panggilan yang mengubah status dan memulai acara di target Anda, gunakan bidang acara. `StreamingSessionId`

Perhatikan bahwa peristiwa ini hanya berlaku untuk operasi `StartCallAnalyticsStreamTranscription`API saat `PostCallAnalyticsSettings`parameter disertakan.

Sebuah `COMPLETED` acara berisi informasi berikut:

```
{
    "version": "0",
    "id": "event ID",
    "detail-type": "Call Analytics Post Call Job State Change",
    "source": "aws.transcribe",
    "account": "111122223333",
```

```
    "time": "timestamp",
    "region": "us-west-2",
    "resources": [],
    "detail": {
        "StreamingSessionId": "session-id",
        "PostCallStatus": "COMPLETED",
        "Transcript": {
            "RedactedTranscriptFileUri": "s3://amzn-s3-demo-bucket/my-output-files/my-
redacted-file.JSON",
            "TranscriptFileUri": "s3://amzn-s3-demo-bucket/my-output-files/my-
file.JSON"
        },
        "Media": {
            "MediaFileUri": "s3://amzn-s3-demo-bucket/my-output-files/my-redacted-
file.WAV",
            "RedactedMediaFileUri": "s3://amzn-s3-demo-bucket/my-output-files/my-
redacted-file.WAV"
        }
    }
}
```

Sebuah `FAILED` acara berisi informasi berikut:

```
{
    "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`: Nomor identifikasi yang ditetapkan untuk permintaan transkripsi Call Analytics real-time Anda.
- `PostCallStatus`: Status transkripsi Analytics Panggilan pasca-panggilan Anda. Ini bisa salah satu `COMPLETED` atau`FAILED`.

- `Transcript`: URI transkrip Anda yang disunting dan tidak disunting.
- `Media`: URI file audio Anda yang disunting dan tidak disunting.

AWS HealthScribe acara analitik pasca aliran

Ketika status berubah untuk operasi analitik AWS HealthScribe pasca-aliran, seperti ClinicalNoteGenerationResultperubahan dari `IN_PROGRESS` ke`COMPLETED`, AWS HealthScribe menghasilkan peristiwa dengan informasi berikut:

```
{
   "version":"0",
   "id":"event ID",
   "detail-type":"MedicalScribe Post Stream Analytics Update",
   "source":"aws.transcribe",
   "account":"111122223333",
   "time":"timestamp",
   "region":"us-east-1",
   "resources":[],
   "detail":{
      "SessionId": <SessionID>,
      "UpdateType": "ClinicalNoteGenerationResult",
      "ClinicalNoteGenerationResult": {
          "ClinicalNoteOutputLocation": s3://amzn-s3-demo-bucket/clinical-note-output-files/clinical-notes.JSON,
          "TranscriptOutputLocation": s3://amzn-s3-demo-bucket/my-output-files/my-file.JSON,
          "Status": <IN_PROGRESS | COMPLETED | FAILED>,
          "FailureReason": <failure_reason>
    }
   }
}
```

- `UpdateType`: Jenis operasi analitik pasca-aliran yang menghasilkan acara. Isi objek hasil bervariasi tergantung pada`UpdateType`.
- `SessionId`: Nomor identifikasi untuk AWS HealthScribe streaming Anda. Gunakan ID ini untuk mengidentifikasi sesi streaming yang berasal dan kemudian temukan analitik pasca-aliran yang menghasilkan acara.
- `Status`: Status operasi analitik pasca-aliran. Ini dapat berupa `IN_PROGRESS`, `COMPLETED`, atau `FAILED`.

- `ClinicalNoteOutputLocation`: URI dari bucket Amazon S3 keluaran untuk. `ClinicalNoteGenerationResult`
- `TranscriptOutputLocation`: URI transkrip Anda.

Acara kosakata

Ketika status kosakata kustom berubah dari `PENDING` ke `READY` atau`FAILED`, Amazon Transcribe menghasilkan peristiwa. Untuk mengidentifikasi kosakata khusus yang mengubah status dan memulai acara di target Anda, gunakan bidang acara. `VocabularyName` Sebuah Amazon Transcribe acara berisi informasi berikut. `FailureReason`Bidang ditambahkan di bawah `detail` jika status kosakata kustom Anda. `FAILED`

> **Note**
>
> Peristiwa ini hanya berlaku untuk operasi `CreateVocabulary`API.

```
{
    "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`: Nama unik kosakata kustom Anda.
- `VocabularyState`: Status pemrosesan kosakata kustom Anda. Ini bisa `READY` atau`FAILED`.

Validasi kepatuhan untuk Amazon Transcribe

Untuk mempelajari apakah an Layanan AWS berada dalam lingkup program kepatuhan tertentu, lihat Layanan AWS di Lingkup oleh Program Kepatuhan Layanan AWS dan pilih program kepatuhan yang Anda minati. Untuk informasi umum, lihat Program AWS Kepatuhan Program AWS .

Anda dapat mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat Mengunduh Laporan di AWS Artifact .

Tanggung jawab kepatuhan Anda saat menggunakan Layanan AWS ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, dan hukum dan peraturan yang berlaku. AWS menyediakan sumber daya berikut untuk membantu kepatuhan:

- Kepatuhan dan Tata Kelola Keamanan – Panduan implementasi solusi ini membahas pertimbangan arsitektur serta memberikan langkah-langkah untuk menerapkan fitur keamanan dan kepatuhan.
- Referensi Layanan yang Memenuhi Syarat HIPAA — Daftar layanan yang memenuhi syarat HIPAA. Tidak semua memenuhi Layanan AWS syarat HIPAA.
- AWS Sumber Daya AWS — Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- AWS Panduan Kepatuhan Pelanggan - Memahami model tanggung jawab bersama melalui lensa kepatuhan. Panduan ini merangkum praktik terbaik untuk mengamankan Layanan AWS dan memetakan panduan untuk kontrol keamanan di berbagai kerangka kerja (termasuk Institut Standar dan Teknologi Nasional (NIST), Dewan Standar Keamanan Industri Kartu Pembayaran (PCI), dan Organisasi Internasional untuk Standardisasi (ISO)).
- Mengevaluasi Sumber Daya dengan Aturan dalam Panduan AWS Config Pengembang — AWS Config Layanan menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- AWS Security Hub— Ini Layanan AWS memberikan pandangan komprehensif tentang keadaan keamanan Anda di dalamnya AWS. Security Hub menggunakan kontrol keamanan untuk sumber daya AWS Anda serta untuk memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik. Untuk daftar layanan dan kontrol yang didukung, lihat Referensi kontrol Security Hub.
- Amazon GuardDuty — Ini Layanan AWS mendeteksi potensi ancaman terhadap beban kerja Akun AWS, kontainer, dan data Anda dengan memantau lingkungan Anda untuk aktivitas mencurigakan dan berbahaya. GuardDuty dapat membantu Anda mengatasi berbagai persyaratan kepatuhan,

seperti PCI DSS, dengan memenuhi persyaratan deteksi intrusi yang diamanatkan oleh kerangka kerja kepatuhan tertentu.

- AWS Audit ManagerIni Layanan AWS membantu Anda terus mengaudit AWS penggunaan Anda untuk menyederhanakan cara Anda mengelola risiko dan kepatuhan terhadap peraturan dan standar industri.

Ketahanan di Amazon Transcribe

Infrastruktur AWS global dibangun di sekitar Wilayah AWS dan Availability Zones. Wilayah AWS menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi, yang terhubung dengan latensi rendah, throughput tinggi, dan jaringan yang sangat redundan. Dengan Zona Ketersediaan, Anda dapat merancang serta mengoperasikan aplikasi dan basis data yang secara otomatis melakukan fail over di antara zona tanpa gangguan. Zona Ketersediaan memiliki ketersediaan dan toleransi kesalahan yang lebih baik, dan dapat diskalakan dibandingkan infrastruktur pusat data tunggal atau multi tradisional.

Untuk informasi selengkapnya tentang Wilayah AWS dan Availability Zone, lihat Infrastruktur AWS Global.

Keamanan infrastruktur di Amazon Transcribe

Sebagai layanan terkelola, Amazon Transcribe dilindungi oleh keamanan jaringan AWS global. Untuk informasi tentang layanan AWS keamanan dan cara AWS melindungi infrastruktur, lihat Keamanan AWS Cloud. Untuk mendesain AWS lingkungan Anda menggunakan praktik terbaik untuk keamanan infrastruktur, lihat Perlindungan Infrastruktur dalam Kerangka Kerja yang AWS Diarsiteksikan dengan Baik Pilar Keamanan.

Anda menggunakan panggilan API yang AWS dipublikasikan untuk mengakses Amazon Transcribe melalui jaringan. Klien harus mendukung hal-hal berikut:

- Keamanan Lapisan Pengangkutan (TLS). Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Sandi cocok dengan sistem kerahasiaan maju sempurna (perfect forward secrecy, PFS) seperti DHE (Ephemeral Diffie-Hellman) atau ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Sebagian besar sistem modern seperti Java 7 dan versi lebih baru mendukung mode-mode ini.

Selain itu, permintaan harus ditandatangani menggunakan ID kunci akses dan kunci akses rahasia yang terkait dengan prinsipal IAM. Atau Anda bisa menggunakan AWS Security Token Service (AWS STS) untuk membuat kredensial keamanan sementara guna menandatangani permintaan.

Analisis dan manajemen kerentanan di Amazon Transcribe

Konfigurasi dan kontrol TI adalah tanggung jawab bersama antara AWS dan Anda, pelanggan kami. Untuk informasi selengkapnya, lihat model tanggung jawab AWS bersama.

Amazon Transcribe dan antarmuka titik akhir VPC ()AWS PrivateLink

Anda dapat membuat koneksi pribadi antara VPC Anda dan Amazon Transcribe dengan membuat antarmuka VPC endpoint. Endpoint antarmuka didukung oleh AWS PrivateLink, teknologi yang dapat Anda gunakan untuk mengakses secara pribadi Amazon Transcribe APIs tanpa gateway internet, perangkat NAT, koneksi VPN, atau koneksi. AWS Direct Connect Instans di VPC Anda tidak memerlukan alamat IP publik untuk berkomunikasi. Amazon Transcribe APIs Lalu lintas antara VPC Anda dan Amazon Transcribe tidak meninggalkan jaringan Amazon.

Setiap titik akhir antarmuka diwakili oleh satu atau beberapa Antarmuka Jaringan Elastis di subnet Anda.

Untuk informasi selengkapnya, lihat Antarmuka VPC endpoint (AWS PrivateLink) dalam Panduan Pengguna Amazon VPC.

Pertimbangan untuk titik akhir Amazon Transcribe VPC

Sebelum menyiapkan titik akhir VPC antarmuka Amazon Transcribe, pastikan Anda meninjau properti dan batasan titik akhir Antarmuka di Panduan Pengguna.Amazon VPC

Amazon Transcribe mendukung panggilan ke semua tindakan API-nya dari VPC Anda.

Buat VPC endpoint antarmuka untuk Amazon Transcribe

Anda dapat membuat titik akhir VPC untuk Amazon Transcribe layanan menggunakan atau. Amazon VPC AWS Management Console AWS CLI Untuk informasi selengkapnya, lihat Membuat titik akhir antarmuka di Panduan Amazon VPC Pengguna.

Untuk transkripsi batch di Amazon Transcribe, buat titik akhir VPC menggunakan nama layanan berikut:

- com.amazonaws. *us-west-2*.transkripsikan

Untuk streaming transkripsi di Amazon Transcribe, buat titik akhir VPC menggunakan nama layanan berikut:

- com.amazonaws. *us-west-2*.transcribestreaming

Jika Anda mengaktifkan DNS pribadi untuk titik akhir, Anda dapat membuat permintaan API untuk Amazon Transcribe menggunakan nama DNS default untuk Wilayah AWS, misalnya,. `transcribestreaming.us-east-2.amazonaws.com`

Untuk informasi selengkapnya, lihat Mengakses layanan melalui titik akhir antarmuka di Amazon VPC Panduan Pengguna.

Membuat kebijakan titik akhir VPC untuk Amazon Transcribe

Anda dapat melampirkan kebijakan titik akhir ke titik akhir VPC Anda yang mengontrol akses ke layanan streaming atau layanan transkripsi batch. Amazon Transcribe Kebijakan titik akhir menentukan informasi berikut:

- Prinsipal yang dapat melakukan tindakan.
- Tindakan yang dapat dilakukan.
- Sumber daya yang menjadi target tindakan.

Untuk informasi selengkapnya, lihat Mengontrol akses ke layanan dengan titik akhir VPC di Panduan Pengguna.Amazon VPC

Contoh: Kebijakan titik akhir VPC untuk Amazon Transcribe tindakan transkripsi batch

Berikut ini adalah contoh kebijakan titik akhir untuk transkripsi batch di. Amazon Transcribe Jika dilampirkan ke sebuah titik akhir, kebijakan ini memberikan akses ke tindakan Amazon Transcribe yang terdaftar untuk semua yang utama di semua sumber daya.

```
{
   "Statement":[
      {
         "Principal":"*",
         "Effect":"Allow",
         "Action":[
            "transcribe:StartTranscriptionJob",
            "transcribe:ListTranscriptionJobs"
```

```
         ],
         "Resource":"*"
      }
   ]
}
```

Contoh: Kebijakan titik akhir VPC untuk Amazon Transcribe streaming tindakan transkripsi

Berikut ini adalah contoh kebijakan titik akhir untuk transkripsi streaming di. Amazon Transcribe Jika dilampirkan ke sebuah titik akhir, kebijakan ini memberikan akses ke tindakan Amazon Transcribe yang terdaftar untuk semua yang utama di semua sumber daya.

```
{
   "Statement":[
      {
         "Principal":"*",
         "Effect":"Allow",
         "Action":[
            "transcribe:StartStreamTranscription",
            "transcribe:StartStreamTranscriptionWebsocket"
         ],
         "Resource":"*"
      }
   ]
}
```

Subnet bersama

Anda tidak dapat membuat, mendeskripsikan, memodifikasi, atau menghapus titik akhir VPC di subnet yang dibagikan dengan Anda. Namun, Anda dapat menggunakan titik akhir VPC di subnet yang dibagikan dengan Anda. Untuk informasi tentang berbagi VPC, lihat Berbagi VPC Anda dengan akun lain dalam panduan. Amazon Virtual Private Cloud

Praktik terbaik keamanan untuk Amazon Transcribe

Praktik terbaik berikut adalah pedoman umum dan tidak mewakili solusi keamanan yang lengkap. Karena praktik terbaik ini mungkin tidak sesuai atau cukup untuk lingkungan Anda, gunakan sebagai pertimbangan yang bermanfaat daripada resep.

- Gunakan enkripsi data, seperti konteks AWS KMS enkripsi

AWS KMS konteks enkripsi adalah peta teks biasa, pasangan kunci non-rahasia: nilai. Peta ini mewakili data otentikasi tambahan, yang dikenal sebagai pasangan konteks enkripsi, yang menyediakan lapisan keamanan tambahan untuk data Anda.

Untuk informasi lebih lanjut, lihatAWS KMS konteks enkripsi.

- Gunakan kredensi sementara bila memungkinkan

  Jika memungkinkan, gunakan kredensi sementara alih-alih kredensi jangka panjang, seperti kunci akses. Untuk skenario di mana Anda membutuhkan IAM pengguna dengan akses terprogram dan kredensi jangka panjang, kami sarankan Anda memutar kunci akses. Memutar kredensi jangka panjang secara teratur membantu Anda membiasakan diri dengan prosesnya. Ini berguna jika Anda pernah berada dalam situasi di mana Anda harus memutar kredensi, seperti ketika seorang karyawan meninggalkan perusahaan Anda. Kami menyarankan Anda menggunakan IAM akses informasi yang terakhir digunakan untuk memutar dan menghapus kunci akses dengan aman.

  Untuk informasi selengkapnya, lihat Memutar kunci akses dan Praktik terbaik keamanan di IAM.

- Gunakan IAM peran untuk aplikasi dan AWS layanan yang memerlukan Amazon Transcribe akses

  Gunakan IAM peran untuk mengelola kredensi sementara untuk aplikasi atau layanan yang perlu diakses. Amazon Transcribe Saat Anda menggunakan peran, Anda tidak perlu mendistribusikan kredensi jangka panjang, seperti kata sandi atau kunci akses, ke Amazon EC2 instans atau AWS layanan. IAM peran dapat menyediakan izin sementara yang dapat digunakan aplikasi saat mereka membuat permintaan ke AWS sumber daya.

  Untuk informasi selengkapnya, lihat IAM peran dan skenario umum untuk peran: Pengguna, aplikasi, dan layanan.

- Gunakan kontrol akses berbasis tag

  Anda dapat menggunakan tag untuk mengontrol akses di dalam Anda Akun AWS. Di Amazon Transcribe. tag dapat ditambahkan ke: pekerjaan transkripsi, kosakata khusus, filter kosakata khusus, dan model bahasa khusus.

  Untuk informasi lebih lanjut, lihatKontrol akses berbasis tag.

- Gunakan alat AWS pemantauan

  Pemantauan adalah bagian penting dari menjaga keandalan, keamanan, ketersediaan, dan kinerja Amazon Transcribe dan AWS solusi Anda. Anda dapat memantau Amazon Transcribe menggunakan CloudTrail.

Untuk informasi lebih lanjut, lihatPemantauan Amazon Transcribe dengan AWS CloudTrail.

- Aktifkan AWS Config

  AWS Config dapat menilai, mengaudit, dan mengevaluasi konfigurasi AWS sumber daya Anda. Dengan menggunakan AWS Config, Anda dapat meninjau perubahan konfigurasi dan hubungan antar AWS sumber daya. Anda juga dapat menyelidiki riwayat konfigurasi sumber daya terperinci dan menentukan kepatuhan Anda secara keseluruhan terhadap konfigurasi yang ditentukan dalam pedoman internal Anda. Ini dapat membantu Anda menyederhanakan audit kepatuhan, analisis keamanan, manajemen perubahan, dan pemecahan masalah operasional.

  Untuk informasi lebih lanjut, lihat Apa itu AWS Config?

Amazon Transcribe Medis

Amazon Transcribe Medis adalah layanan pengenalan suara otomatis (ASR) yang dirancang untuk profesional medis yang ingin menyalin ucapan terkait medis, seperti catatan yang ditentukan dokter, pemantauan keamanan obat, janji telemedicine, atau percakapan dokter-pasien. Amazon Transcribe Medis tersedia melalui streaming real-time (melalui mikrofon) atau transkripsi file yang diunggah (batch).

> **Important**
>
> Amazon Transcribe Medis bukanlah pengganti nasihat medis profesional, diagnosis, atau perawatan. Identifikasi ambang kepercayaan yang tepat untuk kasus penggunaan Anda, dan gunakan ambang kepercayaan tinggi dalam situasi yang membutuhkan akurasi tinggi. Untuk kasus penggunaan tertentu, hasil harus ditinjau dan diverifikasi oleh pengulas manusia yang terlatih dengan tepat. Amazon Transcribe Transkripsi medis hanya boleh digunakan dalam skenario perawatan pasien setelah ditinjau untuk akurasi dan penilaian medis yang baik oleh profesional medis terlatih.

Amazon Transcribe Medis beroperasi di bawah model tanggung jawab bersama, di mana AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan Amazon Transcribe Medis dan Anda bertanggung jawab untuk mengelola data Anda. Untuk informasi selengkapnya, lihat Model Tanggung Jawab Bersama.

Amazon Transcribe Medis tersedia dalam bahasa Inggris AS (en-US).

Untuk hasil terbaik, gunakan format audio lossless, seperti FLAC atau WAV, dengan pengkodean PCM 16-bit. Amazon Transcribe Medical mendukung laju sampel 16.000 Hz atau lebih tinggi.

Untuk analisis transkrip Anda, Anda dapat menggunakan yang lain Layanan AWS, seperti Amazon Comprehend Medical.

Spesialisasi yang didukung

Keistimewaan	Sub-spesialisasi	Masukan audio
Kardiologi	none	streaming saja
Neurologi	none	streaming saja

Keistimewaan	Sub-spesialisasi	Masukan audio
Onkologi	none	streaming saja
Perawatan Primer	Kedokteran Keluarga	batch, streaming
Perawatan Primer	Penyakit Dalam	batch, streaming
Perawatan Primer	Obstetri dan Ginekologi (OB-GYN)	batch, streaming
Perawatan Primer	Pediatri	batch, streaming
Radiologi	none	streaming saja
Urologi	none	streaming saja

Ketersediaan dan kuota wilayah

Call Analytics didukung sebagai berikut Wilayah AWS:

Wilayah	Jenis transkripsi
af-south-1 (Cape Town)	batch
ap-east-1 (Hong Kong)	batch
ap-northeast-1 (Tokyo)	batch, streaming
ap-northeast-2 (Seoul)	batch, streaming
ap-south-1 (Mumbai)	batch
ap-southeast-1 (Singapura)	batch
ap-southeast-2 (Sydney)	batch, streaming
ca-central-1 (Kanada, Tengah)	batch, streaming
eu-central-1 (Frankfurt am Main)	batch, streaming

Wilayah	Jenis transkripsi
eu-north-1 (Stockholm)	batch
eu-west-1 (Irlandia)	batch, streaming
eu-west-2 (London)	batch, streaming
eu-west-3 (Paris)	batch
me-south-1 (Bahrain)	batch
sa-east-1 (sa-east-1)	batch, streaming
us-east-1 (Virginia Utara)	batch, streaming
us-east-2 (Ohio)	batch, streaming
us-gov-east-1 (GovCloud, AS-Timur)	batch, streaming
us-gov-west-1 (GovCloud, AS-Barat)	batch, streaming
us-west-1 (San Francisco)	batch
us-west-2 (Oregon)	batch, streaming

Perhatikan bahwa dukungan Wilayah berbeda untuk Amazon Transcribe Amazon Transcribe Medical, dan Call Analytics. .

Untuk mendapatkan titik akhir untuk setiap Wilayah yang didukung, lihat Titik akhir Layanan di Referensi AWS Umum.

Untuk daftar kuota yang berkaitan dengan transkripsi Anda, lihat kuota Layanan di Referensi Umum.AWS Beberapa kuota dapat diubah berdasarkan permintaan. Jika kolom Adjustable berisi 'Ya', Anda dapat meminta peningkatan. Untuk melakukannya, pilih tautan yang disediakan.

Spesialisasi dan persyaratan medis

Saat membuat pekerjaan transkripsi medis, Anda menentukan bahasa, spesialisasi medis, dan jenis audio dari file sumber. Anda memasukkan US English (en-US) sebagai bahasa dan `PRIMARYCARE`

sebagai spesialisasi medis. Memasuki perawatan primer sebagai nilainya memungkinkan Anda menghasilkan transkripsi dari audio sumber dalam spesialisasi medis berikut:

- Kedokteran Keluarga
- Penyakit Dalam
- Obstetri dan Ginekologi (OB-GYN)
- Pediatri

Anda memiliki pilihan antara dikte dan percakapan untuk jenis audio Anda. Pilih dikte untuk file audio di mana dokter memberikan laporan tentang kunjungan atau prosedur pasien. Pilih percakapan untuk file audio yang melibatkan percakapan antara dokter dan pasien atau percakapan antar dokter.

Untuk menyimpan output dari pekerjaan transkripsi Anda, pilih Amazon S3 bucket yang telah Anda buat. Untuk informasi selengkapnya tentang Amazon S3 bucket, lihat Memulai dengan Amazon Simple Storage Service.

Berikut ini adalah jumlah minimum parameter permintaan untuk dimasukkan dalam sampel JSON.

```
{
   "MedicalTranscriptionJobName": "my-first-transcription-job",
   "LanguageCode": "en-US",
   "Media": {
       "MediaFileUri": "s3://path to your audio file"
   },
   "OutputBucketName": “your output bucket name",
   "Specialty": "PRIMARYCARE",
   "Type": "CONVERSATION"
}
```

Amazon Transcribe Medis memungkinkan Anda menghasilkan transkripsi alternatif. Untuk informasi selengkapnya, lihat Menghasilkan transkripsi alternatif.

Anda juga dapat mengaktifkan partisi speaker atau mengidentifikasi saluran dalam audio Anda. Untuk informasi selengkapnya, silakan lihat Mengaktifkan partisi speaker dan Mentranskripsikan audio multi-saluran.

Mentranskripsikan istilah dan pengukuran medis

Amazon Transcribe Medis dapat mentranskripsikan istilah dan pengukuran medis. Amazon Transcribe Output medis singkatan untuk istilah lisan. Misalnya, “tekanan darah” ditranskripsikan

sebagai BP. Anda dapat menemukan daftar konvensi yang digunakan Amazon Transcribe Medis untuk istilah dan pengukuran medis dalam tabel di halaman ini. Kolom Istilah Lisan mengacu pada istilah yang diucapkan dalam audio sumber. Kolom Output mengacu pada singkatan yang Anda lihat di hasil transkripsi Anda.

Anda dapat melihat bagaimana istilah yang diucapkan dalam audio sumber sesuai dengan keluaran transkripsi di sini.

Istilah yang diucapkan dalam audio sumber	Singkatan yang digunakan dalam output	Contoh Output
Celcius	C	Suhu pasien adalah 37,4 C.
Celsius	C	Suhu pasien adalah 37,4 C.
Fahrenheit	F	Suhu pasien adalah 101 F.
gram	g	Massa 100 g diekstraksi dari pasien.
meter	m	Tinggi pasien 1,8 m.
kaki	ft	Pasien tingginya 6 kaki.
kilo	kg	Pasien memiliki berat 80 kg.
kilogram	kg	Pasien memiliki berat 80 kg.
c c	cc	Pasien menerima 100 cc larutan garam.
sentimeter kubik	cc	Pasien menerima 100 cc larutan garam.
mililiter	mL	Pasien diekskresikan 100 mL urin.
tekanan darah	BP	BP pasien meningkat.
b p	BP	BP pasien meningkat.

Istilah yang diucapkan dalam audio sumber	Singkatan yang digunakan dalam output	Contoh Output
X di atas Y	X/Y	Pasien BP adalah 120/80.
ketukan per menit	BPM	Pasien mengalami fibrilasi atrium dengan denyut jantung 160 BPM.
ketukan per menit	BPM	Pasien mengalami fibrilasi atrium dengan denyut jantung 160 BPM.
O 2	O2	Saturasi O2 pasien adalah 98%.
CO2	CO2	Pasien membutuhkan dukungan pernapasan untuk peningkatan CO2.
pasca operasi	POSTOP	Pasien datang untuk evaluasi POSTOP.
posting op	POSTOP	Pasien datang untuk evaluasi POSTOP.
pemindaian kucing	CT Scan	Indikasi pasien perdarahan otak diperlukan penggunaan CT Scan.
Pulsa 80	P 80	Vital pasien adalah P 80, R 17,...
Respirasi 17	R 17	Vital pasien adalah P 80, R 17,...
masuk dan keluar	I/O	Pasien adalah irama sinus I/O

Istilah yang diucapkan dalam audio sumber	Singkatan yang digunakan dalam output	Contoh Output
L lima	L5	Pungsi lumbal dilakukan antara L4 dan L5

Mentranskripsikan angka

Amazon Transcribe Medis mentranskripsikan digit sebagai angka, bukan kata-kata. Misalnya, nomor yang diucapkan “seribu dua ratus empat puluh dua” ditranskripsikan sebagai. `1242`

Angka ditranskripsikan sesuai dengan aturan berikut.

Aturan	Deskripsi
Ubah angka kardinal lebih besar dari 10 menjadi angka.	• “Lima puluh lima” > 55 • “seratus” > 100 • “Seribu tiga puluh satu” > 1031 • “Seratus dua puluh tiga juta empat ratus lima puluh enam ribu tujuh ratus delapan sembilan” > 123.456.789
Ubah angka kardinal diikuti dengan “juta” atau “miliar” menjadi angka diikuti dengan kata ketika “juta” atau “miliar” tidak diikuti oleh angka.	• “seratus juta” > 100 juta • “satu milyar” > 1 miliar • “dua poin tiga juta” > 2,3 juta
Ubah nomor urut lebih besar dari 10 menjadi angka.	• “Empat puluh tiga” > ke-43 • “jalan dua puluh enam” > jalan ke-26
Konversikan pecahan ke format numeriknya.	• “seperempat” > 1/4 • “tiga enam belas” > 3/16 • “setengah"> 1/2 • “seperseratus"> 1/100

Aturan	Deskripsi
Ubah angka kurang dari 10 menjadi digit jika ada lebih dari satu berturut-turut.	• “tiga empat lima” > 345 • “Nomor telepon saya empat dua lima lima lima lima satu dua dua” > 4255551212
Desimal ditunjukkan dengan “titik” atau “titik.”	• “tiga ratus tiga titik lima” > 303.5 • “tiga poin dua puluh tiga” > 3.23 • “titik nol empat” > 0,4 • “titik tiga” > 0,3
Ubah kata “persen” setelah angka menjadi tanda persen.	• “dua puluh tiga persen"> 23% • “dua puluh tiga poin empat lima persen” > 23,45%
Ubah kata “dolar,” “dolar AS,” “dolar Australia ,” “AUD,” atau “USD” setelah angka menjadi simbol dolar sebelum angka tersebut.	• “satu dolar dan lima belas sen” > $1,15 • “dua puluh tiga USD” > $23 • “dua puluh tiga dolar Australia” > $23
Ubah kata “pound,” atau “miligram” menjadi “lbs” atau “mg”.	• “dua puluh tiga pon” > 23 lbs • “empat puluh lima miligram” > 45 mg
Ubah kata “rupee”, “rupee India,” atau “INR” setelah angka menjadi tanda rupee (#) sebelum nomor tersebut.	• “dua puluh tiga rupee” > #23 • “lima puluh rupee tiga puluh paise” > #50,30
Ubah waktu menjadi angka.	• “tujuh m waktu standar timur” > 7 pagi waktu standar timur • “dua belas tiga puluh p m” > 12:30 malam
Gabungkan tahun yang dinyatakan sebagai dua digit menjadi empat. Hanya berlaku untuk abad ke-20, 21, dan 22.	• “Sembilan belas enam puluh dua” > 1962 • “tahun ini dua puluh dua belas” > tahun 2012 • “dua puluh sembilan belas” > 2019 • “dua puluh satu tiga puluh” > 2130

Aturan	Deskripsi
Ubah tanggal menjadi angka.	• “Lima Mei dua puluh dua belas” > 5 Mei 2012 • “Lima Mei dua puluh dua belas” > 5 Mei 2012 • “lima Mei dua puluh dua belas” > 5 Mei 2012
Pisahkan rentang angka dengan kata “to.”	• “dua puluh tiga sampai tiga puluh tujuh” > 23 hingga 37

Mentranskripsikan percakapan medis

Anda dapat menggunakan Amazon Transcribe Medis untuk mentranskripsikan percakapan medis antara dokter dan pasien menggunakan pekerjaan transkripsi batch atau aliran waktu nyata. Pekerjaan transkripsi batch memungkinkan Anda untuk mentranskripsikan file audio. Untuk memastikan bahwa Amazon Transcribe Medis menghasilkan hasil transkripsi dengan akurasi setinggi mungkin, Anda harus menentukan spesialisasi medis dokter dalam pekerjaan atau aliran transkripsi Anda.

Anda dapat menuliskan kunjungan klinis-pasien dalam spesialisasi medis berikut:

- Kardiologi — hanya tersedia dalam transkripsi streaming
- Neurologi — hanya tersedia dalam transkripsi streaming
- Onkologi — hanya tersedia dalam transkripsi streaming
- Perawatan Primer - termasuk jenis praktik medis berikut:
  - Obat keluarga
  - Penyakit dalam
  - Obstetri dan Ginekologi (OB-GYN)
  - Pediatri
- Urologi — hanya tersedia dalam transkripsi streaming

Anda dapat meningkatkan akurasi transkripsi dengan menggunakan kosakata khusus medis. Untuk informasi tentang cara kerja kosakata khusus medis, lihat. Meningkatkan akurasi transkripsi dengan kosakata khusus medis

Secara default, Amazon Transcribe Medical mengembalikan transkripsi dengan tingkat kepercayaan tertinggi. Jika Anda ingin mengonfigurasinya untuk mengembalikan transkripsi alternatif, lihatMenghasilkan transkripsi alternatif.

Untuk informasi tentang bagaimana angka dan pengukuran medis muncul dalam output transkripsi, lihat Mentranskripsikan angka danMentranskripsikan istilah dan pengukuran medis.

Topik

- Mentranskripsikan file audio percakapan medis
- Mentranskripsikan percakapan medis dalam aliran waktu nyata
- Mengaktifkan partisi speaker
- Mentranskripsikan audio multi-saluran

Mentranskripsikan file audio percakapan medis

Gunakan pekerjaan transkripsi batch untuk mentranskripsikan file audio percakapan medis. Anda dapat menggunakan ini untuk menyalin dialog klinis-pasien. Anda dapat memulai pekerjaan transkripsi batch baik di `StartMedicalTranscriptionJob`API atau. AWS Management Console

Saat Anda memulai pekerjaan transkripsi medis dengan `StartMedicalTranscriptionJob`API, Anda menentukan `PRIMARYCARE` sebagai nilai `Specialty` parameter.

AWS Management Console

Untuk mentranskripsikan dialog klinis-pasien ()AWS Management Console

Untuk menggunakan untuk AWS Management Console mentranskripsikan dialog klinis-pasien, buat pekerjaan transkripsi dan pilih Percakapan untuk jenis input Audio.

1. Masuk ke AWS Management Console.
2. Di panel navigasi, di bawah Amazon Transcribe Medis, pilih Pekerjaan transkripsi.
3. Pilih Buat tugas.
4. Pada halaman Tentukan detail pekerjaan, di bawah Pengaturan Job, tentukan yang berikut ini.
    a. Nama — nama pekerjaan transkripsi.
    b. Jenis input audio - Percakapan
5. Untuk bidang yang tersisa, tentukan Amazon S3 lokasi file audio Anda dan di mana Anda ingin menyimpan output dari pekerjaan transkripsi Anda.

6. Pilih Berikutnya.
7. Pilih Buat.

API

Untuk mentranskripsikan percakapan medis menggunakan pekerjaan transkripsi batch (API)

- Untuk `StartMedicalTranscriptionJob`API, tentukan yang berikut ini.

  a. Untuk`MedicalTranscriptionJobName`, tentukan nama unik di Anda Akun AWS.

  b. Untuk`LanguageCode`, tentukan kode bahasa yang sesuai dengan bahasa yang digunakan dalam file audio Anda dan bahasa filter kosakata Anda.

  c. Untuk `MediaFileUri` parameter `Media` objek, tentukan nama file audio yang ingin Anda transkripsikan.

  d. Untuk`Specialty`, tentukan spesialisasi medis dokter yang berbicara dalam file audio sebagai`PRIMARYCARE`.

  e. Untuk `Type`, tentukan `CONVERSATION`.

  f. Untuk`OutputBucketName`, tentukan Amazon S3 ember untuk menyimpan hasil transkripsi.

  Berikut ini adalah contoh permintaan yang menggunakan AWS SDK untuk Python (Boto3) untuk mentranskripsikan percakapan medis dari seorang dokter di `PRIMARYCARE` bidang spesialisasi dan pasien.

```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
job_name = "my-first-med-transcription-job"
job_uri = "s3://amzn-s3-demo-bucket/my-input-files/my-audio-file.flac"
transcribe.start_medical_transcription_job(
      MedicalTranscriptionJobName = job_name,
      Media = {
        'MediaFileUri': job_uri
      },
      OutputBucketName = 'amzn-s3-demo-bucket',
      OutputKey = 'output-files/',
      LanguageCode = 'en-US',
```

```
      Specialty = 'PRIMARYCARE',
      Type = 'CONVERSATION'
  )

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

Kode contoh berikut menunjukkan hasil transkripsi percakapan dokter-pasien.

```
{
    "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

Untuk mentranskripsikan percakapan medis menggunakan pekerjaan transkripsi batch ()AWS CLI

- Jalankan kode berikut.

```
aws transcribe start-medical-transcription-job \
--region us-west-2 \
--cli-input-json file://example-start-command.json
```

Kode berikut menunjukkan isi dari`example-start-command.json`.

```
{
      "MedicalTranscriptionJobName": "my-first-med-transcription-job",
      "Media": {
          "MediaFileUri": "s3://amzn-s3-demo-bucket/my-input-files/my-audio-
file.flac"
      },
      "OutputBucketName": "amzn-s3-demo-bucket",
      "OutputKey": "my-output-files/",
      "LanguageCode": "en-US",
      "Specialty": "PRIMARYCARE",
      "Type": "CONVERSATION"
  }
```

Mentranskripsikan percakapan medis dalam aliran waktu nyata

Anda dapat mentranskripsikan aliran audio percakapan medis menggunakan HTTP/2 atau protokol. WebSocket Untuk informasi tentang cara memulai streaming menggunakan WebSocket protokol, lihatMenyiapkan WebSocket aliran. Untuk memulai streaming HTTP/2, gunakan API. `StartMedicalStreamTranscription`

Anda dapat mentranskripsikan audio streaming dalam spesialisasi medis berikut:

- Kardiologi
- Neurologi
- Onkologi
- Perawatan Primer
- Urologi

Setiap spesialisasi medis mencakup banyak jenis prosedur dan janji temu. Oleh karena itu, dokter mendikte berbagai jenis catatan. Gunakan contoh berikut sebagai panduan untuk membantu Anda menentukan nilai parameter `specialty` URI WebSocket permintaan, atau `Specialty` parameter `StartMedicalStreamTranscription`API:

- Untuk konsultasi elektrofisiologi atau ekokardiografi, pilih. `CARDIOLOGY`
- Untuk onkologi medis, onkologi bedah, atau konsultasi onkologi radiasi, pilih. `ONCOLOGY`
- Untuk dokter yang memberikan konsultasi kepada pasien yang mengalami stroke, baik serangan iskemik transien atau serangan serebrovaskular, pilih. `NEUROLOGY`
- Untuk konsultasi seputar inkontinensia urin, pilih. `UROLOGY`
- Untuk pemeriksaan tahunan atau kunjungan perawatan mendesak, pilih`PRIMARYCARE`.
- Untuk kunjungan rawat inap rawat inap, pilih. `PRIMARYCARE`
- Untuk konsultasi mengenai kesuburan, ligasi tuba, pemasangan IUD, atau aborsi, pilih. `PRIMARYCARE`

AWS Management Console

Untuk mentranskripsikan percakapan medis streaming ()AWS Management Console

Untuk menggunakan AWS Management Console untuk mentranskripsikan dialog klinis-pasien dalam aliran waktu nyata, pilih opsi untuk menyalin percakapan medis, memulai aliran, dan mulai berbicara ke mikrofon.

1. Masuk ke AWS Management Console.
2. Di panel navigasi, di bawah Amazon Transcribe Medis, pilih Transkripsi waktu nyata.
3. Pilih Percakapan.
4. Untuk spesialisasi medis, pilih spesialisasi dokter.
5. Pilih Mulai streaming.
6. Bicaralah ke mikrofon.

Mentranskripsikan percakapan medis dalam aliran HTTP/2

Berikut ini adalah sintaks untuk parameter permintaan HTTP/2.

Untuk mentranskripsikan aliran HTTP/2 dari percakapan medis, gunakan `StartMedicalStreamTranscription`API dan tentukan yang berikut ini:

- `LanguageCode`- Kode bahasa. Nilai yang valid adalah `en-US`
- `MediaEncoding`— Pengkodean yang digunakan untuk audio input. Nilai yang valid adalah `pcm`, `ogg-opus`, dan `flac`.
- `Specialty`- Keistimewaan profesional medis.
- `Type` – `CONVERSATION`

Untuk meningkatkan akurasi transkripsi istilah tertentu dalam aliran waktu nyata, gunakan kosakata khusus. Untuk mengaktifkan kosakata kustom, tetapkan nilai `VocabularyName` parameter ke nama kosakata kustom yang ingin Anda gunakan. Untuk informasi selengkapnya, lihat Meningkatkan akurasi transkripsi dengan kosakata khusus medis.

Untuk memberi label pidato dari speaker yang berbeda, atur `ShowSpeakerLabel` parameternya ke`true`. Untuk informasi selengkapnya, lihat Mengaktifkan partisi speaker.

Untuk informasi lebih lanjut tentang menyiapkan aliran HTTP/2 untuk mentranskripsikan percakapan medis, lihat. Menyiapkan aliran HTTP/2

Mentranskripsikan percakapan medis dalam aliran WebSocket

Anda dapat menggunakan WebSocket permintaan untuk menyalin percakapan medis. Ketika Anda membuat WebSocket permintaan, Anda membuat URI presigned. URI ini berisi informasi yang diperlukan untuk mengatur aliran audio antara aplikasi Anda dan Amazon Transcribe Medis. Untuk informasi selengkapnya tentang membuat WebSocket permintaan, lihatMenyiapkan WebSocket aliran.

Gunakan template berikut untuk membuat URI presigned Anda.

```
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
```

Untuk meningkatkan akurasi transkripsi istilah tertentu dalam aliran waktu nyata, gunakan kosakata khusus. Untuk mengaktifkan kosakata kustom, tetapkan nilai `vocabulary-name` ke nama kosakata kustom yang ingin Anda gunakan. Untuk informasi selengkapnya, lihat Meningkatkan akurasi transkripsi dengan kosakata khusus medis.

Untuk memberi label pidato dari speaker yang berbeda, atur `show-speaker-label` parameter ke dalam`true`. Untuk informasi selengkapnya, lihat Mengaktifkan partisi speaker.

Untuk informasi selengkapnya tentang membuat pra-ditandatangani URIs, lihatMenyiapkan WebSocket aliran.

Mengaktifkan partisi speaker

Untuk mengaktifkan partisi speaker di Amazon Transcribe Medical, gunakan diarization speaker. Ini memungkinkan Anda untuk melihat apa yang dikatakan pasien dan apa yang dikatakan dokter dalam output transkripsi.

Saat Anda mengaktifkan diarisasi speaker, Amazon Transcribe Medis memberi label pada setiap pembicara dengan pengenal unik untuk setiap pembicara. Ucapan adalah unit pidato yang biasanya dipisahkan dari ucapan lain dengan diam. Dalam transkripsi batch, ucapan dari dokter dapat menerima label `spk_0` dan ucapan pasien dapat menerima label. `spk_1`

Jika ucapan dari satu pembicara tumpang tindih dengan ucapan dari pembicara lain, Amazon Transcribe Medis memerintahkannya dalam transkripsi pada waktu mulai mereka. Ucapan yang tumpang tindih dalam audio input tidak tumpang tindih dalam output transkripsi.

Anda dapat mengaktifkan diarisasi speaker saat Anda mentranskripsikan file audio menggunakan pekerjaan transkripsi batch, atau dalam aliran waktu nyata.

Topik

- Mengaktifkan partisi speaker dalam transkripsi batch
- Mengaktifkan partisi speaker dalam streaming waktu nyata

Mengaktifkan partisi speaker dalam transkripsi batch

Anda dapat mengaktifkan partisi speaker dalam pekerjaan transkripsi batch menggunakan `StartMedicalTranscriptionJob`API atau. AWS Management Console Ini memungkinkan Anda untuk mempartisi teks per pembicara dalam percakapan klinis-pasien dan menentukan siapa yang mengatakan apa dalam output transkripsi.

AWS Management Console

Untuk menggunakan AWS Management Console untuk mengaktifkan diarisasi speaker dalam pekerjaan transkripsi Anda, Anda mengaktifkan identifikasi audio dan kemudian partisi speaker.

1. Masuk ke AWS Management Console.
2. Di panel navigasi, di bawah Amazon Transcribe Medis, pilih Pekerjaan transkripsi.

3. Pilih Buat tugas.
4. Pada halaman Tentukan detail pekerjaan, berikan informasi tentang pekerjaan transkripsi Anda.
5. Pilih Berikutnya.
6. Aktifkan identifikasi Audio.
7. Untuk jenis identifikasi Audio, pilih Partisi speaker.
8. Untuk Jumlah speaker maksimum, masukkan jumlah speaker maksimum yang menurut Anda berbicara dalam file audio Anda.
9. Pilih Buat.

API

Untuk mengaktifkan partisi speaker menggunakan tugas transkripsi batch (API)

- Untuk `StartMedicalTranscriptionJob`API, tentukan yang berikut ini.

    a. Untuk`MedicalTranscriptionJobName`, tentukan nama yang unik di Anda Akun AWS.

    b. Untuk`LanguageCode`, tentukan kode bahasa yang sesuai dengan bahasa yang digunakan dalam file audio.

    c. Untuk `MediaFileUri` parameter `Media` objek, tentukan nama file audio yang ingin Anda transkripsikan.

    d. Untuk`Specialty`, tentukan spesialisasi medis dokter yang berbicara dalam file audio.

    e. Untuk `Type`, tentukan `CONVERSATION`.

    f. Untuk`OutputBucketName`, tentukan Amazon S3 ember untuk menyimpan hasil transkripsi.

    g. Untuk `Settings` objek, tentukan yang berikut ini.

        i. `ShowSpeakerLabels` – `true`.

        ii. `MaxSpeakerLabels`— Bilangan bulat antara 2 dan 10 untuk menunjukkan jumlah speaker yang menurut Anda berbicara dalam audio Anda.

Permintaan berikut menggunakan AWS SDK untuk Python (Boto3) untuk memulai pekerjaan transkripsi batch dari dialog pasien dokter perawatan primer dengan partisi pembicara diaktifkan.

```
from __future__ import print_function
import time
```

```
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
job_name = "my-first-transcription-job"
job_uri = "s3://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
transcribe.start_medical_transcription_job(
    MedicalTranscriptionJobName = job_name,
    Media={
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'amzn-s3-demo-bucket',
    OutputKey = 'my-output-files/',
    LanguageCode = 'en-US',
    Specialty = 'PRIMARYCARE',
    Type = 'CONVERSATION',
    OutputBucketName = 'amzn-s3-demo-bucket',
Settings = {'ShowSpeakerLabels': True,
         'MaxSpeakerLabels': 2
         }
         )
while True:
    status = transcribe.get_medical_transcription_job(MedicalTranscriptionJobName =
 job_name)
    if status['MedicalTranscriptionJob']['TranscriptionJobStatus'] in ['COMPLETED',
 'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
print(status)
```

Kode contoh berikut menunjukkan hasil transkripsi dari pekerjaan transkripsi dengan partisi speaker diaktifkan.

```
{
    "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

Untuk mentranskripsikan file audio percakapan antara dokter yang mempraktikkan perawatan primer dan pasien ()AWS CLI

- Jalankan kode berikut.

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

Kode berikut menunjukkan isi dari`example-start-command.json`.

```
{
    "MedicalTranscriptionJobName": "my-first-med-transcription-job",
     "Media": {
          "MediaFileUri": "s3://amzn-s3-demo-bucket/my-input-files/my-audio-
file.flac"
      },
      "OutputBucketName": "amzn-s3-demo-bucket",
      "OutputKey": "my-output-files/",
      "LanguageCode": "en-US",
```

```
      "Specialty": "PRIMARYCARE",
      "Type": "CONVERSATION",
      "Settings":{
          "ShowSpeakerLabels": true,
          "MaxSpeakerLabels": 2
        }
}
```

Mengaktifkan partisi speaker dalam streaming waktu nyata

Untuk mempartisi speaker dan memberi label pidato mereka dalam aliran waktu nyata, gunakan AWS Management Console atau permintaan streaming. Partisi speaker bekerja paling baik untuk antara dua dan lima speaker dalam satu aliran. Meskipun Amazon Transcribe Medical dapat mempartisi lebih dari lima speaker dalam satu aliran, akurasi partisi berkurang jika Anda melebihi angka itu.

Untuk memulai permintaan HTTP/2, gunakan API. `StartMedicalStreamTranscription` Untuk memulai WebSocket permintaan, gunakan URI yang telah ditandatangani sebelumnya. URI berisi informasi yang diperlukan untuk mengatur komunikasi dua arah antara aplikasi Anda dan Medis. Amazon Transcribe

Mengaktifkan partisi speaker dalam audio yang diucapkan ke mikrofon Anda ()AWS Management Console

Anda dapat menggunakannya AWS Management Console untuk memulai aliran real-time dari percakapan klinis-pasien, atau dikte yang diucapkan ke mikrofon Anda secara real-time.

1. Masuk ke AWS Management Console.
2. Di panel navigasi, untuk Amazon Transcribe Medis pilih Transkripsi waktu nyata.
3. Untuk jenis input Audio, pilih jenis pidato medis yang ingin Anda transkripsikan.
4. Untuk Pengaturan tambahan, pilih Partisi speaker.
5. Pilih Mulai streaming untuk mulai mentranskripsikan audio waktu nyata Anda.
6. Bicaralah ke mikrofon.

Mengaktifkan partisi speaker dalam aliran HTTP/2

Untuk mengaktifkan partisi speaker dalam aliran HTTP/2 percakapan medis, gunakan `StartMedicalStreamTranscription`API dan tentukan yang berikut ini:

- Untuk`LanguageCode`, tentukan kode bahasa yang sesuai dengan bahasa dalam aliran. Nilai yang valid adalah`en-US`.
- Untuk`MediaSampleHertz`, tentukan laju sampel audio.
- Untuk`Specialty`, tentukan spesialisasi medis penyedia.
- `ShowSpeakerLabel` – `true`

Untuk informasi lebih lanjut tentang menyiapkan aliran HTTP/2 untuk mentranskripsikan percakapan medis, lihat. Menyiapkan aliran HTTP/2

Mengaktifkan partisi speaker dalam permintaan WebSocket

Untuk mempartisi speaker dalam WebSocket aliran dengan API, gunakan format berikut untuk membuat URI yang telah ditandatangani sebelumnya untuk memulai WebSocket permintaan dan disetel `show-speaker-label` ke. `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
```

Kode berikut menunjukkan contoh respons terpotong dari permintaan streaming.

```
{
  "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": "XXXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXX",
        "StartTime": 18.74
      }
    ]
  }
}
```

Amazon Transcribe Medis memecah aliran audio masuk Anda berdasarkan segmen ucapan alami, seperti perubahan speaker atau jeda dalam audio. Transkripsi dikembalikan secara progresif ke aplikasi Anda, dengan setiap respons berisi lebih banyak ucapan yang ditranskripsi hingga seluruh segmen ditranskripsi. Kode sebelumnya adalah contoh terpotong dari segmen ucapan yang ditranskripsi sepenuhnya. Label speaker hanya muncul untuk segmen yang sepenuhnya ditranskripsi.

Daftar berikut menunjukkan organisasi objek dan parameter dalam output transkripsi streaming.

`Transcript`

Setiap segmen pidato memiliki `Transcript` objeknya sendiri.

`Results`

Setiap `Transcript` objek memiliki objeknya sendiri`Results`. Objek ini berisi `isPartial` bidang. Ketika nilainya`false`, hasil yang dikembalikan adalah untuk seluruh segmen pidato.

`Alternatives`

Setiap `Results` objek memiliki `Alternatives` objek.

`Items`

Setiap `Alternatives` objek memiliki `Items` objeknya sendiri yang berisi informasi tentang setiap kata dan tanda baca dalam keluaran transkripsi. Saat Anda mengaktifkan partisi speaker, setiap kata memiliki `Speaker` label untuk segmen ucapan yang ditranskripsi sepenuhnya. Amazon Transcribe Medis menggunakan label ini untuk menetapkan bilangan bulat unik untuk setiap pembicara dalam aliran. `Type`Parameter yang memiliki nilai `speaker-change` menunjukkan bahwa satu orang telah berhenti berbicara dan orang lain akan segera memulai.

`Transcript`

Setiap objek Item berisi segmen ucapan yang ditranskripsi sebagai nilai `Transcript` bidang.

Untuk informasi selengkapnya tentang WebSocket permintaan, lihatMenyiapkan WebSocket aliran.

Mentranskripsikan audio multi-saluran

Jika Anda memiliki file audio atau streaming yang memiliki beberapa saluran, Anda dapat menggunakan identifikasi saluran untuk mentranskripsikan ucapan dari masing-masing saluran tersebut. Amazon Transcribe Medis mentranskripsikan pidato dari setiap saluran secara terpisah. Ini menggabungkan transkripsi terpisah dari setiap saluran menjadi satu output transkripsi.

Gunakan identifikasi saluran untuk mengidentifikasi saluran terpisah dalam audio Anda dan transkripsikan ucapan dari masing-masing saluran tersebut. Aktifkan ini dalam situasi seperti skenario penelepon dan agen. Gunakan ini untuk membedakan penelepon dari agen dalam rekaman atau aliran dari pusat kontak yang melakukan pemantauan keamanan obat.

Anda dapat mengaktifkan identifikasi saluran untuk pemrosesan batch dan streaming waktu nyata. Daftar berikut menjelaskan cara mengaktifkannya untuk setiap metode.

- Transkripsi Batch — AWS Management Console dan API `StartMedicalTranscriptionJob`
- Transkripsi streaming — WebSocket streaming dan API `StartMedicalStreamTranscription`

Mentranskripsikan file audio multi-saluran

Saat Anda mentranskripsikan file audio, Amazon Transcribe Medical mengembalikan daftar item untuk setiap saluran. Item adalah kata yang ditranskripsikan atau tanda baca. Setiap kata memiliki waktu mulai dan waktu akhir. Jika seseorang di satu saluran berbicara melalui seseorang di saluran terpisah, waktu mulai dan waktu akhir item untuk setiap saluran tumpang tindih saat individu berbicara satu sama lain.

Secara default, Anda dapat menyalin file audio dengan dua saluran. Anda dapat meminta peningkatan kuota jika perlu menyalin file yang memiliki lebih dari dua saluran. Untuk informasi tentang meminta kenaikan kuota, lihat Layanan AWS kuota.

Untuk mentranskripsikan audio multi-saluran dalam pekerjaan transkripsi batch, gunakan AWS Management Console atau API. `StartMedicalTranscriptionJob`

AWS Management Console

Untuk menggunakan AWS Management Console untuk mengaktifkan identifikasi saluran dalam pekerjaan transkripsi batch Anda, Anda mengaktifkan identifikasi audio dan kemudian identifikasi saluran. Identifikasi saluran adalah bagian dari identifikasi audio di. AWS Management Console

1. Masuk ke AWS Management Console.
2. Di panel navigasi, di bawah Amazon Transcribe Medis, pilih Pekerjaan transkripsi.
3. Pilih Buat tugas.
4. Pada halaman Tentukan detail pekerjaan, berikan informasi tentang pekerjaan transkripsi Anda.
5. Pilih Berikutnya.
6. Aktifkan identifikasi Audio.

7. Untuk jenis identifikasi Audio, pilih Identifikasi saluran.
8. Pilih Buat.

API

Untuk mentranskripsikan file audio multi-saluran (API)

- Untuk `StartMedicalTranscriptionJob` API, tentukan yang berikut ini.
  a. Untuk `TranscriptionJobName`, tentukan nama yang unik untuk Anda Akun AWS.
  b. Untuk `LanguageCode`, tentukan kode bahasa yang sesuai dengan bahasa yang digunakan dalam file audio. Nilai yang valid adalah `en-US`.
  c. Untuk `MediaFileUri` parameter `Media` objek, tentukan nama file media yang ingin Anda transkripsikan.
  d. Untuk `Settings` objek, atur `ChannelIdentification` ke `true`.

Berikut ini adalah contoh permintaan menggunakan AWS SDK untuk Python (Boto3).

```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
job_name = "my-first-transcription-job"
job_name = "my-first-med-transcription-job"
job_uri = "s3://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
transcribe.start_medical_transcription_job(
      MedicalTranscriptionJobName = job_name,
      Media = {
        'MediaFileUri': job_uri
      },
      OutputBucketName = 'amzn-s3-demo-bucket',
      OutputKey = 'output-files/',
      LanguageCode = 'en-US',
      Specialty = 'PRIMARYCARE',
      Type = 'CONVERSATION',
      Settings = {
        'ChannelIdentification': True
      }
)
```

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

AWS CLI

Untuk mentranskripsikan file audio multi-saluran menggunakan pekerjaan transkripsi batch ()AWS CLI

- Jalankan kode berikut.

```
aws transcribe start-medical-transcription-job \
--region us-west-2 \
--cli-input-json file://example-start-command.json
```

Berikut ini adalah kode dari`example-start-command.json`.

```
{
      "MedicalTranscriptionJobName": "my-first-med-transcription-job",
      "Media": {
          "MediaFileUri": "s3://amzn-s3-demo-bucket/my-input-files/my-audio-
file.flac"
      },
      "OutputBucketName": "amzn-s3-demo-bucket",
      "OutputKey": "my-output-files/",
      "LanguageCode": "en-US",
      "Specialty": "PRIMARYCARE",
      "Type": "CONVERSATION",

        "Settings":{
          "ChannelIdentification": true
        }
}
```

Kode berikut menunjukkan output transkripsi untuk file audio yang memiliki percakapan di dua saluran.


```
{
  "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"
        },
        ...
    ]
  }
}
```

Mentranskripsikan aliran audio multi-saluran

Anda dapat mentranskripsikan audio dari saluran terpisah baik di HTTP/2 atau WebSocket streaming menggunakan API. `StartMedicalStreamTranscription`

Secara default, Anda dapat mentranskripsikan aliran dengan dua saluran. Anda dapat meminta peningkatan kuota jika perlu mentranskripsikan aliran yang memiliki lebih dari dua saluran. Untuk informasi tentang meminta peningkatan kuota, lihat kuota AWS layanan.

Mentranskripsikan audio multi-saluran dalam aliran HTTP/2

Untuk mentranskripsikan audio multi-saluran dalam aliran HTTP/2, gunakan StartMedicalStreamTranscriptionAPI dan tentukan yang berikut ini:

- `LanguageCode`— Kode bahasa audio. Nilai yang valid adalah`en-US`.
- `MediaEncoding`— Pengkodean audio. Nilai yang valid adalah `ogg-opus`, `flac`, dan `pcm`.
- `EnableChannelIdentification` – `true`
- `NumberOfChannels`— jumlah saluran dalam audio streaming Anda.

Untuk informasi lebih lanjut tentang menyiapkan aliran HTTP/2 untuk mentranskripsikan percakapan medis, lihat. Menyiapkan aliran HTTP/2

Mentranskripsikan audio multi-saluran dalam aliran WebSocket

Untuk mempartisi speaker dalam WebSocket aliran, gunakan format berikut untuk membuat URI yang telah ditandatangani sebelumnya dan memulai permintaan. WebSocket Tentukan `enable-channel-identification` sebagai `true` dan jumlah saluran dalam aliran Anda di`number-of-channels`. URI yang telah ditandatangani sebelumnya berisi informasi yang diperlukan untuk mengatur komunikasi dua arah antara aplikasi Anda dan Medis. Amazon Transcribe

```
GET wss://transcribestreaming.us-west-2.amazonaws.com:8443/medical-stream-
transcription-websocket
?language-code=languageCode
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIAIOSFODNN7EXAMPLE%2F20220208%2Fus-
west-2%2Ftranscribe%2Faws4_request
&X-Amz-Date=20220208T235959Z
&X-Amz-Expires=300
&X-Amz-Security-Token=security-token
&X-Amz-Signature=Signature Version 4 signature
&X-Amz-SignedHeaders=host
&media-encoding=flac
&sample-rate=16000
&session-id=sessionId
&enable-channel-identification=true
&number-of-channels=2
```

Definisi parameter dapat ditemukan di Referensi API; parameter yang umum untuk semua operasi AWS API tercantum di bagian Parameter Umum.

Untuk informasi selengkapnya tentang WebSocket permintaan, lihatMenyiapkan WebSocket aliran.

Output streaming multi-saluran

Output dari transkripsi streaming sama untuk HTTP/2 dan permintaan. WebSocket Berikut ini adalah output contoh.

```
{
    "resultId": "XXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXX",
    "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"
}
```

Untuk setiap segmen pidato, ada `channelId` bendera yang menunjukkan saluran mana pidato itu dimiliki.

Mentranskripsikan dikte medis

Anda dapat menggunakan Amazon Transcribe Medis untuk menyalin catatan medis yang didikte dokter menggunakan pekerjaan transkripsi batch atau aliran waktu nyata. Pekerjaan transkripsi batch memungkinkan Anda untuk mentranskripsikan file audio. Anda menentukan spesialisasi medis dokter dalam pekerjaan atau aliran transkripsi Anda untuk memastikan bahwa Amazon Transcribe Medis menghasilkan hasil transkripsi dengan akurasi setinggi mungkin.

Anda dapat menuliskan dikte medis dalam spesialisasi berikut:

- Kardiologi — hanya tersedia dalam transkripsi streaming
- Neurologi — hanya tersedia dalam transkripsi streaming
- Onkologi — hanya tersedia dalam transkripsi streaming
- Perawatan Primer - termasuk jenis praktik medis berikut:
  - Obat keluarga
  - Penyakit dalam
  - Obstetri dan Ginekologi (OB-GYN)
  - Pediatri
- Radiologi — hanya tersedia dalam transkripsi streaming
- Urologi — hanya tersedia dalam transkripsi streaming

Anda dapat meningkatkan akurasi transkripsi dengan menggunakan kosakata khusus. Untuk informasi tentang cara kerja kosakata khusus medis, lihat. Meningkatkan akurasi transkripsi dengan kosakata khusus medis

Secara default, Amazon Transcribe Medical mengembalikan transkripsi dengan tingkat kepercayaan tertinggi. Jika Anda ingin mengonfigurasinya untuk mengembalikan transkripsi alternatif, lihatMenghasilkan transkripsi alternatif.

Untuk informasi tentang bagaimana angka dan pengukuran medis muncul dalam output transkripsi, lihat Mentranskripsikan angka danMentranskripsikan istilah dan pengukuran medis.

Topik

- Mentranskripsikan file audio dari dikte medis
- Mentranskripsikan dikte medis dalam aliran waktu nyata

Mentranskripsikan file audio dari dikte medis

Gunakan pekerjaan transkripsi batch untuk mentranskripsikan file audio percakapan medis. Anda dapat menggunakan ini untuk menyalin dialog klinis-pasien. Anda dapat memulai pekerjaan transkripsi batch baik di `StartMedicalTranscriptionJob`API atau. AWS Management Console

Saat Anda memulai pekerjaan transkripsi medis dengan `StartMedicalTranscriptionJob`API, Anda menentukan `PRIMARYCARE` sebagai nilai `Specialty` parameter.

AWS Management Console

Untuk mentranskripsikan dialog klinis-pasien ()AWS Management Console

Untuk menggunakan untuk AWS Management Console mentranskripsikan dialog klinis-pasien, buat pekerjaan transkripsi dan pilih Percakapan untuk jenis input Audio.

1. Masuk ke AWS Management Console.
2. Di panel navigasi, di bawah Amazon Transcribe Medis, pilih Pekerjaan transkripsi.
3. Pilih Buat tugas.
4. Pada halaman Tentukan detail pekerjaan, di bawah Pengaturan Job, tentukan yang berikut ini.
    a. Nama — nama pekerjaan transkripsi.
    b. Jenis input audio - Dikte

5. Untuk bidang yang tersisa, tentukan Amazon S3 lokasi file audio Anda dan di mana Anda ingin menyimpan output dari pekerjaan transkripsi Anda.
6. Pilih Berikutnya.
7. Pilih Buat.

API

Untuk mentranskripsikan percakapan medis menggunakan pekerjaan transkripsi batch (API)

- Untuk `StartMedicalTranscriptionJob`API, tentukan yang berikut ini.

  a. Untuk`MedicalTranscriptionJobName`, tentukan nama unik di Anda Akun AWS.

  b. Untuk`LanguageCode`, tentukan kode bahasa yang sesuai dengan bahasa yang digunakan dalam file audio Anda dan bahasa filter kosakata Anda.

  c. Dalam `MediaFileUri` parameter `Media` objek, tentukan nama file audio yang ingin Anda transkripsikan.

  d. Untuk`Specialty`, tentukan spesialisasi medis dokter yang berbicara dalam file audio.

  e. Untuk `Type`, tentukan `DICTATION`.

  f. Untuk`OutputBucketName`, tentukan Amazon S3 ember untuk menyimpan hasil transkripsi.

  Berikut ini adalah contoh permintaan yang menggunakan AWS SDK untuk Python (Boto3) untuk mentranskripsikan dikte medis dari seorang dokter di bidang spesialisasi. `PRIMARYCARE`

```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe')
job_name = "my-first-med-transcription-job"
job_uri = "s3://amzn-s3-demo-bucket/my-input-files/my-audio-file.flac"
transcribe.start_medical_transcription_job(
    MedicalTranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'amzn-s3-demo-bucket',
    OutputKey = 'my-output-files/',
    LanguageCode = 'en-US',
```

```
     Specialty = 'PRIMARYCARE',
     Type = 'DICTATION'
 )
while True:
    status = transcribe.get_medical_transcription_job(MedicalTranscriptionJobName =
 job_name)
    if status['MedicalTranscriptionJob']['TranscriptionJobStatus'] in ['COMPLETED',
 'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
print(status)
```

Kode contoh berikut menunjukkan hasil transkripsi dikte medis.

```
{
    "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

Untuk mengaktifkan partisi speaker dalam pekerjaan transkripsi batch ()AWS CLI

- Jalankan kode berikut.

```
aws transcribe start-medical-transcription-job \
--region us-west-2 \
--cli-input-json file://example-start-command.json
```

Kode berikut menunjukkan isi dari`example-start-command.json`.

```
{
      "MedicalTranscriptionJobName": "my-first-med-transcription-job",
      "Media": {
      "MediaFileUri": "s3://amzn-s3-demo-bucket/my-input-files/my-audio-file.flac"
      },
      "OutputBucketName": "amzn-s3-demo-bucket",
      "OutputKey": "my-output-files/",
      "LanguageCode": "en-US",
      "Specialty": "PRIMARYCARE",
      "Type": "DICTATION"
}
```

Mentranskripsikan dikte medis dalam aliran waktu nyata

Gunakan WebSocket aliran untuk mentranskripsikan dikte medis sebagai aliran audio. Anda juga dapat menggunakan AWS Management Console untuk menyalin ucapan yang Anda atau orang lain berbicara langsung ke mikrofon.

Untuk HTTP/2 atau WebSocket streaming, Anda dapat mentranskripsikan audio dalam spesialisasi medis berikut:

- Kardiologi
- Onkologi
- Neurologi
- Perawatan Primer
- Radiologi
- Urologi

Setiap spesialisasi medis mencakup banyak jenis prosedur dan janji temu. Oleh karena itu, dokter mendikte berbagai jenis catatan. Gunakan contoh berikut sebagai panduan untuk membantu Anda menentukan nilai parameter `specialty` URI WebSocket permintaan, atau `Specialty` parameter `StartMedicalStreamTranscription`API:

- Untuk dikte setelah prosedur elektrofisiologi atau ekokardiogram, pilih. `CARDIOLOGY`
- Untuk dikte setelah onkologi bedah atau prosedur onkologi radiasi, pilih. `ONCOLOGY`
- Untuk dokter yang mendikte catatan yang menunjukkan diagnosis ensefalitis, pilih. `NEUROLOGY`
- Untuk dikte catatan prosedur untuk memecah batu kandung kemih, pilih`UROLOGY`.
- Untuk dikte catatan dokter setelah konsultasi penyakit dalam, pilih. `PRIMARYCARE`
- Untuk dikte dokter yang mengkomunikasikan temuan CT scan, PET scan, MRI, atau radiografi, pilih. `RADIOLOGY`
- Untuk dikte catatan dokter setelah konsultasi ginekologi, pilih. `PRIMARYCARE`

Untuk meningkatkan akurasi transkripsi istilah tertentu dalam aliran waktu nyata, gunakan kosakata khusus. Untuk mengaktifkan kosakata kustom, tetapkan nilai `vocabulary-name` ke nama kosakata kustom yang ingin Anda gunakan.

Mentranskripsikan dikte yang diucapkan ke mikrofon Anda dengan AWS Management Console

Untuk menggunakan AWS Management Console untuk mentranskripsikan audio streaming dari dikte medis, pilih opsi untuk menyalin dikte medis, memulai aliran, dan mulai berbicara ke mikrofon.

Untuk mentranskripsikan audio streaming dari dikte medis ()AWS Management Console

1. Masuk ke AWS Management Console.
2. Di panel navigasi, di bawah Amazon Transcribe Medis, pilih Transkripsi waktu nyata.
3. Pilih Dikte.
4. Untuk spesialisasi medis, pilih spesialisasi medis dari dokter yang berbicara di sungai.
5. Pilih Mulai streaming.
6. Bicaralah ke mikrofon.

Mentranskripsikan dikte dalam aliran HTTP/2

Untuk mentranskripsikan aliran HTTP/2 dari dikte medis, gunakan `StartMedicalStreamTranscription`API dan tentukan yang berikut ini:

- `LanguageCode`- Kode bahasa. Nilai yang valid adalah `en-US`
- `MediaEncoding`— Pengkodean yang digunakan untuk audio input. Nilai yang valid adalah `pcm`, `ogg-opus`, dan `flac`.
- `Specialty`- Keistimewaan profesional medis.
- `Type` – `DICTATION`

Untuk informasi lebih lanjut tentang menyiapkan aliran HTTP/2 untuk mentranskripsikan dikte medis, lihat. Menyiapkan aliran HTTP/2

Menggunakan permintaan WebSocket streaming untuk menyalin dikte medis

Untuk mentranskripsikan dikte medis dalam aliran waktu nyata menggunakan WebSocket permintaan, Anda membuat URI yang telah ditetapkan sebelumnya. URI ini berisi informasi yang diperlukan untuk mengatur aliran audio antara aplikasi Anda dan Amazon Transcribe Medis. Untuk informasi selengkapnya tentang membuat WebSocket permintaan, lihatMenyiapkan WebSocket aliran.

Gunakan template berikut untuk membuat URI presigned Anda.

```
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
```

Untuk informasi selengkapnya tentang membuat pra-ditandatangani URIs, lihatMenyiapkan WebSocket aliran.

Meningkatkan akurasi transkripsi dengan kosakata khusus medis

Untuk meningkatkan akurasi transkripsi dalam Amazon Transcribe Medis, buat dan gunakan satu atau lebih kosakata khusus medis. Kosakata khusus adalah kumpulan kata atau frasa yang spesifik domain. Koleksi ini membantu meningkatkan kinerja Amazon Transcribe Medis dalam menyalin kata-kata atau frasa tersebut.

Anda bertanggung jawab atas integritas data Anda sendiri ketika Anda menggunakan Amazon Transcribe Medis. Jangan memasukkan informasi rahasia, informasi pribadi (PII), atau informasi kesehatan yang dilindungi (PHI), ke dalam kosakata khusus.

Untuk hasil terbaik, buat kosakata kustom kecil terpisah yang masing-masing membantu menyalin rekaman audio tertentu. Anda menerima peningkatan yang lebih besar dalam akurasi transkripsi daripada jika Anda membuat satu kosakata khusus yang besar untuk digunakan dengan semua rekaman Anda.

Secara default, Anda dapat memiliki hingga 100 kosakata khusus di Anda. Akun AWS Kosakata khusus tidak boleh melebihi 50 KB dalam ukuran. Untuk informasi tentang meminta peningkatan jumlah kosakata khusus yang dapat Anda miliki di Anda Akun AWS, lihat AWS kuota layanan.

Kosakata khusus tersedia dalam bahasa Inggris AS (en-US).

Topik

- Membuat file teks untuk kosakata khusus medis Anda
- Menggunakan file teks untuk membuat kosakata khusus medis
- Mentranskripsikan file audio menggunakan kosakata khusus medis
- Mentranskripsikan aliran waktu nyata menggunakan kosakata khusus medis
- Set karakter untuk Amazon Transcribe Medis

Membuat file teks untuk kosakata khusus medis Anda

Untuk membuat kosakata khusus, Anda membuat file teks yang dalam format UTF-8. Dalam file ini, Anda membuat tabel empat kolom, dengan setiap kolom menentukan bidang. Setiap bidang memberi tahu Amazon Transcribe Medis bagaimana istilah khusus domain diucapkan atau bagaimana menampilkan istilah-istilah ini dalam transkripsi Anda. Anda menyimpan file teks yang berisi bidang ini dalam Amazon S3 ember.

Memahami cara memformat file teks Anda

Untuk membuat kosakata khusus medis, Anda memasukkan nama kolom sebagai baris header. Anda memasukkan nilai untuk setiap kolom di bawah baris header.

Berikut ini adalah nama-nama dari empat kolom tabel:

- `Phrase`- kolom diperlukan, nilai yang diperlukan
- `IPA`- kolom diperlukan, nilai bisa opsional
- `SoundsLike`- kolom diperlukan, nilai bisa opsional
- `DisplayAs`- kolom diperlukan, nilai bisa opsional

Saat Anda membuat kosakata khusus, pastikan Anda:

- Pisahkan setiap kolom dengan satu karakter Tab. Amazon Transcribe melempar pesan kesalahan jika Anda mencoba memisahkan kolom dengan spasi atau beberapa karakter Tab.
- Pastikan tidak ada spasi tambahan atau spasi putih setelah setiap nilai dalam kolom.

Pastikan nilai yang Anda masukkan untuk setiap kolom:

- Memiliki kurang dari 256 karakter, termasuk tanda hubung
- Gunakan hanya karakter dari set karakter yang diizinkan, lihatSet karakter untuk Amazon Transcribe Medis.

Memasukkan nilai untuk kolom tabel

Informasi berikut menunjukkan cara menentukan nilai untuk empat kolom tabel:

- **`Phrase`**— Kata atau frasa yang harus dikenali. Anda harus memasukkan nilai di kolom ini.

  Jika entri adalah frasa, pisahkan kata-kata dengan tanda hubung (-). Misalnya, masukkan **`cerebral autosomal dominant arteriopathy with subcortical infarcts and leukoencephalopathy`** sebagai**`cerebral-autosomal-dominant-arteriopathy-with-subcortical-infarcts-and-leukoencephalopathy`**.

  Masukkan akronim atau kata lain yang hurufnya harus diucapkan satu per satu sebagai huruf tunggal diikuti oleh titik-titik, seperti **`D.N.A.`** atau. **`S.T.E.M.I.`** Untuk memasukkan bentuk jamak akronim, seperti "STEMIs," pisahkan “s” dari akronim dengan tanda hubung: "**`S.T.E.M.I-s`**" Anda dapat menggunakan huruf besar atau kecil untuk akronim.

  `Phrase`Kolom diperlukan. Anda dapat menggunakan salah satu karakter yang diizinkan untuk bahasa input. Untuk karakter yang diizinkan, lihatSet karakter untuk Amazon Transcribe Medis. Jika Anda tidak menentukan `DisplayAs` kolom, Amazon Transcribe Medis menggunakan isi `Phrase` kolom dalam file output.
- **`IPA`**(kolom diperlukan, nilai bisa opsional) - Untuk menentukan pengucapan kata atau frasa, Anda dapat menyertakan karakter dalam Alfabet Fonetik Internasional (IPA) di kolom ini. `IPA`Kolom tidak dapat berisi spasi depan atau belakang, dan Anda harus menggunakan satu spasi untuk memisahkan setiap fonem dalam input. Misalnya, dalam bahasa Inggris Anda akan memasukkan frasa **`acute-respiratory-distress-syndrome`** sebagai**`ə k j u t # # s p # # ə t # # i d # s t # # s s # n d # o# m`**. Anda akan memasukkan frasa **`A.L.L.`** sebagai**`e# # l # l`**.

  Bahkan jika Anda tidak menentukan isi `IPA` kolom, Anda harus menyertakan `IPA` kolom kosong. Jika Anda menyertakan nilai dalam `IPA` kolom, Anda tidak dapat memberikan nilai untuk `SoundsLike` kolom.

Untuk daftar karakter IPA yang diizinkan untuk bahasa tertentu, lihatSet karakter untuk Amazon Transcribe Medis. Bahasa Inggris AS adalah satu-satunya bahasa yang tersedia di bidang Amazon Transcribe medis.

- **`SoundsLike`**(kolom diperlukan, nilai bisa opsional) - Anda dapat memecah kata atau frasa menjadi segmen yang lebih kecil dan memberikan pengucapan untuk setiap segmen menggunakan ortografi standar bahasa untuk meniru cara kata itu terdengar. Misalnya, Anda dapat memberikan petunjuk pengucapan untuk frasa **`cerebral-autosomal-dominant-arteriopathy-with-subcortical-infarcts-and-leukoencephalopathy`** seperti ini:**`sir-e-brul-aut-o-som-ul-dah-mi-nant-ar-ter-ri-o-pa-thy-with-sub-cor-ti-cul-in-farcts-and-lewk-o-en-ce-phul-ah-pu-thy`**. Petunjuk untuk frasa tersebut **`atrioventricular-nodal-reentrant-tachycardia`** akan terlihat seperti ini:**`ay-tree-o-ven-trick-u-lar-node-al-re-entr-ant-tack-ih-card-ia`**. Anda memisahkan setiap bagian dari petunjuk dengan tanda hubung (-).

  Bahkan jika Anda tidak memberikan nilai untuk `SoundsLike` kolom, Anda harus menyertakan `SoundsLike` kolom kosong. Jika Anda menyertakan nilai dalam `SoundsLike` kolom, Anda tidak dapat memberikan nilai untuk `IPA` kolom.

  Anda dapat menggunakan salah satu karakter yang diizinkan untuk bahasa input. Untuk daftar karakter yang diizinkan, lihatSet karakter untuk Amazon Transcribe Medis.

- **`DisplayAs`**(kolom diperlukan, nilai bisa opsional) - Mendefinisikan bagaimana kata atau frase terlihat ketika itu output. Misalnya, jika kata atau frasa itu**`cerebral-autosomal-dominant-arteriopathy-with-subcortical-infarcts-and-leukoencephalopathy`**, Anda dapat menentukan formulir tampilan sebagai`cerebral autosomal dominant arteriopathy with subcortical infarcts and leukoencephalopathy`, sehingga tanda hubung tidak ada. Anda juga dapat menentukan `DisplayAs` `CADASIL` seolah-olah Anda ingin menampilkan akronim alih-alih istilah lengkap dalam output.

  Jika Anda tidak menentukan `DisplayAs` kolom, Amazon Transcribe Medis menggunakan `Phrase` kolom dari file input dalam output.

  Anda dapat menggunakan karakter UTF-8 apa pun di kolom. `DisplayAs`

Anda dapat menyertakan spasi hanya untuk nilai-nilai dalam `IPA` dan `DisplayAs` kolom.

Untuk membuat file teks kosakata kustom Anda, letakkan setiap kata atau frasa dalam file teks Anda pada baris terpisah. Pisahkan kolom dengan karakter Tab. Sertakan spasi hanya untuk nilai di

`DisplayAs` kolom `IPA` dan. Simpan file dengan ekstensi `.txt` dalam Amazon S3 ember di tempat yang sama di Wilayah AWS mana Anda menggunakan Amazon Transcribe Medical untuk membuat kosakata khusus Anda.

Jika Anda mengedit file teks Anda di Windows, pastikan file Anda dalam `LF` format dan tidak dalam `CRLF` format. Jika tidak, Anda tidak akan dapat membuat kosakata khusus Anda. Beberapa editor teks memungkinkan Anda mengubah format dengan perintah Find and Replace.

Contoh berikut menunjukkan teks yang dapat Anda gunakan untuk membuat kosakata khusus. Untuk membuat kosakata kustom dari contoh-contoh ini, salin contoh ke editor teks, ganti `[TAB]` dengan karakter Tab, dan unggah file teks yang disimpan ke. Amazon S3

```
Phrase[TAB]IPA[TAB]SoundsLike[TAB]DisplayAs
acute-respiratory-distress-syndrome[TAB][TAB][TAB]acute respiratory distress syndrome
A.L.L.[TAB]e# # l # l[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]
```

Anda dapat memasukkan kolom dalam urutan apa pun. Contoh berikut menunjukkan struktur valid lainnya untuk file input kosakata kustom.

```
Phrase[TAB]SoundsLike[TAB]IPA[TAB]DisplayAs
acute-respiratory-distress-syndrome[TAB][TAB][TAB]acute respiratory distress syndrome
A.L.L.[TAB][TAB]e# # l # l[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# # l # l[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
```

Untuk kemudahan membaca, tabel berikut menunjukkan contoh sebelumnya dengan lebih jelas dalam format html. Mereka dimaksudkan hanya untuk menggambarkan contoh-contoh.

Frase	IPA	SoundsLike	DisplayAs
acute-respiratory-distress-syndrome			sindrom gangguan pernapasan akut
A.L.L.	el ɛl		SEMUA
atrioventricular-nodal-reentrant-tachycardia		ay-tree-o-ven-trick-u-lar-node-al-re-entr-ant-tack-ih-card-ia	

Frase	SoundsLike	IPA	DisplayAs
acute-respiratory-distress-syndrome			sindrom gangguan pernapasan akut
atrioventricular-nodal-reentrant-tachycardia	ay-tree-o-ven-trick-u-lar-node-al-re-entr-ant-tack-ih-card-ia		
A.L.L.		el ɛl	SEMUA

DisplayAs	SoundsLike	IPA	Frase
sindrom gangguan pernapasan akut			acute-respiratory-distress-syndrome
SEMUA		el ɛl	A.L.L.
	ay-tree-o-ven-trick-u-lar-node-al-re-entr-ant-tack-ih-card-ia		atrioventricular-nodal-reentrant-tachycardia

Menggunakan file teks untuk membuat kosakata khusus medis

Untuk membuat kosakata khusus, Anda harus menyiapkan file teks yang berisi koleksi kata atau frasa. Amazon Transcribe Medis menggunakan file teks ini untuk membuat kosakata khusus yang dapat Anda gunakan untuk meningkatkan akurasi transkripsi kata atau frasa tersebut. Anda dapat membuat kosakata khusus menggunakan `CreateMedicalVocabulary`API atau konsol Amazon Transcribe Medis.

AWS Management Console

Untuk menggunakan AWS Management Console untuk membuat kosakata khusus, Anda memberikan Amazon S3 URI file teks yang berisi kata atau frasa Anda.

1. Masuk ke AWS Management Console.
2. Di panel navigasi, di bawah Amazon Transcribe Medis, pilih Kosakata khusus.
3. Untuk Nama, di bawah pengaturan Kosakata, pilih nama untuk kosakata kustom Anda.
4. Tentukan lokasi file audio atau file video Anda di Amazon S3:
   - Untuk lokasi file input Kosakata di S3 di bawah pengaturan Kosakata, tentukan Amazon S3 URI yang mengidentifikasi file teks yang akan Anda gunakan untuk membuat kosakata khusus Anda.
   - Untuk lokasi file input kosakata di S3, pilih Browse S3 untuk menelusuri file teks dan memilihnya.
5. Pilih Buat kosakata.

Anda dapat melihat status pemrosesan kosakata khusus Anda di. AWS Management Console

API

Untuk membuat kosakata kustom medis (API)

- Untuk `StartTranscriptionJob`API, tentukan yang berikut ini.
  a. Untuk `LanguageCode`, tentukan `en-US`.
  b. Untuk`VocabularyFileUri`, tentukan Amazon S3 lokasi file teks yang Anda gunakan untuk menentukan kosakata kustom Anda.
  c. Untuk`VocabularyName`, tentukan nama untuk kosakata kustom Anda. Nama yang Anda tentukan harus unik di dalam nama Anda Akun AWS.

Untuk melihat status pemrosesan kosakata kustom Anda, gunakan API. GetMedicalVocabulary

Berikut ini adalah contoh permintaan menggunakan AWS SDK untuk Python (Boto3) untuk membuat kosakata kustom.

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

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

AWS CLI

Untuk mengaktifkan partisi speaker dalam pekerjaan transkripsi batch ()AWS CLI

- Jalankan kode berikut.

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

Mentranskripsikan file audio menggunakan kosakata khusus medis

Gunakan `StartMedicalTranscriptionJob`or AWS Management Console untuk memulai pekerjaan transkripsi yang menggunakan kosakata khusus untuk meningkatkan akurasi transkripsi.

AWS Management Console

1. Masuk ke AWS Management Console.
2. Di panel navigasi, di bawah Amazon Transcribe Medis, pilih Pekerjaan transkripsi.
3. Pilih Buat tugas.
4. Pada halaman Tentukan detail pekerjaan, berikan informasi tentang pekerjaan transkripsi Anda.
5. Pilih Berikutnya.
6. Di bawah Kustomisasi, aktifkan kosakata khusus.
7. Di bawah Pemilihan kosakata, pilih kosakata khusus.
8. Pilih Buat.

API

Untuk mengaktifkan partisi speaker dalam file audio menggunakan tugas transkripsi batch (API)

- Untuk `StartMedicalTranscriptionJob`API, tentukan yang berikut ini.
    a. Untuk`MedicalTranscriptionJobName`, tentukan nama yang unik di Anda Akun AWS.
    b. Untuk`LanguageCode`, tentukan kode bahasa yang sesuai dengan bahasa yang digunakan dalam file audio Anda dan bahasa filter kosakata Anda.
    c. Untuk `MediaFileUri` parameter `Media` objek, tentukan nama file audio yang ingin Anda transkripsikan.
    d. Untuk`Specialty`, tentukan spesialisasi medis dokter yang berbicara dalam file audio.
    e. Untuk`Type`, tentukan apakah file audio adalah percakapan atau dikte.
    f. Untuk`OutputBucketName`, tentukan Amazon S3 ember untuk menyimpan hasil transkripsi.
    g. Untuk `Settings` objek, tentukan yang berikut ini.
        - `VocabularyName`— nama kosakata kustom Anda.

Permintaan berikut menggunakan AWS SDK untuk Python (Boto3) untuk memulai pekerjaan transkripsi batch dengan kosakata khusus.

```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
job_name = "my-first-med-transcription-job"
job_uri = "s3://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
transcribe.start_medical_transcription_job(
   MedicalTranscriptionJobName = job_name,
   Media = {
       'MediaFileUri': job_uri
   },
   OutputBucketName = 'amzn-s3-demo-bucket',
   OutputKey = 'my-output-files/',
   LanguageCode = 'en-US',
   Specialty = 'PRIMARYCARE',
   Type = 'CONVERSATION',
   Settings = {
       'VocabularyName': 'example-med-custom-vocab'
       }
 )

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

Mentranskripsikan aliran waktu nyata menggunakan kosakata khusus medis

Untuk meningkatkan akurasi transkripsi dalam aliran waktu nyata, Anda dapat menggunakan kosakata khusus menggunakan HTTP/2 atau aliran. WebSocket Untuk memulai permintaan HTTP/2, gunakan API. StartMedicalStreamTranscription Anda dapat menggunakan kosakata khusus

secara real-time menggunakan StartMedicalStreamTranscriptionAPI AWS Management Console, atau dengan menggunakan protokol. WebSocket

Mentranskripsikan dikte yang diucapkan ke Mikrofon Anda ()AWS Management Console

Untuk menggunakan AWS Management Console untuk mentranskripsikan audio streaming dari dikte medis, pilih opsi untuk menyalin dikte medis, memulai aliran, dan mulai berbicara ke mikrofon.

Untuk mentranskripsikan audio streaming dari dikte medis ()AWS Management Console

1. Masuk ke AWS Management Console.
2. Di panel navigasi, di bawah Amazon Transcribe Medis, pilih Transkripsi waktu nyata.
3. Untuk spesialisasi medis, pilih spesialisasi medis dari dokter yang berbicara di sungai.
4. Untuk jenis input Audio, pilih Percakapan atau Dikte.
5. Untuk Pengaturan tambahan, pilih Kosakata khusus.
   - Untuk pemilihan Kosakata, pilih kosakata khusus.
6. Pilih Mulai streaming.
7. Bicaralah ke mikrofon.

Mengaktifkan partisi speaker dalam aliran HTTP/2

Berikut ini adalah sintaks untuk parameter permintaan HTTP/2.

```
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
```

Deskripsi parameter:

- host: Perbarui Wilayah AWS ('us-west-2' pada contoh sebelumnya) dengan panggilan Anda. Wilayah AWS Untuk daftar yang valid Wilayah AWS, lihat Wilayah AWS dan Endpoints.
- otorisasi: Ini adalah bidang yang dihasilkan. Untuk mempelajari selengkapnya tentang membuat tanda tangan, lihat Menandatangani AWS permintaan dengan Tanda Tangan Versi 4.
- x-amz-target: Jangan mengubah bidang ini; gunakan konten yang ditunjukkan pada contoh sebelumnya.
- x-amz-content-sha256: Ini adalah bidang yang dihasilkan. Untuk mempelajari lebih lanjut tentang menghitung tanda tangan, lihat Menandatangani AWS permintaan dengan Tanda Tangan Versi 4.
- x-amz-date: Tanggal dan waktu tanda tangan dibuat. Formatnya adalah YYYYMMDDTHHMMSSZ, di mana YYYYY=year, mm=month, dd=day, hh=hour, mm=minute, ss=seconds, dan 'T' dan 'Z' adalah karakter tetap. Untuk informasi selengkapnya, lihat Menangani Tanggal di Tanda Tangan Versi 4.
- x-amzn-transcribe-session-id: Nama untuk sesi streaming Anda.
- x-amzn-transcribe-language-code: Pengkodean yang digunakan untuk audio input Anda. Lihat `StartMedicalStreamTranscription`atau Bahasa yang didukung dan fitur khusus bahasa untuk daftar nilai yang valid.
- x-amzn-transcribe-media-encoding: Pengkodean yang digunakan untuk audio input Anda. Nilai yang valid adalah `pcm`, `ogg-opus`, dan `flac`.
- x-amzn-transcribe-sample-rate: Laju sampel audio input (dalam Hertz). Amazon Transcribe mendukung rentang dari 8.000 Hz hingga 48.000 Hz. Audio berkualitas rendah, seperti audio telepon, biasanya sekitar 8.000 Hz. Audio berkualitas tinggi biasanya berkisar dari 16.000 Hz hingga 48.000 Hz. Perhatikan bahwa laju sampel yang Anda tentukan harus sesuai dengan audio Anda.
- x-amzn-transcribe-vocabulary-name: Nama kosakata yang ingin Anda gunakan dengan transkripsi Anda.
- x-amzn-transcribe-specialty: Spesialisasi medis sedang ditranskripsikan.
- x-amzn-transcribe-type: Pilih apakah ini dikte atau percakapan.
- x-amzn-transcribe-show-speaker-label: Untuk mengaktifkan diarisasi, nilai ini harus. `true`
- content-type: Jangan mengubah bidang ini; gunakan konten yang ditampilkan pada contoh sebelumnya.

Mengaktifkan partisi speaker dalam permintaan WebSocket

Untuk mempartisi speaker dalam WebSocket aliran dengan API, gunakan format berikut untuk membuat URI yang telah ditandatangani sebelumnya untuk memulai WebSocket permintaan dan menyetel `vocabulary-name` ke nama kosakata kustom.

```
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
```

Set karakter untuk Amazon Transcribe Medis

Untuk menggunakan kosakata khusus di Amazon Transcribe Medical, gunakan set karakter berikut.

Set karakter bahasa Inggris

Untuk kosakata kustom bahasa Inggris, Anda dapat menggunakan karakter berikut di kolom `Phrase` dan`SoundsLike`:

- a - z
- A - Z
- '(apostrof)
- - (tanda hubung)
- . (periode)

Anda dapat menggunakan karakter Alfabet Fonetik Internasional (IPA) berikut di `IPA` kolom file input kosakata.

Karakter	Kode	Karakter	Kode
a	0061 028A	w	0077
aɪ	0061 026A	z	007A
b	0062	æ	00E6
d	0064	ð	00F0
eɪ	0065 026A	ŋ	014B
f	0066	ɑ	0251
g	0067	ɔ	0254
-h	0068	ɔɪ	0254 026A
saya	0069	ə	0259
j	006A	ɛ	025B
k	006B	ɝ	025D
l	006C	g	0261
l	006C 0329	ɪ	026A
m	006D	ɹ	0279
n	006E	ʃ	0283
n	006E 0329	ʊ	028A
o	006F 028A	ʌ	028C
p	0070	ʍ	028D
detik	0073	ʒ	0292

Karakter	Kode	Karakter	Kode
t	0074	ʤ	02A4
u	0075	ʧ	02A7
v	0076	θ	03B8

Mengidentifikasi informasi kesehatan pribadi (PHI) dalam transkripsi

Gunakan Identifikasi Informasi Kesehatan Pribadi untuk memberi label informasi kesehatan pribadi (PHI) dalam hasil transkripsi Anda. Dengan meninjau label, Anda dapat menemukan PHI yang dapat digunakan untuk mengidentifikasi pasien.

Anda dapat mengidentifikasi PHI menggunakan streaming waktu nyata atau pekerjaan transkripsi batch.

Anda dapat menggunakan pasca-pemrosesan Anda sendiri untuk menyunting PHI yang diidentifikasi dalam output transkripsi.

Gunakan Identifikasi Informasi Kesehatan Pribadi untuk mengidentifikasi jenis-jenis PHI berikut:

- PHI pribadi:
  - Nama - Nama lengkap atau nama belakang dan inisiasi
  - Gender
  - Umur
  - Nomor telepon
  - Tanggal (tidak termasuk tahun) yang berhubungan langsung dengan pasien
  - Alamat email
- PHI geografis:
  - Alamat fisik
  - Kode pos
  - Nama pusat medis atau praktik
- Akun PHI:

  - Nomor faks
  - Nomor jaminan sosial (SSNs)
  - Nomor penerima asuransi kesehatan
  - Nomor akun
  - Sertifikat atau nomor lisensi
- Kendaraan PHI:
  - Nomor identifikasi Mesin (VIN)
  - Nomor plat
- PHI lainnya:
  - Lokasi Sumber Daya Seragam Web (URL)
  - Nomor alamat Internet Protocol (IP)

Amazon Transcribe Medis adalah layanan yang memenuhi syarat Undang-Undang Portabilitas dan Akuntabilitas Asuransi Kesehatan tahun 1996 (HIPAA). Untuk informasi selengkapnya, lihat Amazon Transcribe Medis. Untuk informasi tentang mengidentifikasi PHI dalam file audio, lihatMengidentifikasi PHI dalam file audio. Untuk informasi tentang mengidentifikasi PHI dalam aliran, lihatMengidentifikasi PHI dalam streaming waktu nyata.

Topik

- Mengidentifikasi PHI dalam file audio
- Mengidentifikasi PHI dalam streaming waktu nyata

Mengidentifikasi PHI dalam file audio

Gunakan pekerjaan transkripsi batch untuk menyalin file audio dan mengidentifikasi informasi kesehatan pribadi (PHI) di dalamnya. Saat Anda mengaktifkan Identifikasi Informasi Kesehatan Pribadi (PHI), Amazon Transcribe Medis memberi label PHI yang diidentifikasi dalam hasil transkripsi. Untuk informasi tentang PHI yang dapat diidentifikasi oleh Amazon Transcribe Medis, lihatMengidentifikasi informasi kesehatan pribadi (PHI) dalam transkripsi.

Anda dapat memulai pekerjaan transkripsi batch menggunakan `StartMedicalTranscriptionJob`API atau. AWS Management Console

AWS Management Console

Untuk menggunakan untuk AWS Management Console mentranskripsikan dialog klinis-pasien, buat pekerjaan transkripsi dan pilih Percakapan untuk jenis input Audio.

Untuk mentranskripsikan file audio dan mengidentifikasi PHI () -nya AWS Management Console

1. Masuk ke AWS Management Console.
2. Di panel navigasi, di bawah Amazon Transcribe Medis, pilih Pekerjaan transkripsi.
3. Pilih Buat tugas.
4. Pada halaman Tentukan detail pekerjaan, di bawah Pengaturan Job, tentukan yang berikut ini.
    a. Nama — Nama pekerjaan transkripsi yang unik untuk Anda Akun AWS.
    b. Jenis input audio - Percakapan atau Dikte.
5. Untuk bidang yang tersisa, tentukan Amazon S3 lokasi file audio Anda dan di mana Anda ingin menyimpan output dari pekerjaan transkripsi Anda.
6. Pilih Berikutnya.
7. Di bawah Pengaturan audio, pilih Identifikasi PHI.
8. Pilih Buat.

API

Untuk mentranskripsikan file audio dan mengidentifikasi PHI-nya menggunakan pekerjaan transkripsi batch (API)

- Untuk `StartMedicalTranscriptionJob`API, tentukan yang berikut ini.
    a. Untuk`MedicalTranscriptionJobName`, tentukan nama yang unik untuk Anda Akun AWS.
    b. Untuk`LanguageCode`, tentukan kode bahasa yang sesuai dengan bahasa yang digunakan dalam file audio Anda.
    c. Untuk `MediaFileUri` parameter `Media` objek, tentukan nama file audio yang ingin Anda transkripsikan.
    d. Untuk`Specialty`, tentukan spesialisasi medis dokter yang berbicara dalam file audio sebagai`PRIMARYCARE`.
    e. Untuk `Type`, tentukan salah satu, `CONVERSATION` atau `DICTATION`.

f. Untuk`OutputBucketName`, tentukan Amazon S3 bucket tempat Anda ingin menyimpan hasil transkripsi.

Berikut ini adalah contoh permintaan yang menggunakan AWS SDK untuk Python (Boto3) untuk menyalin file audio dan mengidentifikasi PHI pasien.

```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe')
job_name = "my-first-transcription-job"
job_uri = "s3://amzn-s3-demo-bucket/my-input-files/my-audio-file.flac"
transcribe.start_medical_transcription_job(
      MedicalTranscriptionJobName = job_name,
      Media = {'MediaFileUri': job_uri},
      LanguageCode = 'en-US',
      ContentIdentificationType = 'PHI',
      Specialty = 'PRIMARYCARE',
      Type = 'type', # Specify 'CONVERSATION' for a medical conversation. Specify
 'DICTATION' for a medical dictation.
      OutputBucketName = 'amzn-s3-demo-bucket'
  )
while True:
    status = transcribe.get_medical_transcription_job(MedicalTranscriptionJobName =
 job_name)
    if status['MedicalTranscriptionJob']['TranscriptionJobStatus'] in ['COMPLETED',
 'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
print(status)
```

Kode contoh berikut menunjukkan hasil transkripsi dengan PHI pasien diidentifikasi.

```
{
    "jobName": "my-medical-transcription-job-name",
    "accountId": "111122223333",
    "results": {
```

```
        "transcripts": [{
            "transcript": "The patient's name is Bertrand."
        }],
        "items": [{
                "id": 0,
            "start_time": "0.0",
            "end_time": "0.37",
            "alternatives": [{
                "confidence": "0.9993",
                "content": "The"
            }],
            "type": "pronunciation"
        }, {
                "id": 1,
            "start_time": "0.37",
            "end_time": "0.44",
            "alternatives": [{
                "confidence": "0.9981",
                "content": "patient's"
            }],
            "type": "pronunciation"
        }, {
                "id": 2,
            "start_time": "0.44",
            "end_time": "0.52",
            "alternatives": [{
                "confidence": "1.0",
                "content": "name"
            }],
            "type": "pronunciation"
        }, {
                "id": 3,
            "start_time": "0.52",
            "end_time": "0.92",
            "alternatives": [{
                "confidence": "1.0",
                "content": "is"
            }],
            "type": "pronunciation"
        }, {
                "id": 4,
            "start_time": "0.92",
            "end_time": "0.9989",
            "alternatives": [{
```

```
                "confidence": "1.0",
                "content": "Bertrand"
            }],
            "type": "pronunciation"
        }, {
                "id": 5,
            "alternatives": [{
                "confidence": "0.0",
                "content": "."
            }],
            "type": "punctuation"
        }],
        "entities": [{
            "content": "Bertrand",
            "category": "PHI*-Personal*",
            "startTime": 0.92,
            "endTime": 1.2,
            "confidence": 0.9989
        }],
        "audio_segments": [
            {
                "id": 0,
                "transcript": "The patient's name is Bertrand.",
                "start_time": "0.0",
                "end_time": "0.9989",
                "items": [
                    0,
                    1,
                    2,
                    3,
                    4,
                    5
                ]
            }
        ]
    },
    "status": "COMPLETED"
}
```

AWS CLI

Untuk mentranskripsikan file audio dan mengidentifikasi PHI menggunakan pekerjaan transkripsi batch ()AWS CLI

- Jalankan kode berikut.

```
aws transcribe start-medical-transcription-job \
--medical-transcription-job-name my-medical-transcription-job-name\
--language-code en-US \
--media MediaFileUri="s3://amzn-s3-demo-bucket/my-input-files/my-audio-file.flac" \
--output-bucket-name amzn-s3-demo-bucket \
--specialty PRIMARYCARE \
--type type \ # Choose CONVERSATION to transcribe a medical conversation.
 Choose DICTATION to transcribe a medical dictation.
--content-identification-type PHI
```

Mengidentifikasi PHI dalam streaming waktu nyata

Anda dapat mengidentifikasi Informasi Kesehatan Pribadi (PHI) baik di HTTP/2 atau streaming. WebSocket Saat Anda mengaktifkan Identifikasi PHI, Amazon Transcribe Medis memberi label PHI yang diidentifikasi dalam hasil transkripsi. Untuk informasi tentang PHI yang dapat diidentifikasi oleh Amazon Transcribe Medis, lihatMengidentifikasi informasi kesehatan pribadi (PHI) dalam transkripsi.

Mengidentifikasi PHI dalam dikte yang diucapkan ke mikrofon Anda

Untuk menggunakan AWS Management Console untuk mentranskripsikan ucapan yang diambil oleh mikrofon Anda dan mengidentifikasi PHI apa pun, pilih Dikte sebagai jenis input audio, mulai streaming, dan mulai berbicara ke mikrofon di komputer Anda.

Untuk mengidentifikasi PHI dalam dikte menggunakan AWS Management Console

1. Masuk ke AWS Management Console.
2. Di panel navigasi, pilih Transkripsi waktu nyata.
3. Untuk jenis input Audio, pilih Dikte.
4. Untuk Pengaturan tambahan, pilih identifikasi PHI.

5. Pilih Mulai streaming dan berbicara ke mikrofon.
6. Pilih Hentikan streaming untuk mengakhiri dikte.

Mengidentifikasi PHI dalam aliran HTTP/2

Untuk memulai aliran HTTP/2 dengan Identifikasi PHI diaktifkan, gunakan `StartMedicalStreamTranscription`API dan tentukan yang berikut ini:

- Untuk`LanguageCode`, tentukan kode bahasa untuk bahasa yang digunakan dalam aliran. Untuk bahasa Inggris AS, tentukan`en-US`.
- Untuk`MediaSampleHertz`, tentukan laju sampel audio.
- Untuk `content-identification-type`, tentukan `PHI`.

Mengidentifikasi PHI dalam aliran WebSocket

Untuk memulai WebSocket aliran dengan Identifikasi PHI diaktifkan, gunakan format berikut untuk membuat URL yang telah ditetapkan sebelumnya.

```
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
```

Definisi parameter dapat ditemukan di Referensi API; parameter yang umum untuk semua operasi AWS API tercantum di bagian Parameter Umum.

Menghasilkan transkripsi alternatif

Saat Anda menggunakan Amazon Transcribe Medical, Anda mendapatkan transkripsi yang memiliki tingkat kepercayaan tertinggi. Namun, Anda dapat mengonfigurasi Amazon Transcribe Medis untuk mengembalikan transkripsi tambahan dengan tingkat kepercayaan yang lebih rendah.

Gunakan transkripsi alternatif untuk melihat interpretasi yang berbeda dari audio yang ditranskripsi. Misalnya, dalam aplikasi yang memungkinkan seseorang untuk meninjau transkripsi, Anda dapat menyajikan transkripsi alternatif untuk dipilih orang tersebut.

Anda dapat membuat transkripsi alternatif dengan AWS Management Console atau `StartMedicalTranscriptionJob`API.

AWS Management Console

Untuk menggunakan transkripsi alternatif AWS Management Console untuk menghasilkan transkripsi alternatif, Anda mengaktifkan hasil alternatif saat mengonfigurasi pekerjaan Anda.

1. Masuk ke AWS Management Console.
2. Di panel navigasi, di bawah Amazon Transcribe Medis, pilih Pekerjaan transkripsi.
3. Pilih Buat tugas.
4. Pada halaman Tentukan detail pekerjaan, berikan informasi tentang pekerjaan transkripsi Anda.
5. Pilih Berikutnya.
6. Aktifkan hasil Alternatif.
7. Untuk alternatif Maksimum, masukkan nilai integer antara 2 dan 10, untuk jumlah maksimum transkripsi alternatif yang Anda inginkan dalam output.
8. Pilih Buat.

API

Untuk memisahkan teks per speaker dalam file audio menggunakan tugas transkripsi batch (API)

- Untuk `StartMedicalTranscriptionJob`API, tentukan yang berikut ini.
    a. Untuk`MedicalTranscriptionJobName`, tentukan nama yang unik di Anda Akun AWS.
    b. Untuk`LanguageCode`, tentukan kode bahasa yang sesuai dengan bahasa yang digunakan dalam file audio Anda dan bahasa filter kosakata Anda.

c. Dalam `MediaFileUri` parameter `Media` objek, tentukan lokasi file audio yang ingin Anda transkripsikan.

d. Untuk`Specialty`, tentukan spesialisasi medis dokter yang berbicara dalam file audio.

e. Untuk`Type`, tentukan apakah Anda menyalin percakapan medis atau dikte.

f. Untuk`OutputBucketName`, tentukan Amazon S3 ember untuk menyimpan hasil transkripsi.

g. Untuk `Settings` objek, tentukan yang berikut ini.

   i. `ShowAlternatives` – `true`.

   ii. `MaxAlternatives`- Bilangan bulat antara 2 dan 10 untuk menunjukkan jumlah transkripsi alternatif yang Anda inginkan dalam keluaran transkripsi.

Permintaan berikut menggunakan AWS SDK untuk Python (Boto3) untuk memulai pekerjaan transkripsi yang menghasilkan hingga dua transkripsi alternatif.

```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
job_name = "my-first-transcription-job"
job_uri = s3://amzn-s3-demo-bucket/my-input-files/my-audio-file.flac
transcribe.start_medical_transcription_job(
    MedicalTranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'amzn-s3-demo-bucket',
    OutputKey = 'my-output-files/',
    LanguageCode = 'en-US',
    Specialty = 'PRIMARYCARE',
    Type = 'CONVERSATION',
    Settings = {
        'ShowAlternatives': True,
        'MaxAlternatives': 2
    }
)

while True:
   status = transcribe.get_medical_transcription_job(MedicalTranscriptionJobName =
 job_name)
```

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

AWS CLI

Untuk mentranskripsikan file audio percakapan antara dokter perawatan primer dan pasien dalam file audio ()AWS CLI

- Jalankan kode berikut.

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

Kode berikut menunjukkan isi dari`example-start-command.json`.

```
{
      "MedicalTranscriptionJobName": "my-first-transcription-job",
      "LanguageCode": "en-US",
      "Specialty": "PRIMARYCARE",
      "Type": "CONVERSATION",
      "OutputBucketName":"amzn-s3-demo-bucket",
      "Media": {
          "MediaFileUri": "s3://amzn-s3-demo-bucket/my-input-files/my-audio-
file.flac"
        },
      "Settings":{
          "ShowAlternatives": true,
          "MaxAlternatives": 2
        }
}
```

Amazon Transcribe Titik akhir VPC medis dan antarmuka ()AWS PrivateLink

Anda dapat membuat koneksi pribadi antara VPC dan Amazon Transcribe Medis Anda dengan membuat antarmuka VPC endpoint. Endpoint antarmuka didukung oleh AWS PrivateLink, teknologi yang memungkinkan Anda mengakses Amazon Transcribe Medis secara pribadi APIs tanpa gateway internet, perangkat NAT, koneksi VPN, atau koneksi. AWS Direct Connect Instans di VPC Anda tidak memerlukan alamat IP publik untuk berkomunikasi Amazon Transcribe dengan Medis. APIs Lalu lintas antara VPC dan Amazon Transcribe Medis Anda tidak meninggalkan jaringan Amazon.

Setiap titik akhir antarmuka diwakili oleh satu atau beberapa Antarmuka Jaringan Elastis di subnet Anda.

Untuk informasi selengkapnya, lihat Titik akhir VPC Antarmuka (AWS PrivateLink) di Panduan Pengguna.Amazon VPC

Pertimbangan untuk titik akhir VPC Amazon Transcribe Medis

Sebelum Anda menyiapkan titik akhir VPC antarmuka untuk Amazon Transcribe Medis, pastikan Anda meninjau properti dan batasan titik akhir Antarmuka dalam Panduan Pengguna.Amazon VPC

Amazon Transcribe Dukungan medis melakukan panggilan ke semua tindakan API-nya dari VPC Anda.

Membuat titik akhir VPC antarmuka untuk Medis Amazon Transcribe

Anda dapat membuat titik akhir VPC untuk layanan Amazon Transcribe Medis menggunakan salah satu atau. AWS Management Console AWS CLI Untuk informasi selengkapnya, lihat Membuat titik akhir antarmuka di Panduan Amazon VPC Pengguna.

Untuk transkripsi batch di Amazon Transcribe Medical, buat titik akhir VPC menggunakan nama layanan berikut:

- com.amazonaws. *us-west-2*.transkripsikan

Untuk streaming transkripsi di Amazon Transcribe Medical, buat titik akhir VPC menggunakan nama layanan berikut:

- com.amazonaws. *us-west-2*.transcribestreaming

Jika Anda mengaktifkan DNS pribadi untuk titik akhir, Anda dapat membuat permintaan API ke Amazon Transcribe Medical menggunakan nama DNS default untuk Wilayah AWS, misalnya,. `transcribestreaming.us-east-2.amazonaws.com`

Untuk informasi selengkapnya, lihat Mengakses layanan melalui titik akhir antarmuka di Amazon VPC Panduan Pengguna.

Membuat kebijakan titik akhir VPC untuk streaming Medis Amazon Transcribe

Anda dapat melampirkan kebijakan titik akhir ke titik akhir VPC Anda yang mengontrol akses ke Medis. Amazon Transcribe Kebijakan titik akhir menentukan informasi berikut:

- Prinsipal yang dapat melakukan tindakan.
- Tindakan yang dapat dilakukan.
- Sumber daya yang menjadi target tindakan.

Untuk informasi selengkapnya, lihat Mengontrol Akses ke Layanan dengan titik akhir VPC dalam Panduan Pengguna Amazon VPC.

Contoh: Kebijakan titik akhir VPC untuk tindakan transkripsi streaming Amazon Transcribe Medis

Berikut ini adalah contoh kebijakan titik akhir untuk streaming transkripsi di Amazon Transcribe Medical. Ketika dilampirkan ke titik akhir, kebijakan ini memberikan akses ke tindakan Amazon Transcribe Medis yang terdaftar untuk semua kepala sekolah di semua sumber daya.

```
{
   "Statement":[
      {
         "Principal":"*",
         "Effect":"Allow",
         "Action":[
            "transcribe:StartMedicalStreamTranscription",
         ],
         "Resource":"*"
      }
   ]
}
```

Contoh: Kebijakan titik akhir VPC untuk tindakan transkripsi batch Amazon Transcribe Medis

Berikut ini adalah contoh kebijakan titik akhir untuk transkripsi batch di Amazon Transcribe Medical. Ketika dilampirkan ke titik akhir, kebijakan ini memberikan akses ke tindakan Amazon Transcribe Medis yang terdaftar untuk semua kepala sekolah di semua sumber daya.

```
{
   "Statement":[
      {
         "Principal":"*",
         "Effect":"Allow",
         "Action":[
            "transcribe:StartMedicalTranscriptionJob"
         ],
         "Resource":"*"
      }
   ]
}
```

Subnet bersama

Anda tidak dapat membuat, mendeskripsikan, memodifikasi, atau menghapus titik akhir VPC di subnet yang dibagikan dengan Anda. Namun, Anda dapat menggunakan titik akhir VPC di subnet yang dibagikan dengan Anda. Untuk informasi tentang berbagi VPC, lihat Berbagi VPC Anda dengan akun lain dalam panduan. Amazon Virtual Private Cloud

AWS HealthScribe

AWS HealthScribe adalah kemampuan pembelajaran mesin (ML) yang memenuhi syarat HIPAA yang menggabungkan pengenalan suara dan AI generatif untuk mentranskripsikan percakapan pasien-dokter dan menghasilkan catatan klinis. easy-to-review AWS HealthScribe membantu vendor perangkat lunak perawatan kesehatan membangun aplikasi klinis yang mengurangi beban dokumentasi dan meningkatkan pengalaman konsultasi. Layanan ini secara otomatis menyediakan transkrip percakapan yang kaya, mengidentifikasi peran pembicara, mengklasifikasikan dialog, mengekstrak istilah medis, dan menghasilkan catatan klinis awal. AWS HealthScribemenggabungkan kemampuan ini untuk menghilangkan kebutuhan untuk mengintegrasikan dan mengoptimalkan layanan AI terpisah, memungkinkan Anda untuk mempercepat implementasi.

Kasus penggunaan umum:

- Kurangi waktu dokumentasi - Memungkinkan dokter untuk menyelesaikan dokumentasi klinis dengan cepat dengan catatan klinis yang dihasilkan AI yang mudah ditinjau, disesuaikan, dan diselesaikan dalam aplikasi Anda.
- Tingkatkan efisiensi juru tulis medis - Lengkapi juru tulis medis dengan transkrip dan catatan klinis yang dihasilkan AI, bersama dengan audio konsultasi, untuk mempercepat waktu penyelesaian dokumentasi.
- Rekap kunjungan pasien yang efisien — Ciptakan pengalaman yang memungkinkan pengguna mengingat kembali sorotan utama percakapan mereka dengan cepat di aplikasi Anda.

> **Important**
>
> Hasil yang dihasilkan oleh AWS HealthScribe bersifat probabilistik dan mungkin tidak selalu akurat karena berbagai faktor, termasuk kualitas audio, kebisingan latar belakang, kejernihan speaker, kompleksitas terminologi medis, nuansa bahasa spesifik konteks, dan sifat pembelajaran mesin dan AI generatif. AWS HealthScribe dirancang untuk digunakan dalam peran bantu bagi dokter dan juru tulis medis. AWS HealthScribe output hanya boleh digunakan dalam skenario perawatan pasien, termasuk, tetapi tidak terbatas pada sebagai bagian dari Catatan Kesehatan Elektronik, setelah ditinjau untuk akurasi dan pengenaan penilaian medis yang baik oleh profesional medis terlatih. AWS HealthScribe output bukan pengganti saran medis profesional, diagnosis, atau pengobatan, dan tidak dimaksudkan

untuk menyembuhkan, mengobati, mengurangi, mencegah, atau mendiagnosis penyakit atau kondisi kesehatan apa pun.

Topik

- Keamanan
- Ketersediaan layanan
- Persyaratan teknis
- Spesialisasi Medis yang Didukung
- Alur Kerja
- AWS HealthScribe Berkas transkrip
- AWS HealthScribe File Dokumentasi Klinis
- AWS HealthScribe pekerjaan transkripsi
- AWS HealthScribe streaming
- Enkripsi Data saat istirahat untuk AWS HealthScribe

Keamanan

AWS HealthScribe beroperasi di bawah model tanggung jawab bersama, di mana AWS bertanggung jawab untuk melindungi infrastruktur yang berjalan AWS HealthScribe dan Anda bertanggung jawab untuk mengelola data Anda. Untuk informasi selengkapnya, lihat Model Tanggung Jawab Bersama.

Secara default, AWS HealthScribe menyediakan enkripsi saat istirahat untuk melindungi data pelanggan sensitif menggunakan kunci Amazon S3-managed. Saat membuat pekerjaan AWS HealthScribe transkripsi atau memulai streaming, Anda dapat menentukan kunci yang dikelola pelanggan. Ini menambahkan lapisan enkripsi kedua. Untuk informasi selengkapnya, lihat Enkripsi Data saat istirahat untuk AWS HealthScribe.

Ketersediaan layanan

AWS HealthScribe tersedia di wilayah US East (Virginia N.).

Persyaratan teknis

- Bahasa yang Didukung: US English (en-US)

- Format Audio yang Disarankan: Audio tanpa kehilangan (seperti FLAC atau WAV)
- Pengkodean: PCM 16-bit
- Tingkat Sampel: 16.000 Hz atau lebih tinggi

Spesialisasi Medis yang Didukung

AWS HealthScribe saat ini mendukung spesialisasi berikut:

- Kedokteran Umum
- Ortopedi

Alur Kerja

AWS HealthScribe alur kerja meliputi pekerjaan transkripsi dan streaming. Setelah Anda menjalankan tugas transkripsi atau menyelesaikan aliran, AWS HealthScribe buat file transkrip, dengan keluaran turn-by-turn transkripsi dan wawasan untuk setiap giliran percakapan. Juga menghasilkan file dokumentasi klinis, dengan ringkasan dan tautan bukti. Untuk informasi lebih lanjut, lihat AWS HealthScribe Berkas transkrip dan AWS HealthScribe File Dokumentasi Klinis.

- Pekerjaan transkripsi — Dengan pekerjaan transkripsi, AWS HealthScribe menganalisis file media konsultasi medis yang lengkap dari ember Amazon S3. Berikut ini adalah operasi API khusus untuk pekerjaan AWS HealthScribe transkripsi.
  - StartMedicalScribeJob
  - ListMedicalScribeJobs
  - GetMedicalScribeJob
  - DeleteMedicalScribeJob

  Untuk informasi selengkapnya, termasuk contoh kode, lihatAWS HealthScribe pekerjaan transkripsi.
- Streaming adalah layanan bi-directional HTTP2 berbasis real-time yang menerima aliran audio pada satu saluran dan menjual transkripsi audio di saluran lainnya. AWS HealthScribe

  Berikut ini adalah operasi API khusus untuk AWS HealthScribe streaming:
  - StartMedicalScribeStream

- GetMedicalScribeStream

Untuk informasi selengkapnya, termasuk contoh kode, lihatAWS HealthScribe streaming.

AWS HealthScribe Berkas transkrip

Dalam file transkrip, selain output turn-by-turn transkripsi standar dengan stempel waktu tingkat kata, AWS HealthScribe memberi Anda:

- Deteksi peran peserta sehingga Anda dapat membedakan pasien dari dokter dalam transkrip percakapan.
- Pembagian transkrip, yang mengkategorikan dialog transkrip berdasarkan relevansi klinisnya seperti obrolan ringan, subjektif, objektif, dll. Ini dapat digunakan untuk menunjukkan bagian tertentu dari transkrip.
- Entitas klinis, yang mencakup informasi terstruktur seperti obat-obatan, kondisi medis, dan perawatan yang disebutkan dalam percakapan.

Selain itu, wawasan berikut disediakan untuk setiap giliran percakapan:

- Peran peserta — Setiap peserta diberi label sebagai dokter atau pasien. Jika percakapan memiliki lebih dari satu peserta di setiap kategori, setiap peserta diberi nomor. Misalnya,`CLINICIAN_0`, `CLINICIAN_1` dan`PATIENT_0`,`PATIENT_1`.
- Bagian - Setiap giliran dialog ditugaskan ke salah satu dari empat bagian yang mungkin berdasarkan konten yang diidentifikasi.
  - Subyektif — Informasi yang diberikan oleh pasien tentang masalah kesehatan mereka.
  - Objektif — Informasi yang diamati oleh dokter melalui pemeriksaan fisik, laboratorium, pencitraan, atau tes diagnostik.
  - Penilaian dan Rencana — Informasi yang berkaitan dengan penilaian dokter dan rencana perawatan.
  - Kunjungi Manajemen Aliran — Informasi yang terkait dengan obrolan ringan atau transisi.
- Wawasan — Ekstrak entitas yang relevan secara klinis (`ClinicalEntity`) yang ada dalam percakapan. AWS HealthScribe mendeteksi semua entitas klinis yang didukung oleh Amazon Comprehend Medical.

Untuk contoh transkrip dari pekerjaan transkripsi, lihat keluaran transkrip di. Contoh keluaran pekerjaan transkripsi Untuk contoh transkrip dari streaming, lihat output transkrip di. Contoh keluaran transkripsi streaming

AWS HealthScribe File Dokumentasi Klinis

AWS HealthScribe dapat menggunakan salah satu templat berikut untuk ringkasan catatan klinis. Nilai default-nya `HISTORY_AND_PHYSICAL`.

- HISTORY_AND_PHYSICAL: Memberikan ringkasan untuk bagian kunci dari dokumentasi klinis. Contoh bagian termasuk Keluhan Kepala, Sejarah Penyakit Saat Ini, Tinjauan Sistem, Riwayat Medis Masa Lalu, Penilaian, dan Rencana.
- GIRPP: Memberikan ringkasan berdasarkan kemajuan pasien menuju tujuan. Contoh bagian termasuk Tujuan, Intervensi, Respons, Kemajuan, dan Rencana.
- BIRP: Berfokus pada pola dan respons perilaku pasien. Contoh bagian termasuk Perilaku, Intervensi, Respons, dan Rencana.
- SIRP: Menekankan konteks situasional terapi. Contoh bagian termasuk Situasi, Intervensi, Respons, dan Rencana.
- DAP: Menyediakan format yang disederhanakan untuk dokumentasi klinis. Contoh bagian termasuk Data, Penilaian, dan Rencana.
- BH_SOAP: Format dokumentasi yang berfokus pada kesehatan perilaku. Contoh bagian termasuk Subyektif, Objektif, Penilaian, dan Rencana.
- PH_SOAP: Format dokumentasi yang berfokus pada kesehatan fisik. Contoh bagian termasuk Subyektif, Objektif, Penilaian, dan Rencana.

Untuk menentukan template apa yang akan digunakan, lakukan hal berikut:

- Untuk tugas transkripsi, tentukan template yang akan `NoteTemplate` digunakan dalam operasi StartMedicalScribeJobAPI Anda. ClinicalNoteGenerationSettings`Settings`
- Untuk streaming, Anda menentukan template yang akan digunakan di `NoteTemplate` ClinicalNoteGenerationSettingsbagian `PostStreamAnalyticsSettings` dari Anda MedicalScribeConfigurationEvent.

Topik

- Bagian template HISTORY_AND_PHYSICAL

- Bagian template GIRPP
- Bagian template BIRP
- Bagian template SIRP
- Bagian template DAP
- Bagian template BH_SOAP
- Bagian template PH_SOAP

Bagian template HISTORY_AND_PHYSICAL

Template wawasan HISTORY_AND_PHYSICAL mencakup bagian berikut.

Bagian	Deskripsi
KELUHAN UTAMA	Deskripsi singkat untuk alasan pasien mengunjungi dokter.
RIWAYAT PENYAKIT SAAT INI	Catatan yang memberikan informasi tentang penyakit pasien, termasuk referensi ke tingkat keparahan, onset, waktu gejala, perawatan saat ini, dan daerah yang terkena.
TINJAUAN SISTEM	Evaluasi gejala yang dilaporkan pasien di berbagai sistem tubuh.
RIWAYAT MEDIS MASA LALU	Merinci kondisi medis, operasi, dan perawatan pasien sebelumnya.
PENILAIAN	Catatan yang memberikan informasi tentang penilaian dokter terhadap kesehatan pasien.
RENCANA	Catatan yang merujuk setiap perawatan medis, penyesuaian gaya hidup, dan janji lebih lanjut.
FISIK_PEMERIKSAAN	Dokumentasi temuan dokter dari pemeriksaan fisik sistem tubuh pasien dan tanda-tanda vital.

Bagian	Deskripsi
PAST_FAMILY_HISTORY	Informasi tentang kondisi kesehatan yang berjalan dalam keluarga pasien.
PAST_SOCIAL_HISTORY	Rincian tentang kehidupan sosial pasien, kebiasaan, pekerjaan, dan faktor lingkungan yang mempengaruhi kesehatan.
DIAGNOSTIC_TESTING	Hasil dan interpretasi tes laboratorium, studi pencitraan, dan prosedur diagnostik lainnya.

Setiap kalimat yang ada dalam `Summary` termasuk `EvidenceLinks` yang menyediakan dialog `SegmentId` yang relevan dalam transkrip yang dirangkum. Ini membantu pengguna memvalidasi akurasi ringkasan dalam aplikasi Anda. Seperti penjelasan, menyediakan ketertelusuran dan transparansi untuk wawasan yang dihasilkan AI konsisten dengan prinsip AI yang Bertanggung Jawab. Memberikan referensi ini bersama dengan catatan ringkasan kepada dokter atau juru tulis medis membantu menumbuhkan kepercayaan dan mendorong penggunaan AI yang aman dalam pengaturan klinis.

Untuk contoh file Dokumentasi Klinis dari pekerjaan transkripsi, lihat contoh file Dokumentasi Klinis diContoh keluaran pekerjaan transkripsi. Untuk contoh file Dokumentasi Klinis dari streaming, lihat contoh file Dokumentasi Klinis diContoh keluaran transkripsi streaming.

Bagian template GIRPP

Template wawasan GIRPP mencakup bagian berikut.

Bagian	Deskripsi
Tujuan	Masalah, tantangan, atau perilaku yang diidentifikasi yang perlu ditangani melalui pengobatan.
Intervensi	Perawatan, metode, atau teknik khusus yang digunakan oleh dokter untuk membantu pasien mengatasi tujuan yang diidentifikasi.

Bagian	Deskripsi
Respons	Bagaimana pasien menanggapi intervensi, termasuk tingkat partisipasi, reaksi, dan umpan balik mereka.
Kemajuan	Penilaian dokter tentang gerakan menuju tujuan pengobatan, termasuk pengamatan perbaikan atau hambatan pasien.
Rencana	Langkah selanjutnya dalam perawatan , termasuk intervensi masa depan, tugas pekerjaan rumah, dan rujukan.

Bagian template BIRP

Template wawasan BIRP mencakup bagian berikut.

Bagian	Deskripsi
Perilaku	Masalah yang disajikan pasien dan respons mereka terhadap perawatan.
Intervensi	Perawatan, metode, atau teknik khusus yang digunakan oleh dokter untuk membantu pasien mengatasi tujuan yang diidentifikasi.
Respons	Bagaimana pasien menanggapi intervensi, termasuk tingkat partisipasi, reaksi, dan umpan balik mereka.
Rencana	Langkah selanjutnya dalam perawatan , termasuk intervensi masa depan, tugas pekerjaan rumah, dan rujukan.

Bagian template SIRP

Template wawasan SIRP mencakup bagian-bagian berikut.

Bagian	Deskripsi
Situasi	Masalah yang disajikan pasien dan tujuan mereka untuk mencari terapi.
Intervensi	Perawatan, metode, atau teknik khusus yang digunakan oleh dokter untuk membantu pasien mengatasi tujuan yang diidentifikasi.
Respons	Bagaimana pasien menanggapi intervensi, termasuk tingkat partisipasi, reaksi, dan umpan balik mereka.
Rencana	Langkah selanjutnya dalam perawatan , termasuk intervensi masa depan, tugas pekerjaan rumah, dan rujukan.

Bagian template DAP

Template wawasan DAP mencakup bagian-bagian berikut.

Bagian	Deskripsi
Data	Alasan pasien untuk mencari pengobatan dan informasi tentang pasien.
Penilaian	Diagnosis dokter tentang situasi pasien.
Rencana	Langkah selanjutnya dalam perawatan , termasuk intervensi masa depan, tugas pekerjaan rumah, dan rujukan.

Bagian template BH_SOAP

Template wawasan BH_SOAP mencakup bagian berikut.

Bagian	Deskripsi
Subyektif	Tujuan klien untuk perawatan dan pengalaman mereka, serta masalah yang ada dan masa lalu.
Tujuan	Data dan fakta tentang klien.
Penilaian	Diagnosis dokter tentang situasi pasien.
Rencana	Langkah selanjutnya dalam perawatan, termasuk intervensi masa depan, tugas pekerjaan rumah, dan rujukan.

Bagian template PH_SOAP

Template wawasan PH_SOAP mencakup bagian berikut.

Bagian	Deskripsi
Subyektif	Tujuan klien untuk perawatan dan pengalaman mereka, serta masalah yang ada dan masa lalu.
Tujuan	Data dan fakta tentang klien.
Penilaian	Diagnosis dokter tentang situasi pasien.
Rencana	Langkah selanjutnya dalam perawatan, termasuk intervensi masa depan, tugas pekerjaan rumah, dan rujukan.

AWS HealthScribe pekerjaan transkripsi

Pekerjaan AWS HealthScribe transkripsi memproses file media dari bucket Amazon S3. Ketika memproses file media, ia menyalin percakapan pasien-dokter dan menganalisis konsultasi medis untuk menghasilkan dua file keluaran JSON: file transkrip dan file dokumentasi klinis.

Berikut ini adalah operasi API khusus untuk pekerjaan AWS HealthScribe transkripsi:

- StartMedicalScribeJob
- ListMedicalScribeJobs
- GetMedicalScribeJob
- DeleteMedicalScribeJob

Memulai AWS HealthScribe pekerjaan transkripsi

Anda dapat memulai AWS HealthScribe pekerjaan menggunakan AWS CLI atau. AWS SDKs

AWS CLI

Contoh ini menggunakan start-medical-scribe-jobperintah. Untuk informasi selengkapnya, lihat StartMedicalScribeJob.

```
aws transcribe start-medical-scribe-job \
--region us-west-2 \
--medical-scribe-job-name my-first-medical-scribe-job \
--media MediaFileUri=s3://amzn-s3-demo-bucket/my-input-files/my-media-file.flac \
--output-bucket-name amzn-s3-demo-bucket \
--DataAccessRoleArn=arn:aws:iam::111122223333:role/ExampleRole \
--settings ShowSpeakerLabels=false,ChannelIdentification=true \
--channel-definitions ChannelId=0,ParticipantRole=CLINICIAN
 ChannelId=1,ParticipantRole=PATIENT
```

Berikut adalah contoh lain menggunakan start-medical-scribe-jobperintah, dan badan permintaan dengan pengaturan tambahan.

```
aws transcribe start-medical-scribe-job \
--region us-west-2 \
```

```
--cli-input-json file://filepath/my-first-medical-scribe-job.json
```

File `my-first-medical-scribe-job.json` berisi badan permintaan berikut.

```
{
  "MedicalScribeJobName": "my-first-medical-scribe-job",
  "Media": {
    "MediaFileUri": "s3://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
   },
  "OutputBucketName": "amzn-s3-demo-bucket",
  "DataAccessRoleArn": "arn:aws:iam::111122223333:role/ExampleRole",
  "Settings": {
    "ShowSpeakerLabels": false,
    "ChannelIdentification": true
  },
  "ChannelDefinitions": [
    {
      "ChannelId": 0,
      "ParticipantRole":"CLINICIAN"
    }, {
      "ChannelId": 1,
      "ParticipantRole":"PATIENT"
    }
  ]
}
```

AWS SDK untuk Python (Boto3)

Contoh berikut menggunakan AWS SDK untuk Python (Boto3) untuk membuat permintaan start_medical_scribe_job. Untuk informasi selengkapnya, lihat StartMedicalScribeJob.

```
from __future__ import print_functionimport timeimport boto3
transcribe = boto3.client('transcribe', 'us-west-2')
job_name = "my-first-medical-scribe-job"
job_uri = "s3://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
transcribe.start_medical_scribe_job(
    MedicalScribeJobName = job_name,
    Media = {
      'MediaFileUri': job_uri
```

```
    },
    OutputBucketName = 'amzn-s3-demo-bucket',
    DataAccessRoleArn = 'arn:aws:iam::111122223333:role/ExampleRole',
    Settings = {
      'ShowSpeakerLabels': false,
      'ChannelIdentification': true
    },
    ChannelDefinitions = [
      {
        'ChannelId': 0,
        'ParticipantRole': 'CLINICIAN'
      }, {
        'ChannelId': 1,
        'ParticipantRole': 'PATIENT'
      }
    ]
)
while True:
    status = transcribe.get_medical_scribe_job(MedicalScribeJobName = job_name)
    if status['MedicalScribeJob']['MedicalScribeJobStatus'] in ['COMPLETED', 'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
print(status)
```

> **Note**
>
> Konsol AWS Manajemen saat ini tidak mendukung AWS HealthScribe pekerjaan.

Contoh keluaran pekerjaan transkripsi

Selain transkrip, `StartMedicalScribeJob` permintaan menghasilkan file dokumentasi klinis terpisah. Kedua file dalam format JSON dan disimpan di lokasi output yang Anda tentukan dalam permintaan Anda. Berikut adalah contoh dari setiap jenis output:

Contoh keluaran transkrip

File AWS HealthScribe transkrip (dari `StartMedicalScribeJob` permintaan) memiliki format berikut:

```
{
  "Conversation": {
    "ConversationId": "sampleConversationUUID",
    "JobName": "sampleJobName",
    "JobType": "ASYNC",
    "LanguageCode": "en-US",
    "ClinicalInsights": [
      {
        "Attributes": [],
        "Category": "MEDICAL_CONDITION",
        "InsightId": "insightUUID1",
        "InsightType": "ClinicalEntity",
        "Spans": [
          {
            "BeginCharacterOffset": 12,
            "Content": "pain",
            "EndCharacterOffset": 15,
            "SegmentId": "uuid1"
          }
        ],
        "Type": "DX_NAME"
      },
      {
        "Attributes": [],
        "Category": "TEST_TREATMENT_PROCEDURE",
        "InsightId": "insightUUID2",
        "InsightType": "ClinicalEntity",
        "Spans": [
          {
            "BeginCharacterOffset": 4,
            "Content": "mammogram",
            "EndCharacterOffset": 12,
            "SegmentId": "uuid2"
          }
        ],
        "Type": "TEST_NAME"
      },
      {
        "Attributes": [],
        "Category": "TEST_TREATMENT_PROCEDURE",
        "InsightId": "insightUUID3",
        "InsightType": "ClinicalEntity",
```

```
        "Spans": [
          {
            "BeginCharacterOffset": 15,
            "Content": "pap smear",
            "EndCharacterOffset": 23,
            "SegmentId": "uuid3"
          }
        ],
        "Type": "TEST_NAME"
      },
      {
        "Attributes": [],
        "Category": "MEDICATION",
        "InsightId": "insightUUID4",
        "InsightType": "ClinicalEntity",
        "Spans": [
          {
            "BeginCharacterOffset": 28,
            "Content": "phentermine",
            "EndCharacterOffset": 38,
            "SegmentId": "uuid4"
          }
        ],
        "Type": "GENERIC_NAME"
      },
      {
        "Attributes": [
          {
            "AttributeId": "attributeUUID1",
            "Spans": [
              {
                "BeginCharacterOffset": 38,
                "Content": "high",
                "EndCharacterOffset": 41,
                "SegmentId": "uuid5"
              }
            ],
            "Type": "TEST_VALUE"
          }
        ],
        "Category": "TEST_TREATMENT_PROCEDURE",
        "InsightId": "insightUUID5",
        "InsightType": "ClinicalEntity",
        "Spans": [
```

```
          {
            "BeginCharacterOffset": 14,
            "Content": "weight",
            "EndCharacterOffset": 19,
            "SegmentId": "uuid6"
          }
        ],
        "Type": "TEST_NAME"
      },
      {
        "Attributes": [],
        "Category": "ANATOMY",
        "InsightId": "insightUUID6",
        "InsightType": "ClinicalEntity",
        "Spans": [
          {
            "BeginCharacterOffset": 60,
            "Content": "heart",
            "EndCharacterOffset": 64,
            "SegmentId": "uuid7"
          }
        ],
        "Type": "SYSTEM_ORGAN_SITE"
      }
    ],
    "TranscriptItems": [
      {
        "Alternatives": [
          {
            "Confidence": 0.7925,
            "Content": "Okay"
          }
        ],
        "BeginAudioTime": 0.16,
        "EndAudioTime": 0.6,
        "Type": "PRONUNCIATION"
      },
      {
        "Alternatives": [
          {
            "Confidence": 0,
            "Content": "."
          }
        ],
```

```
      "BeginAudioTime": 0.17,
      "EndAudioTime": 0.9,
      "Type": "PUNCTUATION"
    },
    {
      "Alternatives": [
        {
          "Confidence": 1,
          "Content": "Good"
        }
      ],
      "BeginAudioTime": 0.61,
      "EndAudioTime": 0.92,
      "Type": "PRONUNCIATION"
    },
    {
      "Alternatives": [
        {
          "Confidence": 1,
          "Content": "afternoon"
        }
      ],
      "BeginAudioTime": 0.92,
      "EndAudioTime": 1.54,
      "Type": "PRONUNCIATION"
    },
    {
      "Alternatives": [
        {
          "Confidence": 0,
          "Content": "."
        }
      ],
      "BeginAudioTime": 0,
      "EndAudioTime": 0,
      "Type": "PUNCTUATION"
    },
    {
      "Alternatives": [
        {
          "Confidence": 0.9924,
          "Content": "You"
        }
      ],
```

```
        "BeginAudioTime": 1.55,
        "EndAudioTime": 1.88,
        "Type": "PRONUNCIATION"
      },
      {
        "Alternatives": [
          {
            "Confidence": 1,
            "Content": "lost"
          }
        ],
        "BeginAudioTime": 1.88,
        "EndAudioTime": 2.19,
        "Type": "PRONUNCIATION"
      },
      {
        "Alternatives": [
          {
            "Confidence": 1,
            "Content": "one"
          }
        ],
        "BeginAudioTime": 2.19,
        "EndAudioTime": 2.4,
        "Type": "PRONUNCIATION"
      },
      {
        "Alternatives": [
          {
            "Confidence": 1,
            "Content": "lb"
          }
        ],
        "BeginAudioTime": 2.4,
        "EndAudioTime": 2.97,
        "Type": "PRONUNCIATION"
      }
    ],
    "TranscriptSegments": [
      {
        "BeginAudioTime": 0.16,
        "Content": "Okay.",
        "EndAudioTime": 0.6,
        "ParticipantDetails": {
```

```
        "ParticipantRole": "CLINICIAN_0"
      },
      "SectionDetails": {
        "SectionName": "SUBJECTIVE"
      },
      "SegmentId": "uuid1"
    },
    {
      "BeginAudioTime": 0.61,
      "Content": "Good afternoon.",
      "EndAudioTime": 1.54,
      "ParticipantDetails": {
        "ParticipantRole": "CLINICIAN_0"
      },
      "SectionDetails": {
        "SectionName": "OTHER"
      },
      "SegmentId": "uuid2"
    },
    {
      "BeginAudioTime": 1.55,
      "Content": "You lost one lb.",
      "EndAudioTime": 2.97,
      "ParticipantDetails": {
        "ParticipantRole": "CLINICIAN_0"
      },
      "SectionDetails": {
        "SectionName": "SUBJECTIVE"
      },
      "SegmentId": "uuid3"
    },
    {
      "BeginAudioTime": 2.98,
      "Content": "Yeah, I think it, uh, do you feel more energy?",
      "EndAudioTime": 6.95,
      "ParticipantDetails": {
        "ParticipantRole": "CLINICIAN_0"
      },
      "SectionDetails": {
        "SectionName": "SUBJECTIVE"
      },
      "SegmentId": "uuid5"
    },
    {
```

```
        "BeginAudioTime": 6.96,
        "Content": "Yes.",
        "EndAudioTime": 7.88,
        "ParticipantDetails": {
          "ParticipantRole": "CLINICIAN_0"
        },
        "SectionDetails": {
          "SectionName": "SUBJECTIVE"
        },
        "SegmentId": "uuid6"
      },
      {
        "BeginAudioTime": 7.89,
        "Content": "Uh, how about craving for the carbohydrate or sugar or fat or
 anything?",
        "EndAudioTime": 17.93,
        "ParticipantDetails": {
          "ParticipantRole": "CLINICIAN_0"
        },
        "SectionDetails": {
          "SectionName": "SUBJECTIVE"
        },
        "SegmentId": "uuid7"
      }
    ]
  }
}
```

Berikut adalah contoh lain menggunakan start-medical-scribe-jobperintah, dan badan permintaan dengan pengaturan tambahan.

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

File `my-first-medical-scribe-job.json` berisi badan permintaan berikut.

```
{
  "MedicalScribeJobName": "my-first-medical-scribe-job",
```

```
  "Media": {
    "MediaFileUri": "s3://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
   },
  "OutputBucketName": "amzn-s3-demo-bucket",
  "DataAccessRoleArn": "arn:aws:iam::111122223333:role/ExampleRole",
  "Settings": {
    "ShowSpeakerLabels": false,
    "ChannelIdentification": true
  },
  "ChannelDefinitions": [
    {
      "ChannelId": 0,
      "ParticipantRole":"CLINICIAN"
    }, {
      "ChannelId": 1,
      "ParticipantRole":"PATIENT"
    }
  ]
}
```

Contoh keluaran dokumentasi klinis

File wawasan dokumentasi (dari `StartMedicalScribeJob` permintaan) memiliki format berikut:

```
{
  "ClinicalDocumentation": {
    "Sections": [
      {
        "SectionName": "CHIEF_COMPLAINT",
        "Summary": [
          {
            "EvidenceLinks": [
              {
                "SegmentId": "uuid1"
              },
              {
                "SegmentId": "uuid2"
              },
              {
                "SegmentId": "uuid3"
              },
              {
```

```
              "SegmentId": "uuid4"
            },
            {
              "SegmentId": "uuid5"
            },
            {
              "SegmentId": "uuid6"
            }
          ],
          "SummarizedSegment": "Weight loss."
        }
      ]
    },
    {
      "SectionName": "HISTORY_OF_PRESENT_ILLNESS",
      "Summary": [
        {
          "EvidenceLinks": [
            {
              "SegmentId": "uuid7"
            },
            {
              "SegmentId": "uuid8"
            },
            {
              "SegmentId": "uuid9"
            },
            {
              "SegmentId": "uuid10"
            }
          ],
          "SummarizedSegment": "The patient is seen today for a follow-up of weight
loss."
        },
        {
          "EvidenceLinks": [
            {
              "SegmentId": "uuid11"
            },
            {
              "SegmentId": "uuid12"
            },
            {
              "SegmentId": "uuid13"
```

```
            }
          ],
          "SummarizedSegment": "They report feeling more energy and craving
carbohydrates, sugar, and fat."
        },
        {
          "EvidenceLinks": [
            {
              "SegmentId": "uuid14"
            },
            {
              "SegmentId": "uuid15"
            },
            {
              "SegmentId": "uuid16"
            }
          ],
          "SummarizedSegment": "The patient is up to date on their mammogram and pap
smear."
        },
        {
          "EvidenceLinks": [
            {
              "SegmentId": "uuid17"
            },
            {
              "SegmentId": "uuid18"
            },
            {
              "SegmentId": "uuid19"
            },
            {
              "SegmentId": "uuid20"
            }
          ],
          "SummarizedSegment": "The patient is taking phentermine and would like to
continue."
        }
      ]
    },
    {
      "SectionName": "REVIEW_OF_SYSTEMS",
      "Summary": [
        {
```

```
            "EvidenceLinks": [
              {
                "SegmentId": "uuid21"
              },
              {
                "SegmentId": "uuid22"
              }
            ],
            "SummarizedSegment": "Patient reports intermittent headaches, occasional
chest pains but denies any recent fevers or chills."
          },
          {
            "EvidenceLinks": [
              {
                "SegmentId": "uuid23"
              },
              {
                "SegmentId": "uuid24"
              }
            ],
            "SummarizedSegment": "No recent changes in vision, hearing, or any
respiratory complaints."
          }
        ]
      },
      {
        "SectionName": "PAST_MEDICAL_HISTORY",
        "Summary": [
          {
            "EvidenceLinks": [
              {
                "SegmentId": "uuid25"
              },
              {
                "SegmentId": "uuid26"
              }
            ],
            "SummarizedSegment": "Patient has a history of hypertension and was
diagnosed with Type II diabetes 5 years ago."
          },
          {
            "EvidenceLinks": [
              {
                "SegmentId": "uuid27"
```

```
            },
            {
              "SegmentId": "uuid28"
            }
          ],
          "SummarizedSegment": "Underwent an appendectomy in the early '90s and had a
fracture in the left arm during childhood."
        }
      ]
    },
    {
      "SectionName": "ASSESSMENT",
      "Summary": [
        {
          "EvidenceLinks": [
            {
              "SegmentId": "uuid29"
            },
            {
              "SegmentId": "uuid30"
            }
          ],
          "SummarizedSegment": "Weight loss"
        }
      ]
    },
    {
      "SectionName": "PLAN",
      "Summary": [
        {
          "EvidenceLinks": [
            {
              "SegmentId": "uuid31"
            },
            {
              "SegmentId": "uuid32"
            },
            {
              "SegmentId": "uuid33"
            },
            {
              "SegmentId": "uuid34"
            }
          ],
```

```
            "SummarizedSegment": "For the condition of Weight loss: The patient was
 given a 30-day supply of phentermine and was advised to follow up in 30 days."
          }
        ]
      }
    ]
  }
}
```

AWS HealthScribe streaming

Dengan AWS HealthScribe streaming, Anda dapat menyalin percakapan medis secara real-time. AWS HealthScribe streaming adalah layanan bi-directional HTTP2 berbasis real-time yang menerima aliran audio pada satu saluran dan menjual transkripsi audio di saluran lain. Setelah streaming selesai, AWS HealthScribe analisis konten aliran dan menghasilkan file JSON transkrip dan file JSON catatan klinis.

Untuk memulai streaming, gunakan operasi StartMedicalScribeStreamAPI. API ini memulai saluran bi-directional HTTP2 berbasis yang Anda gunakan untuk melakukan streaming peristiwa audio.

Saat Anda memulai streaming, tentukan terlebih dahulu konfigurasi aliran di file`MedicalScribeConfigurationEvent`. Acara ini mencakup definisi saluran, pengaturan enkripsi, dan pengaturan analitik pasca-aliran, seperti konfigurasi keluaran untuk transkrip agregat dan pembuatan catatan klinis.

Setelah Anda mulai streaming audio, Anda mengelola streaming sebagai berikut:

- Setelah selesai, untuk mulai memproses hasil dengan analitik pasca-aliran, kirim `MedicalScribeSessionControlEvent` dengan a `Type` `END_OF_SESSION` dan AWS HealthScribe mulai analitik.
- Untuk menjeda streaming, selesaikan aliran input tanpa mengirim file. `MedicalScribeSessionControlEvent`
- Untuk melanjutkan aliran yang dijeda, gunakan operasi `StartMedicalScribeStream` API dan tentukan hal yang sama. `SessionId` Ini adalah yang `SessionId` Anda gunakan ketika Anda awalnya memulai streaming.

Topik

- Pedoman dan persyaratan
- ResourceAccessRoleArn izin peran
- Memulai transkripsi AWS HealthScribe streaming

Pedoman dan persyaratan

Berikut ini adalah pedoman dan persyaratan untuk AWS HealthScribe streaming:

- Sebelum mengirim acara audio, Anda harus terlebih dahulu menentukan konfigurasi aliran di file`MedicalScribeConfigurationEvent`.
- Untuk menjalankan analitik pasca-aliran, di `ResourceAccessRoleArn` dalam Anda `MedicalScribeConfigurationEvent` harus memiliki izin yang benar. Untuk informasi selengkapnya, lihat ResourceAccessRoleArn izin peran.
- Anda dapat melanjutkan sesi beberapa kali dalam waktu 5 jam dari pembuatan aliran awal.
- Anda dapat melakukan streaming paling banyak 2 jam audio selama satu sesi di semua permintaan streaming.
- Secara default, AWS HealthScribe menyediakan enkripsi saat istirahat untuk melindungi data pelanggan sensitif menggunakan kunci Amazon S3-managed. Saat memulai streaming, Anda dapat menentukan AWS KMS kunci untuk lapisan enkripsi kedua. Anda `ResourceAccessRoleArn` harus memiliki izin untuk menggunakan AWS KMS kunci Anda. Untuk informasi selengkapnya, lihat Enkripsi Data saat istirahat untuk AWS HealthScribe.
- Anda dapat menggunakan AWS HealthScribe streaming dengan AWS SDKs, tidak termasuk SDK for Python (Boto3) dan SDK for PHP.
- Jika `LimitExceededException` pengecualian terjadi setelah Anda mengakhiri streaming, Anda dapat memulai ulang sesi dan masih menghasilkan analitik pasca-aliran. Untuk memulai ulang aliran, gunakan StartMedicalScribeStreamAPI dan gunakan yang sama`SessionID`. Kemudian kirim `MedicalScribeSessionControlEvent` dengan a `Type` `END_OF_SESSION` dan AWS HealthScribe mulai analitik.

ResourceAccessRoleArn izin peran

Untuk menjalankan analitik pasca-aliran, `ResourceAccessRoleArn` in Anda `MedicalScribeConfigurationEvent` harus dapat mengakses bucket keluaran Amazon S3 dan, jika Anda menyediakannya, AWS KMS kunci Anda. Selain itu, kebijakan kepercayaan peran harus

memberikan izin `transcribe.streaming.amazonaws.com` layanan untuk mengambil peran tersebut.

Berikut ini adalah contoh kebijakan IAM yang memberikan izin bucket Amazon S3 dan izin AWS KMS kunci. Untuk informasi selengkapnya, lihat Enkripsi Data saat istirahat untuk AWS HealthScribe.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "kms:DescribeKey",
                "kms:Decrypt",
                "kms:Encrypt",
                "kms:GenerateDataKey*"
            ],
            "Resource": "arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
            "Effect": "Allow",
        }
    ]
}
```

Berikut ini adalah contoh kebijakan kepercayaan.

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

```
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Memulai transkripsi AWS HealthScribe streaming

Contoh kode berikut menunjukkan cara mengatur transkripsi AWS HealthScribe streaming menggunakan. AWS SDKs

Topik

- SDK untuk Java 2.x
- Contoh keluaran transkripsi streaming

SDK untuk Java 2.x

Contoh berikut menggunakan SDK for Java 2.x untuk mengatur streaming dan membuat StartMedicalScribeStreampermintaan.

```
package org.example;

import io.reactivex.rxjava3.core.BackpressureStrategy;
import io.reactivex.rxjava3.core.Flowable;
import org.reactivestreams.Publisher;
import org.reactivestreams.Subscriber;
import software.amazon.awssdk.auth.credentials.AwsCredentialsProvider;
import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
import software.amazon.awssdk.core.SdkBytes;
import software.amazon.awssdk.regions.Region;
import
 software.amazon.awssdk.services.transcribestreaming.TranscribeStreamingAsyncClient;
import
 software.amazon.awssdk.services.transcribestreaming.model.ClinicalNoteGenerationSettings;
import software.amazon.awssdk.services.transcribestreaming.model.LanguageCode;
import software.amazon.awssdk.services.transcribestreaming.model.MediaEncoding;

import
 software.amazon.awssdk.services.transcribestreaming.model.MedicalScribeInputStream;
```

```
import
 software.amazon.awssdk.services.transcribestreaming.model.MedicalScribePostStreamAnalyticsSett
import
 software.amazon.awssdk.services.transcribestreaming.model.MedicalScribeSessionControlEventType
import
 software.amazon.awssdk.services.transcribestreaming.model.MedicalScribeTranscriptEvent;
import
 software.amazon.awssdk.services.transcribestreaming.model.MedicalScribeTranscriptSegment;
import
 software.amazon.awssdk.services.transcribestreaming.model.StartMedicalScribeStreamRequest;
import
 software.amazon.awssdk.services.transcribestreaming.model.StartMedicalScribeStreamResponseHand
import
 software.amazon.awssdk.services.transcribestreaming.model.medicalscribeinputstream.DefaultConf

import software.amazon.awssdk.http.nio.netty.NettyNioAsyncHttpClient;

import javax.sound.sampled.AudioFormat;
import javax.sound.sampled.AudioInputStream;
import javax.sound.sampled.AudioSystem;
import javax.sound.sampled.DataLine;
import javax.sound.sampled.LineUnavailableException;
import javax.sound.sampled.TargetDataLine;
import java.io.BufferedInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.UncheckedIOException;
import java.util.Arrays;
import java.util.concurrent.CompletableFuture;


public class HealthScribeStreamingDemoApp {
    private static final int CHUNK_SIZE_IN_BYTES = 6400;
    private static final int SAMPLE_RATE = 16000;
    private static final Region REGION = Region.US_EAST_1;
    private static final String sessionId = "1234abcd-12ab-34cd-56ef-123456SAMPLE";
    private static final String bucketName = "amzn-s3-demo-bucket";
    private static final String resourceAccessRoleArn =
 "arn:aws:iam::123456789012:role/resource-access-role";
    private static TranscribeStreamingAsyncClient client;

    public static void main(String args[]) {

        client = TranscribeStreamingAsyncClient.builder()
```

```
                .credentialsProvider(getCredentials())
                .httpClientBuilder(NettyNioAsyncHttpClient.builder())
                .region(REGION)
                .build();
        try {
            StartMedicalScribeStreamRequest request =
StartMedicalScribeStreamRequest.builder()
                    .languageCode(LanguageCode.EN_US.toString())
                    .mediaSampleRateHertz(SAMPLE_RATE)
                    .mediaEncoding(MediaEncoding.PCM.toString())
                    .sessionId(sessionId)
                    .build();

            MedicalScribeInputStream endSessionEvent =
MedicalScribeInputStream.sessionControlEventBuilder()
                    .type(MedicalScribeSessionControlEventType.END_OF_SESSION)
                    .build();

            CompletableFuture<Void> result = client.startMedicalScribeStream(
                    request,
                    new AudioStreamPublisher(getStreamFromMic(),
getConfigurationEvent(),endSessionEvent),
                    getMedicalScribeResponseHandler());
            result.get();
            client.close();
        } catch (Exception e) {
            System.err.println("Error occurred: " + e.getMessage());
            e.printStackTrace();
        }
    }

    private static AudioInputStream getStreamFromMic() throws LineUnavailableException
{
        // Signed PCM AudioFormat with 16kHz, 16 bit sample size, mono
        AudioFormat format = new AudioFormat(SAMPLE_RATE, 16, 1, true, false);
        DataLine.Info info = new DataLine.Info(TargetDataLine.class, format);

        if (!AudioSystem.isLineSupported(info)) {
            System.out.println("Line not supported");
            throw new LineUnavailableException("The audio system microphone line is not
supported.");
        }
        TargetDataLine line = (TargetDataLine) AudioSystem.getLine(info);
```

```
        int bufferSize = (CHUNK_SIZE_IN_BYTES / format.getFrameSize()) *
format.getFrameSize();
        line.open(format);
        line.start();

        // Create a wrapper class that can be closed when Enter is pressed
        AudioInputStream audioStream = new AudioInputStream(line);

        // Start a thread to monitor for Enter key
        System.out.println("Recording... Press Enter to stop");
        Thread monitorThread = new Thread(() -> {
            try {
                System.in.read();
                line.stop();
                line.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        });
        monitorThread.setDaemon(true);  // Set as daemon thread so it doesn't prevent
JVM shutdown
        monitorThread.start();

        return new AudioInputStream(
            new BufferedInputStream(new AudioInputStream(line)),
            format,
            AudioSystem.NOT_SPECIFIED
        );
    }

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

    private static StartMedicalScribeStreamResponseHandler
getMedicalScribeResponseHandler() {

        return StartMedicalScribeStreamResponseHandler.builder()
            .onResponse(r -> {
                System.out.println("Received Initial response");
            })
            .onError(Throwable::printStackTrace)
            .onComplete(() -> {
                System.out.println("=== All records streamed successfully ===");
```

```
            })
            .subscriber(event -> {
                if (event instanceof MedicalScribeTranscriptEvent) {
                    MedicalScribeTranscriptSegment segment =
 ((MedicalScribeTranscriptEvent) event).transcriptSegment();
                    if (segment != null && segment.content() != null && !
segment.content().isEmpty()) {
                        System.out.println(segment.content());
                    }
                }
            })
            .build();
    }

    private static DefaultConfigurationEvent getConfigurationEvent() {
        MedicalScribePostStreamAnalyticsSettings postStreamSettings =
 MedicalScribePostStreamAnalyticsSettings
                .builder()
                .clinicalNoteGenerationSettings(
                        ClinicalNoteGenerationSettings.builder()
                                .outputBucketName(bucketName)
                                .build()
                )
                .build();
        return (DefaultConfigurationEvent)
 MedicalScribeInputStream.configurationEventBuilder()
                .resourceAccessRoleArn(resourceAccessRoleArn)
                .postStreamAnalyticsSettings(postStreamSettings)
                .build();
    }

    private static class AudioStreamPublisher implements
 Publisher<MedicalScribeInputStream> {
        private final InputStream audioInputStream;
        private final MedicalScribeInputStream configEvent;
        private final MedicalScribeInputStream endSessionEvent;

        private AudioStreamPublisher(AudioInputStream audioInputStream,
                                     MedicalScribeInputStream configEvent,
                                     MedicalScribeInputStream endSessionEvent) {
            this.audioInputStream = audioInputStream;
            this.configEvent = configEvent;
            this.endSessionEvent = endSessionEvent;
        }
```

```
        @Override
        public void subscribe(Subscriber<? super MedicalScribeInputStream> subscriber)
{
            createAudioFlowable()
                    .doOnComplete(() -> {
                        try {
                            audioInputStream.close();
                        } catch (IOException e) {
                            throw new UncheckedIOException(e);
                        }
                    })
                    .subscribe(subscriber);
        }

        private Flowable<MedicalScribeInputStream> createAudioFlowable() {
            // Start with config event
            Flowable<MedicalScribeInputStream> configFlow = Flowable.just(configEvent);

            // Create audio chunk flowable
            Flowable<MedicalScribeInputStream> audioFlow = Flowable.create(emitter -> {
                byte[] buffer = new byte[CHUNK_SIZE_IN_BYTES];
                int bytesRead;

                try {
                    while (!emitter.isCancelled() && (bytesRead =
audioInputStream.read(buffer)) > 0) {
                        byte[] audioData = bytesRead < buffer.length
                                ? Arrays.copyOfRange(buffer, 0, bytesRead)
                                : buffer;

                        MedicalScribeInputStream audioEvent =
MedicalScribeInputStream.audioEventBuilder()
                                .audioChunk(SdkBytes.fromByteArray(audioData))
                                .build();

                        emitter.onNext(audioEvent);
                    }
                    emitter.onComplete();
                } catch (IOException e) {
                    emitter.onError(e);
                }
            }, BackpressureStrategy.BUFFER);
```

```
            // End with session end event
            Flowable<MedicalScribeInputStream> endFlow =
 Flowable.just(endSessionEvent);

            // Concatenate all flows
            return Flowable.concat(configFlow, audioFlow, endFlow);
        }
    }
}
```

Contoh keluaran transkripsi streaming

Setelah streaming selesai, AWS HealthScribe analisis konten aliran dan menghasilkan file JSON transkrip dan file JSON catatan klinis. Berikut adalah contoh dari setiap jenis output:

Contoh keluaran transkrip

Berikut ini adalah contoh file AWS HealthScribe transkrip dari sesi streaming.

```
{
    "Conversation": {
        "ClinicalInsights": [{
            "Attributes": [],
            "Category": "MEDICAL_CONDITION",
            "InsightId": "insightUUID1",
            "InsightType": "ClinicalEntity",
            "Spans": [{
                "BeginCharacterOffset": 12,
                "Content": "pain",
                "EndCharacterOffset": 15,
                "SegmentId": "uuid1"
            }],
            "Type": "DX_NAME"
        }, {
            "Attributes": [],
            "Category": "TEST_TREATMENT_PROCEDURE",
            "InsightId": "insightUUID2",
            "InsightType": "ClinicalEntity",
            "Spans": [{
                "BeginCharacterOffset": 4,
                "Content": "mammogram",
                "EndCharacterOffset": 12,
                "SegmentId": "uuid2"
```

```
            }],
            "Type": "TEST_NAME"
        }, {
            "Attributes": [],
            "Category": "TEST_TREATMENT_PROCEDURE",
            "InsightId": "insightUUID3",
            "InsightType": "ClinicalEntity",
            "Spans": [{
                "BeginCharacterOffset": 15,
                "Content": "pap smear",
                "EndCharacterOffset": 23,
                "SegmentId": "uuid3"
            }],
            "Type": "TEST_NAME"
        }, {
            "Attributes": [],
            "Category": "MEDICATION",
            "InsightId": "insightUUID4",
            "InsightType": "ClinicalEntity",
            "Spans": [{
                "BeginCharacterOffset": 28,
                "Content": "phentermine",
                "EndCharacterOffset": 38,
                "SegmentId": "uuid4"
            }],
            "Type": "GENERIC_NAME"
        }, {
            "Attributes": [{
                "AttributeId": "attributeUUID1",
                "Spans": [{
                    "BeginCharacterOffset": 38,
                    "Content": "high",
                    "EndCharacterOffset": 41,
                    "SegmentId": "uuid5"
                }],
                "Type": "TEST_VALUE"
            }],
            "Category": "TEST_TREATMENT_PROCEDURE",
            "InsightId": "insightUUID5",
            "InsightType": "ClinicalEntity",
            "Spans": [{
                "BeginCharacterOffset": 14,
                "Content": "weight",
                "EndCharacterOffset": 19,
```

```
                "SegmentId": "uuid6"
            }],
            "Type": "TEST_NAME"
        }, {
            "Attributes": [],
            "Category": "ANATOMY",
            "InsightId": "insightUUID6",
            "InsightType": "ClinicalEntity",
            "Spans": [{
                "BeginCharacterOffset": 60,
                "Content": "heart",
                "EndCharacterOffset": 64,
                "SegmentId": "uuid7"
            }],
            "Type": "SYSTEM_ORGAN_SITE"
        }],
        "ConversationId": "sampleConversationUUID",
        "LanguageCode": "en-US",
        "SessionId": "sampleSessionUUID",
        "TranscriptItems": [{
            "Alternatives": [{
                "Confidence": 0.7925,
                "Content": "Okay"
            }],
            "BeginAudioTime": 0.16,
            "EndAudioTime": 0.6,
            "Type": "PRONUNCIATION"
        },
        {
            "Alternatives": [{
                "Confidence": 0,
                "Content": "."
            }],
            "BeginAudioTime": 0,
            "EndAudioTime": 0,
            "Type": "PUNCTUATION"
        },
        {
            "Alternatives": [{
                "Confidence": 1,
                "Content": "Good"
            }],
            "BeginAudioTime": 0.61,
            "EndAudioTime": 0.92,
```

```
            "Type": "PRONUNCIATION"
        },
        {
            "Alternatives": [{
                "Confidence": 1,
                "Content": "afternoon"
            }],
            "BeginAudioTime": 0.92,
            "EndAudioTime": 1.54,
            "Type": "PRONUNCIATION"
        },
        {
            "Alternatives": [{
                "Confidence": 0,
                "Content": "."
            }],
            "BeginAudioTime": 0,
            "EndAudioTime": 0,
            "Type": "PUNCTUATION"
        },
        {
            "Alternatives": [{
                "Confidence": 0.9924,
                "Content": "You"
            }],
            "BeginAudioTime": 1.55,
            "EndAudioTime": 1.88,
            "Type": "PRONUNCIATION"
        },
        {
            "Alternatives": [{
                "Confidence": 1,
                "Content": "lost"
            }],
            "BeginAudioTime": 1.88,
            "EndAudioTime": 2.19,
            "Type": "PRONUNCIATION"
        },
        {
            "Alternatives": [{
                "Confidence": 1,
                "Content": "one"
            }],
            "BeginAudioTime": 2.19,
```

```
            "EndAudioTime": 2.4,
            "Type": "PRONUNCIATION"
        },
        {
            "Alternatives": [{
                "Confidence": 1,
                "Content": "lb"
            }],
            "BeginAudioTime": 2.4,
            "EndAudioTime": 2.97,
            "Type": "PRONUNCIATION"
        }
        ],
        "TranscriptSegments": [{
            "BeginAudioTime": 0.16,
            "Content": "Okay.",
            "EndAudioTime": 0.6,
            "ParticipantDetails": {
                "ParticipantRole": "CLINICIAN_0"
            },
            "SectionDetails": {
                "SectionName": "SUBJECTIVE"
            },
            "SegmentId": "uuid1"
        }, {
            "BeginAudioTime": 0.61,
            "Content": "Good afternoon.",
            "EndAudioTime": 1.54,
            "ParticipantDetails": {
                "ParticipantRole": "CLINICIAN_0"
            },
            "SectionDetails": {
                "SectionName": "OTHER"
            },
            "SegmentId": "uuid2"
        }, {
            "BeginAudioTime": 1.55,
            "Content": "You lost one lb.",
            "EndAudioTime": 2.97,
            "ParticipantDetails": {
                "ParticipantRole": "CLINICIAN_0"
            },
            "SectionDetails": {
                "SectionName": "SUBJECTIVE"
```

```
            },
            "SegmentId": "uuid3"
        }, {
            "BeginAudioTime": 2.98,
            "Content": "Yeah, I think it, uh, do you feel more energy?",
            "EndAudioTime": 6.95,
            "ParticipantDetails": {
                "ParticipantRole": "CLINICIAN_0"
            },
            "SectionDetails": {
                "SectionName": "SUBJECTIVE"
            },
            "SegmentId": "uuid4"
        }, {
            "BeginAudioTime": 6.96,
            "Content": "Yes.",
            "EndAudioTime": 7.88,
            "ParticipantDetails": {
                "ParticipantRole": "CLINICIAN_0"
            },
            "SectionDetails": {
                "SectionName": "SUBJECTIVE"
            },
            "SegmentId": "uuid5"
        }, {
            "BeginAudioTime": 7.89,
            "Content": "Uh, how about craving for the carbohydrate or sugar or fat or
 anything?",
            "EndAudioTime": 17.93,
            "ParticipantDetails": {
                "ParticipantRole": "CLINICIAN_0"
            },
            "SectionDetails": {
                "SectionName": "SUBJECTIVE"
            },
            "SegmentId": "uuid6"
        }]
    }
}
```


Contoh keluaran Dokumentasi Klinis

Berikut ini adalah contoh file wawasan dokumentasi AWS HealthScribe klinis dari sesi streaming.

```
{
  "ClinicalDocumentation": {
    "Sections": [
      {
        "SectionName": "CHIEF_COMPLAINT",
        "Summary": [
          {
            "EvidenceLinks": [
              {
                "SegmentId": "uuid1"
              },
              {
                "SegmentId": "uuid2"
              },
              {
                "SegmentId": "uuid3"
              },
              {
                "SegmentId": "uuid4"
              },
              {
                "SegmentId": "uuid5"
              },
              {
                "SegmentId": "uuid6"
              }
            ],
            "SummarizedSegment": "Weight loss."
          }
        ]
      },
      {
        "SectionName": "HISTORY_OF_PRESENT_ILLNESS",
        "Summary": [
          {
            "EvidenceLinks": [
              {
                "SegmentId": "uuid7"
              },
              {
                "SegmentId": "uuid8"
              },
              {
```

```
                "SegmentId": "uuid9"
              },
              {
                "SegmentId": "uuid10"
              }
            ],
            "SummarizedSegment": "The patient is seen today for a follow-up of weight
loss."
          },
          {
            "EvidenceLinks": [
              {
                "SegmentId": "uuid11"
              },
              {
                "SegmentId": "uuid12"
              },
              {
                "SegmentId": "uuid13"
              }
            ],
            "SummarizedSegment": "They report feeling more energy and craving
carbohydrates, sugar, and fat."
          },
          {
            "EvidenceLinks": [
              {
                "SegmentId": "uuid14"
              },
              {
                "SegmentId": "uuid15"
              },
              {
                "SegmentId": "uuid16"
              }
            ],
            "SummarizedSegment": "The patient is up to date on their mammogram and pap
smear."
          },
          {
            "EvidenceLinks": [
              {
                "SegmentId": "uuid17"
              },
```

```
            {
              "SegmentId": "uuid18"
            },
            {
              "SegmentId": "uuid19"
            },
            {
              "SegmentId": "uuid20"
            }
          ],
          "SummarizedSegment": "The patient is taking phentermine and would like to
continue."
        }
      ]
    },
    {
      "SectionName": "REVIEW_OF_SYSTEMS",
      "Summary": [
        {
          "EvidenceLinks": [
            {
              "SegmentId": "uuid21"
            },
            {
              "SegmentId": "uuid22"
            }
          ],
          "SummarizedSegment": "Patient reports intermittent headaches, occasional
chest pains but denies any recent fevers or chills."
        },
        {
          "EvidenceLinks": [
            {
              "SegmentId": "uuid23"
            },
            {
              "SegmentId": "uuid24"
            }
          ],
          "SummarizedSegment": "No recent changes in vision, hearing, or any
respiratory complaints."
        }
      ]
    },
```

```
    {
      "SectionName": "PAST_MEDICAL_HISTORY",
      "Summary": [
        {
          "EvidenceLinks": [
            {
              "SegmentId": "uuid25"
            },
            {
              "SegmentId": "uuid26"
            }
          ],
          "SummarizedSegment": "Patient has a history of hypertension and was
diagnosed with Type II diabetes 5 years ago."
        },
        {
          "EvidenceLinks": [
            {
              "SegmentId": "uuid27"
            },
            {
              "SegmentId": "uuid28"
            }
          ],
          "SummarizedSegment": "Underwent an appendectomy in the early '90s and had a
fracture in the left arm during childhood."
        }
      ]
    },
    {
      "SectionName": "ASSESSMENT",
      "Summary": [
        {
          "EvidenceLinks": [
            {
              "SegmentId": "uuid29"
            },
            {
              "SegmentId": "uuid30"
            }
          ],
          "SummarizedSegment": "Weight loss"
        }
      ]
```

```
      },
      {
        "SectionName": "PLAN",
        "Summary": [
          {
            "EvidenceLinks": [
              {
                "SegmentId": "uuid31"
              },
              {
                "SegmentId": "uuid32"
              },
              {
                "SegmentId": "uuid33"
              },
              {
                "SegmentId": "uuid34"
              }
            ],
            "SummarizedSegment": "For the condition of Weight loss: The patient was
 given a 30-day supply of phentermine and was advised to follow up in 30 days."
          }
        ]
      }
    ],
    "SessionId": "sampleSessionUUID"
  }
}
```

Enkripsi Data saat istirahat untuk AWS HealthScribe

Secara default, AWS HealthScribe menyediakan enkripsi saat istirahat untuk melindungi data pelanggan sensitif menggunakan kunci AWS HealthScribe managed AWS Key Management Service (AWS KMS). Enkripsi data saat istirahat secara default membantu mengurangi overhead operasional dan kompleksitas yang terlibat dalam melindungi data sensitif. Selain itu, ini memungkinkan Anda untuk membangun aplikasi aman yang memenuhi kepatuhan enkripsi yang ketat dan persyaratan peraturan. Saat membuat pekerjaan AWS HealthScribe transkripsi atau memulai streaming, Anda dapat menentukan kunci yang dikelola pelanggan. Ini menambahkan lapisan enkripsi kedua.

- AWS HealthScribe AWS KMS kunci terkelola — AWS HealthScribe menggunakan kunci AWS HealthScribe managed AWS Key Management Service (AWS KMS) secara default untuk

mengenkripsi file perantara secara otomatis. Anda tidak dapat menonaktifkan lapisan enkripsi ini atau memilih jenis enkripsi alternatif. Anda tidak dapat melihat, mengelola, atau menggunakan kunci, atau mengaudit penggunaannya. Namun, Anda tidak perlu mengambil tindakan apa pun atau mengubah program apa pun untuk melindungi kunci yang mengenkripsi data Anda.

- Kunci terkelola pelanggan — AWS HealthScribe mendukung penggunaan kunci terkelola pelanggan simetris yang Anda buat, miliki, dan kelola untuk menambahkan enkripsi lapisan kedua di atas enkripsi milik AWS yang ada. Karena Anda memiliki kontrol penuh atas lapisan enkripsi ini, Anda dapat melakukan tugas-tugas seperti:
  - Menetapkan dan memelihara kebijakan utama
  - Menetapkan dan memelihara IAM kebijakan dan hibah
  - Mengaktifkan dan menonaktifkan kebijakan utama
  - Memutar bahan kriptografi kunci
  - Menambahkan tanda
  - Membuat alias kunci
  - Kunci penjadwalan untuk penghapusan

  Untuk informasi selengkapnya, lihat kunci terkelola pelanggan di Panduan AWS Key Management Service Pengembang.

> **Note**
>
> AWS HealthScribe secara otomatis mengaktifkan enkripsi saat istirahat menggunakan kunci yang AWS dimiliki untuk melindungi data yang dapat diidentifikasi secara pribadi tanpa biaya. Namun, AWS KMS biaya berlaku untuk menggunakan kunci yang dikelola pelanggan. Untuk informasi selengkapnya tentang harga, silakan lihat harga AWS Key Management Service. Untuk informasi lebih lanjut tentang AWS KMS, lihat Apa itu AWS Key Management Service.

Topik

- Membuat kunci yang dikelola pelanggan
- Menentukan kunci yang dikelola pelanggan untuk AWS HealthScribe
- AWS KMS konteks enkripsi
- Memantau kunci enkripsi Anda untuk AWS HealthScribe

Membuat kunci yang dikelola pelanggan

Anda dapat membuat kunci yang dikelola pelanggan simetris dengan menggunakan AWS Management Console, atau. AWS KMS APIs Untuk membuat kunci terkelola pelanggan simetris, ikuti langkah-langkah untuk Membuat kunci terkelola pelanggan simetris di Panduan AWS Key Management Service Pengembang.

Kebijakan utama mengontrol akses ke kunci yang dikelola pelanggan Anda. Setiap kunci yang dikelola pelanggan harus memiliki persis satu kebijakan utama, yang berisi pernyataan yang menentukan siapa yang dapat menggunakan kunci dan bagaimana mereka dapat menggunakannya. Saat membuat kunci terkelola pelanggan, Anda dapat menentukan kebijakan kunci. Untuk informasi selengkapnya, lihat Mengelola akses ke kunci yang dikelola pelanggan di Panduan AWS Key Management Service Pengembang.

AWS KMS kebijakan utama untuk AWS HealthScribe

Jika Anda menggunakan kunci di akun yang sama dengan IAM peran yang Anda tentukan sebagai `ResourceAccessRole` dalam StartMedicalScribeJobatau StartMedicalScribeStreampermintaan Anda, Anda tidak perlu memperbarui Kebijakan Kunci. `DataAccessRole` Untuk menggunakan kunci terkelola pelanggan Anda di akun yang berbeda sebagai DataAccessRole (untuk pekerjaan transkripsi) atau ResourceAccessRole (untuk streaming), Anda harus mempercayai peran masing-masing dalam Kebijakan Utama untuk tindakan berikut:

- `kms:Encrypt`— Memungkinkan enkripsi menggunakan kunci yang dikelola pelanggan
- `kms:Decrypt`— Memungkinkan dekripsi menggunakan kunci yang dikelola pelanggan
- `kms:DescribeKey`— Memberikan detail kunci yang dikelola pelanggan untuk memungkinkan AWS HealthScribe memvalidasi kunci

Berikut ini adalah contoh kebijakan utama yang dapat Anda gunakan untuk memberikan izin ResourceAccessRole lintas akun Anda untuk menggunakan kunci yang dikelola pelanggan Anda untuk AWS HealthScribe streaming. Untuk menggunakan kebijakan ini untuk pekerjaan transkripsi, `Principal` perbarui penggunaan DataAccessRole ARN, lalu hapus atau ubah konteks enkripsi.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid": "Allow access for key administrators",
```

```
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:iam::123456789012:root"
         },
         "Action" : [
           "kms:*"
         ],
         "Resource": "*"
      },
      {
         "Sid":"Allow access to the ResourceAccessRole for StartMedicalScribeStream",
         "Effect":"Allow",
         "Principal":{
            "AWS": "arn:aws:iam::123456789012:role/ResourceAccessRole"
         },
         "Action":[
            "kms:Encrypt",
            "kms:Decrypt",
            "kms:GenerateDataKey*"
         ]
         "Resource":"*",
         "Condition": {
            "StringEquals": {
                "EncryptionContext":[
                    "aws:us-east-1:transcribe:medical-scribe:session-
id": "1234abcd-12ab-34cd-56ef-123456SAMPLE"
                ]
            }
         }
      },
      {
         "Sid":"Allow access to the ResourceAccessRole for DescribeKey",
         "Effect":"Allow",
         "Principal":{
             "AWS": "arn:aws:iam::123456789012:role/ResourceAccessRole"
         },
         "Action": "kms:DescribeKey",
         "Resource":"*"
     }
   ]
}
```

Izin kebijakan IAM untuk peran akses

Kebijakan IAM yang dilampirkan pada DataAccessRole atau ResourceAccessRole harus memberikan izin untuk melakukan AWS KMS tindakan yang diperlukan, terlepas dari apakah kunci dan peran yang dikelola pelanggan berada di akun yang sama atau berbeda. Juga, kebijakan kepercayaan peran harus memberikan AWS HealthScribe izin untuk mengambil peran tersebut.

Contoh kebijakan IAM berikut menunjukkan cara memberikan ResourceAccessRole izin untuk AWS HealthScribe streaming. Untuk menggunakan kebijakan ini untuk pekerjaan transkripsi, ganti `transcribe.streaming.amazonaws.com` dengan `transcribe.amazonaws.com` dan hapus atau ubah konteks enkripsi.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:GenerateDataKey*"
            ],
            "Resource": "arn:aws:kms:us-west-2:123456789012:key/Key_ID",
            "Effect": "Allow",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": "transcribe.streaming.amazonaws.com",
                    "EncryptionContext":[
                        "aws:us-east-1:transcribe:medical-scribe:session-id":
 "1234abcd-12ab-34cd-56ef-123456SAMPLE"
                    ]
                }
            }
        },
        {
            "Action": [
                "kms:DescribeKey"
            ],
            "Resource": "arn:aws:kms:us-west-2:123456789012:key/Key_ID",
            "Effect": "Allow",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": "transcribe.streaming.amazonaws.com"
```

```
                }
            }
        }
    ]
}
```

Berikut ini adalah contoh kebijakan kepercayaan untuk ResourceAccessRole. Untuk DataAccessRole, ganti `transcribe.streaming.amazonaws.com` dengan`transcribe.amazonaws.com`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "transcribe.streaming.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                    },
                "StringLike": {
                    "aws:SourceArn": "arn:aws:transcribe:us-west-2:123456789012:*"
                }
            }
        }
    ]
}
```

Untuk informasi selengkapnya tentang menentukan izin dalam kebijakan atau akses kunci pemecahan masalah, lihat Panduan Pengembang. AWS Key Management Service

Menentukan kunci yang dikelola pelanggan untuk AWS HealthScribe

Anda dapat menentukan kunci terkelola pelanggan sebagai enkripsi lapisan kedua untuk pekerjaan transkripsi atau streaming.

- Untuk pekerjaan transkripsi, Anda menentukan kunci Anda di OutputEncryptionKMSKeyId operasi StartMedicalScribeJobAPI Anda.

- Untuk streaming, Anda menentukan kunci di MedicalScribeEncryptionSettingsdalam MedicalScribeConfigurationEvent.

AWS KMS konteks enkripsi

AWS KMS konteks enkripsi adalah peta teks biasa, pasangan kunci non-rahasia: nilai. Peta ini mewakili data tambahan yang diautentikasi, yang dikenal sebagai pasangan konteks enkripsi, yang menyediakan lapisan keamanan tambahan untuk data Anda. AWS HealthScribe memerlukan kunci enkripsi simetris untuk mengenkripsi AWS HealthScribe output ke dalam bucket yang ditentukan pelanggan Amazon S3 . Untuk mempelajari lebih lanjut, lihat Kunci asimetris di AWS KMS.

Saat membuat pasangan konteks enkripsi Anda, jangan sertakan informasi sensitif. Konteks enkripsi bukanlah rahasia — ini terlihat dalam teks biasa di dalam CloudTrail log Anda (sehingga Anda dapat menggunakannya untuk mengidentifikasi dan mengkategorikan operasi kriptografi Anda). Pasangan konteks enkripsi Anda dapat menyertakan karakter khusus, seperti garis bawah (_), tanda hubung (), garis miring (,-) dan / titik dua (\). :

> **Tip**
>
> Ini dapat berguna untuk menghubungkan nilai-nilai dalam pasangan konteks enkripsi Anda dengan data yang dienkripsi. Meskipun tidak diperlukan, kami sarankan Anda menggunakan metadata non-sensitif yang terkait dengan konten terenkripsi Anda, seperti nama file, nilai header, atau bidang database yang tidak terenkripsi.
>
> Untuk menggunakan enkripsi keluaran dengan API, atur parameter KMSEncryptionKonteks dalam StartMedicalScribeJoboperasi. Untuk menyediakan konteks enkripsi untuk operasi enkripsi output, parameter OutputEncryptionKMSKeyId harus mereferensikan ID AWS KMS kunci simetris.
>
> Untuk streaming, Anda menentukan pasangan nilai kunci untuk `KmsEncryptionContext` di MedicalScribeEncryptionSettingsdalam MedicalScribeConfigurationEvent.
>
> Anda dapat menggunakan kunci AWS KMS kondisi dengan IAM kebijakan untuk mengontrol akses ke AWS KMS kunci enkripsi simetris berdasarkan konteks enkripsi yang digunakan dalam permintaan operasi kriptografi. Untuk contoh kebijakan konteks enkripsi, lihat kebijakan konteks AWS KMS enkripsi.
>
> Menggunakan konteks enkripsi adalah opsional, tetapi disarankan. Untuk informasi selengkapnya, lihat Konteks enkripsi.

AWS HealthScribe konteks enkripsi

AWS HealthScribe menggunakan konteks enkripsi yang sama di semua operasi AWS Key Management Service kriptografi. Konteks enkripsi adalah peta String to String yang dapat disesuaikan dengan apa pun yang Anda inginkan.

```
"encryptionContext": {
   "ECKey": "ECValue"
   ...
}
```

Untuk AWS HealthScribe aliran, berikut ini adalah konteks enkripsi yang dihasilkan layanan default. Ini menerapkan konteks ini di atas konteks enkripsi apa pun yang Anda berikan.

```
"encryptionContext": {
  "aws:<region>:transcribe:medical-scribe:session-id":
 "1234abcd-12ab-34cd-56ef-123456SAMPLE"
}
```

Untuk pekerjaan AWS HealthScribe transkripsi, berikut ini adalah konteks enkripsi yang dihasilkan layanan default. Ini menerapkan konteks ini di atas konteks enkripsi apa pun yang Anda berikan.

```
"encryptionContext": {
  "aws:<region>:transcribe:medical-scribe:job-name": "<job-name>",
  "aws:<region>:transcribe:medical-scribe:start-time-epoch-ms": "<job-start-time>"
}
```

Jika Anda tidak memberikan konteks enkripsi apa pun, hanya konteks enkripsi yang dihasilkan layanan yang akan digunakan untuk semua operasi AWS KMS kriptografi.

Pemantauan AWS HealthScribe dengan konteks enkripsi

Saat Anda menggunakan kunci terkelola pelanggan simetris untuk mengenkripsi data Anda saat istirahat AWS HealthScribe, Anda juga dapat menggunakan konteks enkripsi dalam catatan audit dan log untuk mengidentifikasi bagaimana kunci yang dikelola pelanggan digunakan. Konteks enkripsi juga muncul di log yang dihasilkan oleh AWS CloudTrail atau CloudWatch Log.

Menggunakan konteks enkripsi untuk mengontrol akses ke kunci terkelola pelanggan Anda

Anda dapat menggunakan konteks enkripsi dalam kebijakan utama dan kebijakan IAM sebagai kondisi untuk mengontrol akses ke kunci terkelola pelanggan simetris Anda.

Berikut ini adalah contoh pernyataan kebijakan kunci untuk memberikan akses ke kunci yang dikelola pelanggan untuk konteks enkripsi tertentu. Kondisi dalam pernyataan kebijakan ini mengharuskan penggunaan kunci KMS memiliki batasan konteks enkripsi yang menentukan konteks enkripsi.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid":"Allow access to the ResourceAccessRole for
 StartMedicalScribeStream",
            "Effect":"Allow",
            "Principal":{
                "AWS": "arn:aws:iam::123456789012:role/ResourceAccessRole"
            },
            "Action":[
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:GenerateDataKey*"
            ],
            "Resource":"arn:aws:kms:us-west-2:123456789012:key/Key_ID",
            "Condition": {
                "StringEquals": {
                    // below is the service generated encryption context example
                    "kms:EncryptionContext:aws:us-east-1:transcribe:medical-
scribe:session-id":"1234abcd-12ab-34cd-56ef-123456SAMPLE",
                    // plus any encryption context that you specify in the request
                    "kms:EncryptionContext:${ECKey}": "${ECValue}"
                }
            }
        },
        {
            "Sid":"Allow access to the ResourceAccessRole for DescribeKey",
            "Effect":"Allow",
            "Principal":{
                "AWS": "arn:aws:iam::123456789012:role/ResourceAccessRole"
            },
            "Action": "kms:DescribeKey",
            "Resource":"arn:aws:kms:us-west-2:123456789012:key/Key_ID"
        }
}
```

Memantau kunci enkripsi Anda untuk AWS HealthScribe

Saat Anda menggunakan kunci yang dikelola AWS Key Management Service pelanggan AWS HealthScribe, Anda dapat menggunakan AWS CloudTrail atau CloudWatch mencatat untuk melacak permintaan yang AWS HealthScribe dikirim AWS KMS.

Contoh berikut adalah CloudTrail Enkripsi dan Dekripsi peristiwa yang dapat Anda gunakan yang memungkinkan Anda memantau bagaimana AWS HealthScribe penggunaan kunci yang dikelola pelanggan Anda.

Enkripsi


```
{
   "eventVersion":"1.09",
   "userIdentity":{
      "type":"AssumedRole",
      "principalId":"AROAIGDTESTANDEXAMPLE:Sampleuser01",
      "arn":"arn:aws:sts::123456789012:assumed-role/Admin/Sampleuser01",
      "accountId":"123456789012",
      "accessKeyId":"AKIAIOSFODNN7EXAMPLE3",
      "sessionContext":{
         "sessionIssuer":{
            "type":"Role",
            "principalId":"AROAIGDTESTANDEXAMPLE:Sampleuser01",
            "arn":"arn:aws:sts::123456789012:assumed-role/Admin/Sampleuser01",
            "accountId":"123456789012",
            "userName":"Admin"
         },
         "attributes":{
            "creationDate":"2024-08-16T01:10:05Z",
            "mfaAuthenticated":"false"
         }
      },
      "invokedBy":"transcribe.streaming.amazonaws.com"
   },
   "eventTime":"2024-08-16T01:10:05Z",
   "eventSource":"kms.amazonaws.com",
   "eventName":"Encrypt",
   "awsRegion":"us-east-1",
   "sourceIPAddress":"transcribe.streaming.amazonaws.com",
   "userAgent":"transcribe.streaming.amazonaws.com",
   "requestParameters":{
      "encryptionContext":{
```

```
         "aws:us-east-1:transcribe:medical-scribe:session-
id":"1234abcd-12ab-34cd-56ef-123456SAMPLE"
      },
      "encryptionAlgorithm":"SYMMETRIC_DEFAULT",
      "keyId":"1234abcd-12ab-34cd-56ef-1234567890ab"
   },
   "responseElements":null,
   "requestID":"cbe0ac33-8cca-49e5-9bb5-dc2b8dfcb389",
   "eventID":"1b9fedde-aa96-48cc-9dd9-a2cce2964b3c",
   "readOnly":true,
   "resources":[
      {
         "accountId":"123456789012",
         "type":"AWS::KMS::Key",
         "ARN":"arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
      }
   ],
   "eventType":"AwsApiCall",
   "managementEvent":true,
   "recipientAccountId":"123456789012",
   "eventCategory":"Management"
}
```

Dekripsi

```
{
   "eventVersion":"1.09",
   "userIdentity":{
      "type":"AssumedRole",
      "principalId":"AROAIGDTESTANDEXAMPLE:Sampleuser01",
      "arn":"arn:aws:sts::123456789012:assumed-role/Admin/Sampleuser01",
      "accountId":"123456789012",
      "accessKeyId":"AKIAIOSFODNN7EXAMPLE3",
      "sessionContext":{
         "sessionIssuer":{
            "type":"Role",
            "principalId":"AROAIGDTESTANDEXAMPLE:Sampleuser01",
            "arn":"arn:aws:sts::123456789012:assumed-role/Admin/Sampleuser01",
            "accountId":"123456789012",
            "userName":"Admin"
```

```
         },
         "attributes":{
            "creationDate":"2024-08-16T20:47:04Z",
            "mfaAuthenticated":"false"
         }
      },
      "invokedBy":"transcribe.streaming.amazonaws.com"
   },
   "eventTime":"2024-08-16T20:47:04Z",
   "eventSource":"kms.amazonaws.com",
   "eventName":"Decrypt",
   "awsRegion":"us-east-1",
   "sourceIPAddress":"transcribe.streaming.amazonaws.com",
   "userAgent":"transcribe.streaming.amazonaws.com",
   "requestParameters":{
      "keyId":"mrk-de27f019178f4fbf86512ab03ba860be",
      "encryptionAlgorithm":"SYMMETRIC_DEFAULT",
      "encryptionContext":{
         "aws:us-east-1:transcribe:medical-scribe:session-
id":"1234abcd-12ab-34cd-56ef-123456SAMPLE"
      }
   },
   "responseElements":null,
   "requestID":"8b7fb865-48be-4e03-ac3d-e7bee3ba30a1",
   "eventID":"68b7a263-d410-4701-9e2b-20c196628966",
   "readOnly":true,
   "resources":[
      {
         "accountId":"123456789012",
         "type":"AWS::KMS::Key",
         "ARN":"arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
      }
   ],
   "eventType":"AwsApiCall",
   "managementEvent":true,
   "recipientAccountId":"123456789012",
   "eventCategory":"Management"
}
```

Riwayat dokumen untuk Amazon Transcribe

- Pembaruan dokumentasi terbaru: 13 November 2023

Tabel berikut menjelaskan perubahan penting dalam setiap rilis Amazon Transcribe. Untuk notifikasi tentang pembaruan dokumentasi ini, Anda dapat berlangganan ke umpan RSS.

Perubahan	Deskripsi	Tanggal
Fitur baru	AWS HealthScribe sekarang mendukung template baru untuk ringkasan catatan klinis: BIRP, SIRP, DAP, BH_SOAP, dan PH_SOAP. Untuk informasi lebih lanjut, lihat file Dokumentasi AWS HealthScribe Klinis.	30 Mei 2025
Fitur baru	AWS HealthScribe sekarang mendukung template GIRPP untuk ringkasan catatan klinis. Untuk informasi lebih lanjut, lihat file Dokumentasi AWS HealthScribe Klinis.	Februari 4, 2025
Fitur baru	AWS HealthScribe sekarang mendukung streaming untuk menyalin percakapan medis secara real time.	Januari 29, 2025
Pembaruan Bagian	Memperbarui dukungan bahasa AR-sa untuk 8000 frekuensi.	Januari 16, 2025
Pembaruan Bagian	Memperbarui dukungan bahasa AR-sa untuk 8000 frekuensi.	Januari 16, 2025

Pembaruan Bagian	Memperbarui bagian bahasa yang didukung dengan kode bahasa yang tidak didukung yang tidak kami dukung di wilayah AWS GovCloud (US) (AS-Barat, us-gov-west -1), AWS GovCloud (US) (AS-Timur, us-gov-east -1), atau Afrika (Cape Town, af-south-1).	Januari 14, 2025
Pembaruan Bagian	Perluas output json Transcribe Batch dengan bidang baru yang disebut “audio_segment”.	Juli 15, 2024
Pembaruan Fitur	Perbarui Speaker Maks dalam diarisasi menjadi 30, bukan 10.	10 Mei 2024
Pembaruan Bagian	Pembaruan untuk ringkasan panggilan generatif dan menambahkan detail keluaran kesalahan.	April 30, 2024
Pembaruan Bagian	Pembaruan pada kolom kosakata khusus - IPA dan. SoundsLike	April 30, 2024
Pembaruan Fitur	Amazon Transcribe Call Analytics sekarang mendukung ringkasan panggilan generatif.	November 29, 2023
Pembaruan Bagian	Perbarui redaksi PII baru dan format keluaran Identifikasi Bahasa.	13 November 2023

Pembaruan Fitur	Diarisasi sekarang dapat dikombinasikan dengan identifikasi saluran.	6 Maret 2023
Pembaruan Fitur	Identifikasi saluran sekarang dapat dikombinasikan dengan diarisasi.	6 Maret 2023
Pembaruan Bagian	IAM praktik terbaik telah diperbarui.	13 Februari 2023
Bahasa baru	Amazon Transcribe sekarang mendukung Vietnam dan Swedia.	6 Desember 2022
Fitur baru	Amazon Transcribe sekarang mendukung Analytics Panggilan real-time.	28 November 2022
Pembaruan Fitur	Streaming redaksi dan identifikasi sekarang tersedia dengan bahasa Hindi dan Thailand.	11 November 2022
Pembaruan Bagian	Kategori PII baru tersedia untuk streaming redaksi dan identifikasi.	14 September 2022
Pembaruan Bagian	Bagian model bahasa khusus telah direvisi.	Juni 18, 2022
Pembaruan Bagian	Identifikasi bahasa Batch sekarang dapat mengidentifikasi beberapa bahasa per file audio.	31 Mei 2022
Panduan Pembaruan	Referensi Amazon Transcribe API sekarang menjadi panduan mandiri.	1 April 2022

Bab baru	Tabel perbandingan baru untuk Amazon Transcribe, Amazon Transcribe Medical, dan Amazon Transcribe Call Analytics disertakan.	Maret 21, 2022
Bab baru	Bab contoh kode SDK baru disertakan.	Maret 21, 2022
Pembaruan fitur	Call Analytics sekarang menyediakan ringkasan panggilan.	Maret 21, 2022
Pembaruan Bab	Bab pengantar sekarang menampilkan kasus Amazon Transcribe penggunaan.	Maret 21, 2022
Pembaruan Bab	Bab Memulai telah diperbarui agar spesifik metode.	Maret 21, 2022
Pembaruan Bab	Bab Streaming telah diperbaru i dan direstrukturisasi.	Maret 21, 2022
Pembaruan fitur	Identifikasi bahasa sekarang mendukung kosakata khusus dan filter kosakata khusus dengan transkripsi streaming.	11 Maret 2022
Acara baru	Ada jenis acara baru: Acara kosakata.	Februari 7, 2022
Pembaruan bagian	Pembaruan telah dilakukan ke bagian kosakata khusus.	20 Januari 2022
Fitur baru	Identifikasi bahasa sekarang dapat digunakan dengan transkripsi streaming.	23 November 2021

Fitur baru	Identifikasi bahasa sekarang dapat digunakan dengan model bahasa khusus, kosakata khusus, pemfilteran kosakata, dan redaksi konten.	29 Oktober 2021
Fitur baru	Amazon Transcribe sekarang mendukung model bahasa khusus dengan transkripsi streaming.	20 Oktober 2021
Fitur baru	Amazon Transcribe sekarang dapat menghasilkan subtitle untuk file video Anda.	September 16, 2021
Fitur baru	Amazon Transcribe sekarang mendukung redaksi PII dan identifikasi untuk streaming.	14 September 2021
Fitur baru	Amazon Transcribe sekarang mendukung konteks AWS KMS enkripsi untuk tingkat keamanan tambahan untuk Akun AWS sumber daya Anda.	10 September 2021
Bahasa baru	Amazon Transcribe sekarang mendukung Afrikaans, Denmark, Mandarin Mandarin (Tradisional), Thailand, Inggris Selandia Baru, dan Inggris Afrika Selatan.	26 Agustus 2021
Fitur baru	Amazon Transcribe sekarang mendukung penandaan sumber daya.	Agustus 24, 2021

Fitur baru	Amazon Transcribe sekarang mendukung Call Analytics untuk pekerjaan transkripsi batch.	4 Agustus 2021
Fitur baru	Amazon Transcribe sekarang mendukung penggunaan kosakata khusus dengan model bahasa kustom batch.	12 Mei 2021
Fitur baru	Amazon Transcribe sekarang mendukung stabilisasi hasil sebagian untuk transkripsi streaming.	11 Mei 2021
Fitur baru	Amazon Transcribe sekarang mendukung bahasa Inggris Australia, Inggris Inggris, Hindi, dan Spanyol AS untuk model bahasa khusus.	19 Maret 2021
Fitur baru	Amazon Transcribe sekarang mendukung OGG/OPUS dan codec FLAC untuk streaming transkripsi audio.	24 November 2020
Bahasa baru	Amazon Transcribe menambahkan dukungan untuk Italia dan Jerman untuk streaming transkripsi audio.	4 November 2020
Wilayah AWS ekspansi	Amazon Transcribe sekarang tersedia di Frankfurt (eu-central-1) dan London (eu-west-2).	4 November 2020

Fitur baru	Amazon Transcribe menambahkan dukungan untuk titik akhir VPC antarmuka dalam transkripsi batch.	9 Oktober 2020
Fitur baru	Amazon Transcribe menambahkan dukungan untuk identifikasi saluran dalam streaming.	17 September 2020
Fitur baru	Amazon Transcribe menambahkan dukungan untuk identifikasi bahasa otomatis dalam transkripsi batch.	15 September 2020
Fitur baru	Amazon Transcribe menambahkan dukungan untuk partisi speaker dalam streaming.	19 Agustus 2020
Fitur baru	Amazon Transcribe menambahkan dukungan untuk model bahasa khusus.	Agustus 5, 2020
Fitur baru	Amazon Transcribe menambahkan dukungan untuk titik akhir VPC antarmuka dalam streaming.	26 Juni 2020
Fitur baru	Amazon Transcribe menambahkan dukungan untuk pemfilteran kosakata dalam streaming.	20 Mei 2020

Fitur baru	Amazon Transcribe menambahkan dukungan untuk secara otomatis menyunting informasi yang dapat diidentifikasi secara pribadi.	26 Februari 2020
Fitur baru	Amazon Transcribe menambahkan dukungan untuk membuat kosakata khusus kata-kata untuk difilter dari transkripsi.	20 Desember 2019
Fitur baru	Amazon Transcribe menambahkan dukungan untuk pekerjaan transkripsi antrian.	19 Desember 2019
Bahasa baru	Amazon Transcribe menambahkan dukungan untuk Teluk Arab, Ibrani, Jepang, Melayu, Swiss Jerman, Telugu, dan Turki.	21 November 2019
Wilayah AWS ekspansi	Amazon Transcribe sekarang tersedia di Asia Pasifik (Tokyo) (ap-northeast-1).	21 November 2019
Fitur baru	Amazon Transcribe menambahkan dukungan untuk transkripsi alternatif.	20 November 2019
Bahasa baru	Amazon Transcribe menambahkan dukungan untuk bahasa Belanda, Farsi, Indonesia, Inggris Irlandia, Portugis, Inggris Skotlandia, Tamil, dan Inggris Welsh.	12 November 2019

Bahasa baru	Amazon Transcribe sekarang mendukung transkripsi streaming untuk Bahasa Inggris Australia (en-AU).	Oktober 25, 2019
Wilayah AWS ekspansi	Amazon Transcribe sekarang tersedia di Tiongkok (Beijing) (cn-utara-1) dan Tiongkok (Ningxia) (cn-barat laut-1).	9 Oktober 2019
Fitur baru	Amazon Transcribe memungkinkan Anda menyediakan sendiri KMS key untuk mengenkripsi file keluaran transkripsi Anda. Untuk informasi selengkapnya, lihat parameter OutputEncryptionKMSKeyId StartStreamTranscriptionAPI.	24 September 2019
Bahasa baru	Amazon Transcribe menambahkan dukungan untuk China (Mandarin), Sederhana, Tiongkok Daratan dan Rusia.	Agustus 23, 2019
Fitur baru	Amazon Transcribe menambahkan dukungan untuk streaming transkripsi audio menggunakan WebSocket protokol.	19 Juli 2019
Fitur baru	AWS CloudTrail sekarang merekam peristiwa untuk StartStreamTranscriptionAPI.	19 Juli 2019

Wilayah AWS ekspansi	Amazon Transcribe sekarang tersedia di AS Barat (California N.) (us-west-1).	27 Juni 2019
Bahasa baru	Amazon Transcribe menambahkan dukungan untuk Bahasa Arab Standar Modern.	28 Mei 2019
Fitur baru	Amazon Transcribe sekarang mentranskripsikan kata-kata numerik menjadi angka untuk bahasa Inggris AS. Misalnya, “empat puluh dua” ditranskr ipsikan sebagai “42".	23 Mei 2019
Bahasa baru	Amazon Transcribe menambahkan dukungan untuk bahasa Hindi dan India Inggris.	15 Mei 2019
SDK baru	AWS SDK for C++ Amazon Transcribe sekarang mendukung.	8 Mei 2019
Bahasa baru	Amazon Transcribe menambahkan dukungan untuk Spanyol.	19 April 2019
Wilayah AWS ekspansi	Amazon Transcribe sekarang tersedia di UE (Frankfurt) (eu-central-1) dan Asia Pasifik (Seoul) (ap-northeast-2).	18 April 2019

Bahasa baru	Amazon Transcribe menambahkan dukungan untuk streaming transkripsi dalam bahasa Inggris Inggris, Prancis, dan Prancis Kanada.	5 April 2019
Fitur baru	AWS SDK for Ruby V3 sekarang mendukung Amazon Transcribe	25 Maret 2019
Fitur baru	Amazon Transcribe memungkinkan kosakata khusus, yang merupakan daftar kata-kata tertentu yang Amazon Transcribe ingin Anda kenali dalam input audio Anda.	25 Maret 2019
Bahasa baru	Amazon Transcribe menambahkan dukungan untuk Jerman dan Korea.	22 Maret 2019
Bahasa baru	Amazon Transcribe sekarang mendukung transkripsi streaming untuk US Spanish (es-US).	7 Februari 2019
Wilayah AWS ekspansi	Amazon Transcribe sekarang tersedia di Amerika Selatan (São Paulo) (sa-east-1).	7 Februari 2019
Wilayah AWS ekspansi	Amazon Transcribe sekarang tersedia di Asia Pasifik (Mumbai) (ap-south-1), Asia Pasifik (Singapura) (ap-south east-1), EU (London) (eu-west-2), dan EU (Paris) (eu-west3).	24 Januari 2019

Bahasa baru	Amazon Transcribe menambahkan dukungan untuk Portugis Prancis, Italia, dan Brasil.	20 Desember 2018
Fitur baru	Amazon Transcribe sekarang mendukung transkripsi aliran audio.	19 November 2018
Bahasa baru	Amazon Transcribe menambahkan dukungan untuk Bahasa Inggris Australia , Inggris Inggris, dan Prancis Kanada.	15 November 2018
Wilayah AWS ekspansi	Amazon Transcribe sekarang tersedia di Kanada (Tengah) (ca-central-1) dan Asia Pasifik (Sydney) (ap-southeast-2).	17 Juli 2018
Fitur baru	Anda sekarang dapat menentukan lokasi Anda sendiri untuk menyimpan output dari pekerjaan transkrip si.	11 Juli 2018
Fitur baru	Ditambahkan AWS CloudTrai l dan Amazon CloudWatch Events integrasi.	28 Juni 2018
Fitur baru	Amazon Transcribe menambahkan dukungan untuk kosakata khusus.	4 April 2018
Panduan baru	Panduan ini adalah perilisan pertama dari Panduan Developer Amazon Transcribe .	29 November 2017

AWS Glosarium

Untuk AWS terminologi terbaru, lihat AWS glosarium di Referensi.Glosarium AWS

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.