



개발자 가이드

Amazon Transcribe



Amazon Transcribe: 개발자 가이드

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

Amazon의 상표 및 트레이드 드레스는 Amazon 외 제품 또는 서비스와 함께, Amazon 브랜드 이미지를 떨어뜨리거나 고객에게 혼동을 일으킬 수 있는 방식으로 사용할 수 없습니다. Amazon이 소유하지 않은 기타 모든 상표는 Amazon과 제휴 관계이거나 관련이 있거나 후원 관계와 관계없이 해당 소유자의 자산입니다.

Table of Contents

Amazon Transcribe란 무엇인가요?	1
Amazon Transcribe 및 HIPAA 자격	1
가격 책정	2
리전 가용성 및 할당량	2
사용 가능한 기능	4
지원되는 언어	6
지원되는 프로그래밍 언어	14
문자 집합	16
압하스어	19
아프리카나스어	21
아랍어	22
아스투리아스어	23
아제르바이잔어	24
아르메니아어	24
바쉬르어	26
바스크어	28
벨라루스어	29
벵골어	30
보스니아어	32
불가리아어	32
카탈루냐어	33
소라니어	34
중국어, 간체	35
중국어, 한자	36
중국어, 광둥어	37
크로아티아어	37
체코어	38
덴마크어	39
네덜란드어	39
영어	40
에스토니아어	41
페르시아어	41
핀란드어	42
프랑스어	43

갈리시아어	44
조지아어	44
독일어	46
그리스어	46
구자라트어	47
하우사어	49
히브리어	50
힌디어	51
헝가리어	53
아이슬란드어	54
인도네시아어	54
이탈리아어	55
일본어	56
커바일어	56
칸나다어	56
카자흐어	58
키냐르완다어	60
한국어	61
키르기스어	61
라트비아어	63
리투아니아어	64
간다어	64
마케도니아어	65
말레이어	67
말라얄람어	68
몰타어	70
마라티어	70
메도우 마리	72
몽골어	75
노르웨이어(보크몰)	77
오디아/오리야어	77
파슈토어	79
폴란드어	81
포르투갈어	82
펀자브어	83
루마니아어	85

러시아어	85
세르비아어	86
싱할라어	89
슬로바키아어	91
슬로베니아어	92
소말리아어	92
스페인어	93
순다어	94
스와힐리어	94
스웨덴어	95
타갈로그어/필리핀어	95
타밀어	96
타타르어	97
텔루구어	100
태국어	101
터키어	103
우크라이나어	104
위구르어	105
우즈베크어	108
베트남어	109
웨일스어	113
월로프어	114
줄루어	115
작동 방법	116
데이터 입력 및 출력	116
미디어 형식	117
오디오 채널	118
샘플 속도	118
출력	119
번호 트랜스크립션	122
시작하기	123
에 가입 AWS 계정	123
AWS CLI 및 SDKs 설치	124
IAM 자격 증명 구성	124
Amazon S3 버킷 생성	125
IAM 정책 생성	125

로 트랜스크립션 AWS Management Console	128
로 트랜스크립션 AWS CLI	136
새 트랜스크립션 작업 시작	137
트랜스크립션 작업의 상태 가져오기	138
트랜스크립션 작업 나열	139
트랜스크립션 작업 삭제	140
AWS SDKs	140
AWS SDKs 작업	152
HTTP 또는 WebSocket을 통한 트랜스크립션	153
스트리밍 트랜스크립션	156
모범 사례	157
스트리밍 및 부분 결과	158
부분 결과 안정화	159
스트리밍 트랜스크립션 설정	163
이벤트 스트림 인코딩	176
데이터 프레임	179
작업 대기열	180
작업 대기열 활성화	180
리소스에 태그 지정	184
태그 기반 액세스 제어	185
Amazon Transcribe 리소스에 태그 추가	185
화자 분할(분할)	190
배치 트랜스크립션의 화자 분할	190
스트리밍 트랜스크립션의 화자 분할	194
출력 예시	196
다중 채널 오디오 트랜스크립션	203
배치 트랜스크립션에서 채널 식별 사용	204
스트리밍 트랜스크립션에서 채널 식별 사용	207
출력 예시	209
식별 언어	218
배치 언어 식별	218
다국어 오디오의 언어 식별	219
언어 식별 정확도 향상	220
언어 식별을 다른 Amazon Transcribe 기능과 결합	221
배치 트랜스크립션 작업을 통한 언어 식별 사용	221
스트리밍 언어 식별	227

다국어 오디오의 언어 식별	229
스트리밍 미디어를 통한 언어 식별 사용	229
대체 트랜스크립션	235
대체 트랜스크립션 요청	237
트랜스크립션 정확도 향상	242
사용자 지정 어휘	243
사용자 지정 어휘 테이블과 목록 비교	244
테이블을 사용하여 사용자 지정 어휘 생성	245
목록을 사용하여 사용자 지정 어휘 생성	254
사용자 지정 어휘 사용	257
사용자 지정 언어 모델	264
데이터 소스	264
훈련 데이터와 미세 조정 데이터 비교	265
사용자 지정 언어 모델 생성	266
사용자 지정 언어 모델 사용	271
단어 필터링	279
어휘 필터 생성	280
사용자 지정 어휘 필터 생성	281
사용자 지정 어휘 필터 사용	285
배치 트랜스크립션 작업에서의 어휘 필터 사용	257
스트리밍 트랜스크립션에서의 사용자 지정 어휘 필터 사용	261
유해 언어 감지	293
유해 언어 감지 사용	294
배치 트랜스크립션에 유해 언어 감지 기능 사용	294
출력 예시	298
트랜스크립트 편집	301
배치 작업의 PII 편집	301
실시간 스트림에서 PII 편집 또는 식별	307
출력 예시	313
편집된 출력(배치) 예시	313
편집된 스트리밍 출력 예시	316
PII 식별 출력 예시	317
자막 생성	320
자막 파일 생성	321
콜센터 오디오 분석	324
일반 사용 사례	324

고려 사항 및 추가 정보	326
리전 가용성 및 할당량	326
통화 후 분석	327
통화 후 인사이트	328
범주 생성	330
트랜스크립션 시작	340
통화 후 분석 출력	348
생성형 통화 요약 활성화	361
실시간 통화 분석	365
실시간 인사이트	366
범주 생성	369
실시간 트랜스크립션을 통한 통화 후 분석	376
트랜스크립션 시작	383
실시간 통화 분석 출력	391
Amazon Chime 통화 트랜스크립션	397
코드 예제	399
기본 사항	400
작업	400
시나리오	469
Amazon Transcribe 스트리밍 앱 구축	470
텍스트를 스피치로, 다시 스피치에서 텍스트로 변환	470
사용자 지정 어휘 생성 및 세부 조정	471
오디오 전사 및 작업 데이터 가져오기	481
보안	493
자격 증명 및 액세스 관리	494
대상	494
ID를 통한 인증	494
정책을 사용하여 액세스 관리	496
Amazon Transcribe 에서 IAM을 사용하는 방법	497
혼동된 대리자 방지	502
ID 기반 정책 예시	503
문제 해결	513
데이터 보호	514
인터넷워크 트래픽 개인 정보 보호	515
데이터 암호화	515
서비스 개선을 위한 데이터 사용 거부	518

모니터링 Amazon Transcribe	518
를 사용한 모니터링 CloudWatch	519
Amazon Transcribe 를 사용한 모니터링 CloudTrail	520
Amazon Transcribe과 함께 Amazon EventBridge 사용	524
규정 준수 확인	532
복원성	532
인프라 보안	532
취약성 분석 및 관리	533
VPC 엔드포인트(AWS PrivateLink)	533
공유 서브넷	535
보안 모범 사례	535
Amazon Transcribe Medical	537
리전 가용성 및 할당량	538
의료 전문 분야	539
의학 용어 및 측정치 트랜스크립션	540
번호 트랜스크립션	543
의료 대화 트랜스크립션	545
오디오 파일 트랜스크립션	546
실시간 스트림 트랜스크립션	551
화자 파티셔닝 활성화	554
다중 채널 오디오 트랜스크립션	563
의료 구술 트랜스크립션	571
오디오 파일 트랜스크립션	572
스트리밍 의료 트랜스크립션	576
사용자 지정 의학 어휘 생성 및 사용	579
사용자 지정 의학 어휘를 위한 텍스트 파일 생성	580
텍스트 파일을 사용하여 사용자 지정 의학 어휘 생성	584
사용자 지정 의학 어휘를 사용한 오디오 파일 트랜스크립션	586
사용자 지정 의학 어휘를 사용한 실시간 스트림의 트랜스크립션	588
Amazon Transcribe Medical의 문자 집합	590
트랜스크립트의 PHI 식별	592
오디오 파일에서 PHI 식별	593
실시간 스트림에서 PHI 식별	598
대체 트랜스크립션 생성	599
VPC 엔드포인트(AWS PrivateLink)	602
Amazon Transcribe Medical VPC 엔드포인트에 대한 고려 사항	603

Amazon Transcribe Medical용 인터페이스 VPC 엔드포인트 생성	603
Amazon Transcribe Medical 스트리밍을 위한 VPC 엔드포인트 정책 생성	604
공유 서브넷	605
AWS HealthScribe	606
보안	607
서비스 가용성	607
기술 요구 사항	607
지원되는 의료 전문 분야	607
워크플로	608
트랜스크립트 파일	609
임상 문서 파일	610
Medical Scribe 컨텍스트	611
HISTORY_AND_PHYSICAL 템플릿 섹션	611
GIRPP 템플릿 섹션	612
BIRP 템플릿 섹션	613
SIRP 템플릿 섹션	614
DAP 템플릿 섹션	614
BH_SOAP 템플릿 섹션	614
PH_SOAP 템플릿 섹션	615
트랜스크립션 작업	615
AWS HealthScribe 트랜스크립션 작업 시작	616
스트리밍	631
지침 및 요구 사항	631
ResourceAccessRoleArn 역할 권한	632
Starting AWS HealthScribe 스트리밍 트랜스크립션	634
AWS HealthScribe의 유희 데이터 암호화	650
고객 관리형 키 생성	651
AWS HealthScribe에 대한 고객 관리형 키 지정	655
AWS KMS 암호화 컨텍스트	655
AWS HealthScribe에 대한 암호화 키 모니터링	658
문서 기록	662
AWS 용어집	673
.....	dclxxiv

Amazon Transcribe란 무엇인가요?

Amazon Transcribe 는 기계 학습 모델을 사용하여 오디오를 텍스트로 변환하는 자동 음성 인식 서비스입니다. Amazon Transcribe 를 독립 실행형 트랜스크립션 서비스로 사용하거나 모든 애플리케이션에 speech-to-text 기능을 추가할 수 있습니다.

를 사용하면 언어 사용자 지정을 통해 특정 사용 사례의 정확도를 높이고, 콘텐츠를 필터링하여 고객 개인 정보 보호 또는 대상에 적합한 언어를 보장하고, 다중 채널 오디오의 콘텐츠를 분석하고, 개별 화자의 음성을 분할하는 등의 Amazon Transcribe 작업을 수행할 수 있습니다.

실시간으로 미디어를 트랜스크립션하거나(스트리밍) Amazon S3 버킷에 있는 미디어 파일을 트랜스크립션할 수 있습니다(배치). 각 트랜스크립션 유형에 지원되는 언어를 확인하려면 [지원되는 언어 및 언어별 기능](#) 테이블을 참조하세요.

주제

- [Amazon Transcribe 및 HIPAA 자격](#)
- [가격 책정](#)
- [리전 가용성 및 할당량](#)

이 서비스에 대한 짧은 비디오 둘러보기는 [란 무엇입니까 Amazon Transcribe?](#)를 참조하세요.

자세한 내용은 [Amazon Transcribe 작동 방법](#) 및 [시작하기 Amazon Transcribe](#) 섹션을 참조하세요.

Tip

Amazon Transcribe API에 대한 정보는 [API 참조](#)에 있습니다.

Amazon Transcribe 및 HIPAA 자격

Amazon Transcribe 는의 HIPAA 자격 및 BAA AWS의 적용을 받습니다. BAA 고객은 사용 시 저장 및 전송 중인 모든 PHI를 암호화해야 합니다. 자동 PHI 식별은가 Amazon Transcribe 운영되는 모든 리전에서 추가 비용 없이 사용할 수 있습니다. 자세한 내용은 [HIPAA 자격 및 BAA](#)를 참조하세요.

가격 책정

Amazon Transcribe 는 pay-as-you-go 서비스입니다. 요금은 초 단위의 트랜스크립션된 오디오를 기준으로 하며 월별로 청구됩니다.

사용량은 1초 단위로 청구되며 최소 청구 시간은 15초입니다. PII 콘텐츠 수정 및 사용자 지정 언어 모델과 같은 기능에는 추가 요금이 부과됩니다.

각에 대한 비용 정보는 [Amazon Transcribe 요금을](#) AWS 리전참조하세요.

리전 가용성 및 할당량

Amazon Transcribe 는 AWS 리전다음에서 지원됩니다.

리전	트랜스크립션 유형
af-south-1(케이프타운)	배치, 스트리밍
ap-east-1(홍콩)	일괄
ap-northeast-1(도쿄)	배치, 스트리밍
ap-northeast-2(서울)	배치, 스트리밍
ap-south-1(뭄바이)	배치, 스트리밍
ap-southeast-1(싱가포르)	배치, 스트리밍
ap-southeast-2(시드니)	배치, 스트리밍
ap-southeast-5(말레이시아)	스트리밍
ap-southeast-7(태국)	스트리밍
ca-central-1(캐나다, 중부)	배치, 스트리밍
eu-central-1(프랑크푸르트)	배치, 스트리밍
eu-central-2(취리히)	배치, 스트리밍
eu-north-1(스톡홀름)	일괄

리전	트랜스크립션 유형
eu-west-1(아일랜드)	배치, 스트리밍
eu-west-2(런던)	배치, 스트리밍
eu-west-3(파리)	일괄
me-south-1(바레인)	일괄
mx-central-1(멕시코)	스트리밍
sa-east-1(상파울루)	배치, 스트리밍
us-east-1(버지니아 북부)	배치, 스트리밍
us-east-2(오하이오)	배치, 스트리밍
us-gov-east-1(GovCloud, 미국 동부)	배치, 스트리밍
us-gov-west-1,(GovCloud, 미국 서부)	배치, 스트리밍
us-west-1(샌프란시스코)	일괄
us-west-2(오리건)	일괄, 스트림

Important

리전 지원은 Amazon Transcribe [Amazon Transcribe Medical](#), 및 [Call Analytics](#)에 따라 다릅니다.

지원되는 각 리전에 대한 엔드포인트를 얻으려면 AWS 일반 참조의 [서비스 엔드포인트](#)를 참조하세요.

트랜스크립션과 관련된 할당량 목록은 AWS 일반 참조의 [서비스 할당량](#)을 참조하세요. 일부 할당량은 요청 시 변경될 수 있습니다. 조정 가능 열에 '예'가 포함된 경우 증가를 요청할 수 있습니다. 이렇게 하려면 제공된 링크를 선택합니다.

Amazon Transcribe features

사용 사례에 가장 적합한 Amazon Transcribe 솔루션을 결정하는 데 도움이 되도록 다음 표에서는 기능 비교를 제공합니다.

참고로 '배치'와 'post-call'은 Amazon S3 버킷에 있는 파일을 트랜스크립션하는 것을 의미하고 '스트리밍'과 'real-time'은 미디어를 실시간으로 트랜스크립션하는 것을 의미합니다.

기능	Amazon Transcribe	Amazon Transcribe Medical ¹	Amazon Transcribe 통화 분석
구성 옵션			
대체 트랜스크립션	배치, 스트리밍	배치, 스트리밍	아니요
채널 식별	배치, 스트리밍	배치, 스트리밍	post-call, real-time
작업 대기열	일괄	아니요	post-call
언어 식별	배치, 스트리밍	아니요	post-call
다국어 식별	배치, 스트리밍	아니요	아니요
화자 분할	배치, 스트리밍	배치, 스트리밍	post-call
숫자 트랜스크립션 ²	배치, 스트리밍	배치, 스트리밍	post-call, real-time
대화 분석			
통화 특성	아니요	아니요	post-call
통화 요약 ²	아니요	아니요	post-call
사용자 지정 분류	아니요	아니요	post-call
실시간 범주 이벤트	아니요	아니요	real-time
실시간 문제 감지 ²	아니요	아니요	real-time
실시간 화자 감정	아니요	아니요	real-time

기능	Amazon Transcribe	Amazon Transcribe Medical ¹	Amazon Transcribe 통화 분석
화자 감정	아니요	아니요	post-call
언어 사용자 지정			
사용자 지정 언어 모델 ²	배치, 스트리밍	아니요	post-call, real-time
사용자 지정 어휘	배치, 스트리밍	배치, 스트리밍	post-call, real-time
리소스 조직			
태그 지정	일괄	일괄	post-call
민감한 데이터			
개인 건강 정보 식별 ²	아니요	배치, 스트리밍	아니요
개인 식별 정보 식별 ²	스트리밍	아니요	real-time
오디오 수정 ²	아니요	아니요	post-call, real-time
트랜스크립트 수정 ²	배치, 스트리밍	아니요	post-call, real-time
어휘 필터링	배치, 스트리밍	아니요	post-call, real-time
동영상			
자막	일괄	아니요	아니요

¹ Amazon Transcribe Medical은 미국 영어로만 제공됩니다.

² 일부 언어에서는 이 기능을 사용할 수 없습니다. 자세한 내용은 [지원되는 언어 및 언어별 기능](#) 테이블을 참조하세요.

지원되는 언어 및 언어별 기능

에서 지원하는 언어 Amazon Transcribe 는 다음 표에 나열되어 있으며 언어별 기능도 나열되어 있습니다. 트랜스크립션을 계속 진행하기 전에, 사용할 기능이 미디어의 해당 언어에 지원되는지 확인하세요.

전체 Amazon Transcribe 기능 목록을 보려면 [기능 요약](#)을 참조하세요.

다음 표에서 '배치'는 Amazon S3 버킷에 있는 미디어 파일 트랜스크립션을 나타내고 '스트리밍'은 스트리밍된 미디어를 실시간으로 트랜스크립션하는 것을 나타냅니다. Call Analytics 트랜스크립션의 경우 'post-call'는 Amazon S3 버킷에 있는 미디어 파일을 트랜스크립션하는 것을 의미하고는 스트리밍된 미디어를 실시간으로 트랜스크립션하는 것을 'real-time' 의미합니다.

언어	언어 코드	데이터 입력	번호 트랜스크립션	두문자어	사용자 지정 언어 모델	수정	Call Analytics * -
압하스어	ab-GE	일괄	아니요	일괄	아니요	아니요	아니요
아프리카나스어	af-ZA	배치, 스트리밍	아니요	배치, 스트리밍	아니요	아니요	아니요
아랍어, 걸프	ar-AE	배치, 스트리밍	아니요	아니요	아니요	아니요	post-call
아랍어, 현대 표준어	ar-SA	배치, 스트리밍	아니요	아니요	아니요	아니요	아니요
아르메니아어	hy-AM	일괄	아니요	일괄	아니요	아니요	아니요
아스투리아스어	ast-ES	일괄	아니요	일괄	아니요	아니요	아니요
아제르바이잔어	az-AZ	일괄	아니요	일괄	아니요	아니요	아니요
바쉬르어	ba-RU	일괄	아니요	일괄	아니요	아니요	아니요

언어	언어 코드	데이터 입력	번호 트랜스크립션	두문자어	사용자 지정 언어 모델	수정	Call Analytics * -
바스크어	eu-ES	배치, 스트리밍	아니요	배치, 스트리밍	아니요	아니요	아니요
벨라루스어	be-BY	일괄	아니요	일괄	아니요	아니요	아니요
벵골어	bn-IN	일괄	아니요	일괄	아니요	아니요	아니요
보스니아어	bs-BA	일괄	아니요	일괄	아니요	아니요	아니요
불가리아어	bg-BG	일괄	아니요	일괄	아니요	아니요	아니요
카탈루냐어	ca-ES	배치, 스트리밍	스트리밍	배치, 스트리밍	아니요	아니요	아니요
소라니어, 이란	ckb-IR	일괄	아니요	일괄	아니요	아니요	아니요
소라니어, 이라크	ckb-IQ	일괄	아니요	일괄	아니요	아니요	아니요
중국어, 광둥어	zh-HK (yue-HK)	배치, 스트리밍	아니요	아니요	아니요	아니요	아니요
중국어, 간체	zh-CN	배치, 스트리밍	아니요	아니요	아니요	아니요	post-call
중국어, 한자	zh-TW	배치, 스트리밍	아니요	아니요	아니요	아니요	아니요
크로아티아어	hr-HR	배치, 스트리밍	아니요	배치, 스트리밍	아니요	아니요	아니요

언어	언어 코드	데이터 입력	번호 트랜스크립션	두문자어	사용자 지정 언어 모델	수정	Call Analytics * -
체코어	cs-CZ	배치, 스트리밍	아니요	배치, 스트리밍	아니요	아니요	아니요
덴마크어	da-DK	배치, 스트리밍	아니요	배치, 스트리밍	아니요	아니요	아니요
네덜란드어	nl-NL	배치, 스트리밍	아니요	배치, 스트리밍	아니요	아니요	아니요
영어 , 호주	en-AU	배치, 스트리밍	배치, 스트리밍	배치, 스트리밍	배치, 스트리밍	스트리밍	post-call, real-time
영어 , 영국	en-GB	배치, 스트리밍	배치, 스트리밍	배치, 스트리밍	배치, 스트리밍	스트리밍	post-call, real-time
영어 , 인도	en-IN	배치, 스트리밍	배치, 스트리밍	배치, 스트리밍	아니요	스트리밍	post-call
영어 , 아일랜드	en-IE	배치, 스트리밍	배치, 스트리밍	배치, 스트리밍	아니요	스트리밍	post-call
영어 , 뉴질랜드	en-NZ	배치, 스트리밍	배치, 스트리밍	배치, 스트리밍	아니요	스트리밍	아니요
영어 , 스코틀랜드	en-AB	배치, 스트리밍	배치, 스트리밍	배치, 스트리밍	아니요	스트리밍	post-call
영어 , 남아프리카 공화국	en-ZA	배치, 스트리밍	배치, 스트리밍	배치, 스트리밍	아니요	스트리밍	아니요
영어 , 미국	en-US	배치, 스트리밍	배치, 스트리밍	배치, 스트리밍	배치, 스트리밍	배치, 스트리밍	post-call, real-time

언어	언어 코드	데이터 입력	번호 트랜스크립션	두문자어	사용자 지정 언어 모델	수정	Call Analytics *-
영어 , 웨일스	en-WL	배치, 스트리밍	배치, 스트리밍	배치, 스트리밍	아니요	스트리밍	post-call
에스토니아어	et-EE	일괄	아니요	일괄	아니요	아니요	아니요
에스토니아어	et-ET	일괄	아니요	일괄	아니요	아니요	아니요
페르시아어	fa-IR	배치, 스트리밍	아니요	아니요	아니요	아니요	아니요
핀란드어	fi-FI	배치, 스트리밍	아니요	배치, 스트리밍	아니요	아니요	아니요
프랑스어	fr-FR	배치, 스트리밍	배치, 스트리밍	배치, 스트리밍	아니요	스트리밍	post-call, real-time
프랑스어 , 캐나다	fr-CA	배치, 스트리밍	배치, 스트리밍	배치, 스트리밍	아니요	스트리밍	post-call, real-time
갈리시아어	gl-ES	배치, 스트리밍	아니요	배치, 스트리밍	아니요	아니요	아니요
조지아어	ka-GE	일괄	아니요	일괄	아니요	아니요	아니요
독일어	de-DE	배치, 스트리밍	배치, 스트리밍	배치, 스트리밍	배치, 스트리밍	스트리밍	post-call, real-time
독일어 , 스위스	de-CH	배치, 스트리밍	배치, 스트리밍	배치, 스트리밍	아니요	스트리밍	post-call
그리스어	el-GR	배치, 스트리밍	아니요	배치, 스트리밍	아니요	아니요	아니요

언어	언어 코드	데이터 입력	번호 트랜스크립션	두문자어	사용자 지정 언어 모델	수정	Call Analytics *-
구자라트어	gu-IN	일괄	아니요	일괄	아니요	아니요	아니요
하우사어	ha-NG	일괄	아니요	일괄	아니요	아니요	아니요
히브리어	he-IL	배치, 스트리밍	아니요	아니요	아니요	아니요	아니요
힌디어 , 인도	hi-IN	배치, 스트리밍	아니요	배치, 스트리밍	일괄	아니요	post-call
헝가리어	hu-HU	일괄	아니요	일괄	아니요	아니요	아니요
아이슬란드어	is-IS	일괄	아니요	일괄	아니요	아니요	아니요
인도네시아어	id-ID	배치, 스트리밍	아니요	배치, 스트리밍	아니요	아니요	아니요
이탈리아어	it-IT	배치, 스트리밍	배치, 스트리밍	배치, 스트리밍	아니요	스트리밍	post-call, real-time
일본어	ja-JP	배치, 스트리밍	배치, 스트리밍	아니요	배치, 스트리밍	아니요	post-call
커바일어	kab-DZ	일괄	아니요	일괄	아니요	아니요	아니요
칸나다어	kn-IN	일괄	아니요	일괄	아니요	아니요	아니요
카자흐어	kk-KZ	일괄	아니요	일괄	아니요	아니요	아니요
키냐르완다어	rw-RW	일괄	아니요	일괄	아니요	아니요	아니요
한국어	ko-KR	배치, 스트리밍	아니요	아니요	아니요	아니요	post-call

언어	언어 코드	데이터 입력	번호 트랜스크립션	두문자어	사용자 지정 언어 모델	수정	Call Analytics *-
키르기스어	ky-KG	일괄	아니요	일괄	아니요	아니요	아니요
라트비아어	lv-LV	배치, 스트리밍	아니요	배치, 스트리밍	아니요	아니요	아니요
리투아니아어	lt-LT	일괄	아니요	일괄	아니요	아니요	아니요
간다어	lg-IN	일괄	아니요	일괄	아니요	아니요	아니요
마케도니아어	mk-MK	일괄	아니요	일괄	아니요	아니요	아니요
말레이어	ms-MY	배치, 스트리밍	아니요	배치, 스트리밍	아니요	아니요	아니요
말라얄람어	ml-IN	일괄	아니요	일괄	아니요	아니요	아니요
몰타어	mt-MT	일괄	아니요	일괄	아니요	아니요	아니요
마라티어	mr-IN	일괄	아니요	일괄	아니요	아니요	아니요
메도우 마리	mhr-RU	일괄	아니요	일괄	아니요	아니요	아니요
몽골어	mn-MN	일괄	아니요	일괄	아니요	아니요	아니요
노르웨이어(보크몰)	no-NO	배치, 스트리밍	아니요	배치, 스트리밍	아니요	아니요	아니요
오디아/오리아어	or-IN	일괄	아니요	일괄	아니요	아니요	아니요

언어	언어 코드	데이터 입력	번호 트랜스크립션	두문자어	사용자 지정 언어 모델	수정	Call Analytics * -
파슈토어	ps-AF	일괄	아니요	일괄	아니요	아니요	아니요
폴란드어	pl-PL	배치, 스트리밍	아니요	배치, 스트리밍	아니요	아니요	아니요
포르투갈어	pt-PT	배치, 스트리밍	배치, 스트리밍	배치, 스트리밍	아니요	스트리밍	post-call
포르투갈어, 브라질	pt-BR	배치, 스트리밍	배치, 스트리밍	배치, 스트리밍	아니요	스트리밍	post-call, real-time
편자브어	pa-IN	일괄	아니요	일괄	아니요	아니요	아니요
루마니아어	ro-RO	배치, 스트리밍	아니요	배치, 스트리밍	아니요	아니요	아니요
러시아어	ru-RU	배치, 스트리밍	아니요	아니요	아니요	아니요	아니요
세르비아어	sr-RS	배치, 스트리밍	아니요	배치, 스트리밍	아니요	아니요	아니요
싱할라어	si-LK	일괄	아니요	일괄	아니요	아니요	아니요
슬로바키아어	sk-SK	배치, 스트리밍	아니요	배치, 스트리밍	아니요	아니요	아니요
슬로베니아어	sl-SI	일괄	아니요	일괄	아니요	아니요	아니요
소말리아어	so-SO	배치, 스트리밍	아니요	배치, 스트리밍	아니요	아니요	아니요
스페인어	es-ES	배치, 스트리밍	배치, 스트리밍	배치, 스트리밍	아니요	스트리밍	post-call

언어	언어 코드	데이터 입력	번호 트랜스크립션	두문자어	사용자 지정 언어 모델	수정	Call Analytics * -
스페인어 , 미국	es-US	배치, 스트리밍	배치, 스트리밍	배치, 스트리밍	배치, 스트리밍	배치, 스트리밍	post-call, real-time
순다어	su-ID	일괄	아니요	일괄	아니요	아니요	아니요
스와힐리어 , 케냐	sw-KE	일괄	아니요	일괄	아니요	아니요	아니요
스와힐리어 , 부룬디	sw-BI	일괄	아니요	일괄	아니요	아니요	아니요
스와힐리어 , 르완다	sw-RW	일괄	아니요	일괄	아니요	아니요	아니요
스와힐리어 , 탄자니아	sw-TZ	일괄	아니요	일괄	아니요	아니요	아니요
스와힐리어 , 우간다	sw-UG	일괄	아니요	일괄	아니요	아니요	아니요
스웨덴어	sv-SE	배치, 스트리밍	아니요	배치, 스트리밍	아니요	아니요	아니요
타갈로그어/필리핀어	tl-PH	배치, 스트리밍	아니요	배치, 스트리밍	아니요	아니요	아니요
타밀어	ta-IN	일괄	아니요	아니요	아니요	아니요	아니요
타타르어	tt-RU	일괄	아니요	일괄	아니요	아니요	아니요

언어	언어 코드	데이터 입력	번호 트랜스크립션	두문자어	사용자 지정 언어 모델	수정	Call Analytics *-
텔루구어	te-IN	일괄	아니요	아니요	아니요	아니요	아니요
태국어	th-TH	배치, 스트리밍	아니요	배치, 스트리밍	아니요	아니요	아니요
터키어	tr-TR	일괄	아니요	일괄	아니요	아니요	아니요
우크라이나어	uk-UA	배치, 스트리밍	아니요	배치, 스트리밍	아니요	아니요	아니요
위구르어	ug-CN	일괄	아니요	일괄	아니요	아니요	아니요
우즈베크어	uz-UZ	일괄	아니요	일괄	아니요	아니요	아니요
베트남어	vi-VN	배치, 스트리밍	아니요	배치, 스트리밍	아니요	아니요	아니요
웨일스어	cy-WL	일괄	아니요	일괄	아니요	아니요	아니요
월로프어	wo-SN	일괄	아니요	일괄	아니요	아니요	아니요
줄루어	zu-ZA	배치, 스트리밍	아니요	배치, 스트리밍	아니요	아니요	아니요

*다음의 Call Analytics 인사이트는 일부 영어 방언에서만 지원됩니다.

- [통화 요약](#): en-*(모든 영어 방언)
- [문제 감지](#): en-AU, en-GB, en-US

지원되는 프로그래밍 언어

Amazon Transcribe 는 다음 AWS SDKs를 지원합니다.

배치 트랜스크립션	스트리밍 트랜스크립션
.NET	.NET (Transcribe Medical and HealthScribe는 지원되지 않음)
AWS 명령줄 인터페이스(CLI)	스트리밍에는 CLI가 지원되지 않습니다.
C++	C++
Go	Go
Java V2	Java V2
JavaScript	JavaScript V3
PHP V3	SDK는 스트리밍에 지원되지 않습니다.
Python Boto3	용 Python 스트리밍 SDK Amazon Transcribe
Ruby V3	Ruby V3
Rust	Rust

에서 SDKs 사용하는 방법에 대한 자세한 내용은 단원을 Amazon Transcribe참조하십시오 [AWS SDKs](#).

사용 가능한 모든 AWS SDKs 및 빌더 도구에 대한 자세한 내용은 [빌드 기반 도구를 AWS](#) 참조하세요.

 Tip

SDK 코드 샘플은 다음 GitHub 리포지토리에서 찾을 수 있습니다.

- [AWS 코드 예제](#)
- [Amazon Transcribe 예시](#)

사용자 지정 어휘 및 어휘 필터를 위한 문자 집합

각 언어 Amazon Transcribe 지원마다 Amazon Transcribe가 인식할 수 있는 특정 문자 집합이 있습니다. 사용자 지정 어휘 또는 어휘 필터를 생성할 때는 해당 언어의 문자 집합에 나열된 문자만 사용합니다. 지원되지 않는 문자를 사용하는 경우 사용자 지정 어휘 또는 어휘 필터가 작동하지 않습니다.

Important

사용자 지정 어휘 파일이 다음 문자 집합 내에 나열된 지원되는 유니코드 코드 포인트와 코드 포인트 시퀀스만 사용하는지 확인합니다.

대부분의 유니코드 문자는 다른 코드 포인트를 사용하는 경우에도 많이 사용되는 글꼴에서 동일하게 나타날 수 있습니다. 이 설명서에 나열된 코드 포인트만 지원됩니다. 예를 들어, 프랑스어 단어 déjà는 사전 구성된 문자(여기에서 하나의 유니코드 값은 억양 표시가 되어 있는 문자를 나타냄) 또는 분해된 문자(여기에서 두 개의 유니코드 값은 억양 표시가 되어 있는 문자를 나타내며 하나의 값은 기본 문자를, 다른 값은 억양 표시를 나타냄)를 사용하여 렌더링할 수 있습니다.

- 사전 구성된 버전: 0064 **00E9** 006A **00E0**(déjà로 렌더링됨)
- 분해된 버전: 0064 **0065 0301** 006A **0061 0300**(déjà로 렌더링됨)

주제

- [압하스어 문자 집합](#)
- [아프리카나스어 문자 집합](#)
- [아랍어 문자 집합](#)
- [아스투리아스어 문자 집합](#)
- [아제르바이잔어 문자 집합](#)
- [아르메니아어 문자 집합](#)
- [바슈키르어 문자 집합](#)
- [바스크어 문자 집합](#)
- [벨라루스어 문자 집합](#)
- [벵골어 문자 집합](#)
- [보스니아어 문자 집합](#)
- [불가리아어 문자 집합](#)

- [카탈루냐어 문자 집합](#)
- [소라니어 문자 집합](#)
- [중국어, 만다린\(중국 본토\), 간체 문자 집합](#)
- [중국어, 만다린\(대만\), 한자 문자 집합](#)
- [중국어, 광둥어\(홍콩\), 한자 문자 집합](#)
- [크로아티아어 문자 집합](#)
- [체코어 문자 집합](#)
- [덴마크어 문자 집합](#)
- [네덜란드어 문자 집합](#)
- [영어 문자 집합](#)
- [에스토니아어 문자 집합](#)
- [페르시아어 문자 집합](#)
- [핀란드어 문자 집합](#)
- [프랑스어 문자 집합](#)
- [갈리시아어 문자 집합](#)
- [조지아어 문자 집합](#)
- [독일어 문자 집합](#)
- [그리스어 문자 집합](#)
- [구자라트어 문자 집합](#)
- [하우사어 문자 집합](#)
- [히브리어 문자 집합](#)
- [힌디어 문자 집합](#)
- [헝가리어 문자 집합](#)
- [아이슬란드어 문자 집합](#)
- [인도네시아어 문자 집합](#)
- [이탈리아어 문자 집합](#)
- [일본어 문자 집합](#)
- [커바일어 문자 집합](#)
- [칸나다어 문자 집합](#)
- [카자흐어 문자 집합](#)

- [키냐르완다어 문자 집합](#)
- [한국어 문자 집합](#)
- [키르기스어 문자 집합](#)
- [라트비아어 문자 집합](#)
- [리투아니아어 문자 집합](#)
- [간다어 문자 집합](#)
- [마케도니아어 문자 집합](#)
- [말레이어 문자 집합](#)
- [말라얄람어 문자 집합](#)
- [몰타어 문자 집합](#)
- [마라티어 문자 집합](#)
- [메도우 마리 문자 집합](#)
- [몽골어 문자 집합](#)
- [노르웨이어\(보크몰\) 문자 집합](#)
- [오디아/오리아어 문자 집합](#)
- [파슈토어 문자 집합](#)
- [폴란드어 문자 집합](#)
- [포르투갈어 문자 집합](#)
- [편자브어 문자 집합](#)
- [루마니아어 문자 집합](#)
- [러시아어 문자 집합](#)
- [세르비아어 문자 집합](#)
- [싱할라어 문자 집합](#)
- [슬로바키아어 문자 집합](#)
- [슬로베니아어 문자 집합](#)
- [소말리아어 문자 집합](#)
- [스페인어 문자 집합](#)
- [순다어 문자 집합](#)
- [스와힐리어 문자 집합](#)
- [스웨덴어 문자 집합](#)

- [타갈로그어/필리핀어 문자 집합](#)
- [타밀어 문자 집합](#)
- [타타르어 문자 집합](#)
- [텔루구어 문자 집합](#)
- [태국어 문자 집합](#)
- [터키어 문자 집합](#)
- [우크라이나어 문자 집합](#)
- [위구르어 문자 집합](#)
- [우즈베키스탄어 문자 집합](#)
- [베트남어 문자 집합](#)
- [웨일스어 문자 집합](#)
- [월로프어 문자 집합](#)
- [줄루어 문자 집합](#)

압하스어 문자 집합

압하스어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
a	0430	Ӏ	0459
б	0431	ӑ	045A
в	0432	ӕ	045B
г	0433	ӗ	045C

문자	코드	문자	코드
д	0434	#	045D
е	0435	ŷ	045E
ж	0436	ұ	045F
з	0437	ѓ	0491
и	0438	ƒ	0493
й	0439	ж	0497
к	043A	ƒ	0499
л	043B	ќ	049B
м	043C	ќ	049F
н	043D	ќ	04A1
о	043E	ћ	04A3
п	043F	ћ	04A5
р	0440	ќ	04A9
с	0441	ћ	04AB
т	0442	ћ	04AD
у	0443	ŷ	04AF
ф	0444	ŷ	04B1
х	0445	х	04B3
ц	0446	ц	04B5
ч	0447	ч	04B7

문자	코드	문자	코드
ш	0448	h	04BB
щ	0449	е	04BD
ъ	044A	ё	04BF
ы	044B	#	04CA
ь	044C	ă	04D1
э	044D	ä	04D3
ю	044E	ě	04D7
я	044F	ə	04D9
#	0450	з	04E1
ë	0451	й	04E3
ђ	0452	ö	04E7
í	0453	ө	04E9
є	0454	ȳ	04EF
s	0455	ÿ	04F1
i	0456	ý	04F3
ï	0457	#	04F7
j	0458	Ы	04F9
#	0525		

아프리카ンス어 문자 집합

아프리카ンス어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

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

아랍어 문자 집합

아랍어 사용자 지정 어휘의 경우 Phrase 필드에 다음 유니코드 문자를 사용할 수 있습니다. 하이픈(-) 문자를 사용하여 단어를 구분할 수도 있습니다.

문자	코드	문자	코드
ء	0621	س	0633
آ	0622	ش	0634
إ	0623	ص	0635
ؤ	0624	ض	0636
ط	0625	ط	0637

문자	코드	문자	코드
ئ	0626	ظ	0638
ل	0627	ع	0639
ب	0628	غ	063A
ة	0629	ف	0641
ت	062A	ق	0642
ث	062B	ك	0643
ج	062C	ل	0644
ح	062D	م	0645
خ	062E	ن	0646
د	062F	ه	0647
ذ	0630	و	0648
ر	0631	ى	0649
ز	0632	ي	064A

아스투리아스어 문자 집합

아스투리아스어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
á	00E1	ñ	00F1
é	00E9	ó	00F3
í	00ED	ú	00FA
ü	00FC		

아제르바이잔어 문자 집합

아제르바이잔어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
ä	00E4	ğ	011F
ç	00E7	ı	0131
ö	00F6	ş	015F
ü	00FC	ə	0259
.	0307		

아르메니아어 문자 집합

아르메니아어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z

- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
ᄀ	0561	ᄁ	0574
ᄂ	0562	ᄃ	0575
ᄄ	0563	ᄅ	0576
ᄆ	0564	ᄇ	0577
ᄈ	0565	ᄉ	0578
ᄊ	0566	ᄋ	0579
ᄌ	0567	ᄍ	057A
ᄎ	0568	ᄏ	057B
ᄐ	0569	ᄑ	057C
ᄒ	056A	ᄓ	057D
ᄔ	056B	ᄕ	057E
ᄖ	056C	ᄗ	057F
ᄙ	056D	ᄚ	0580
ᄛ	056E	ᄜ	0581
ᄝ	056F	ᄞ	0582
ᄟ	0570	ᄠ	0583
ᄡ	0571	ᄢ	0584

문자	코드	문자	코드
ŋ	0572	o	0585
fi	0573	ɸ	0586

바슈키르어 문자 집합

바슈키르어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
a	0430	ӑ	0459
б	0431	ӕ	045A
в	0432	ӓ	045B
г	0433	ӕ̃	045C
д	0434	#	045D
е	0435	ӕ̄	045E
ж	0436	ӕ̅	045F
з	0437	ӕ̆	0491
и	0438	ӕ̇	0493
й	0439	ӕ̈	0497
к	043A	ӕ̉	0499

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

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

바스크어 문자 집합

바스크어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
á	00E1	ñ	00F1
é	00E9	ó	00F3
í	00ED	ú	00FA

문자	코드	문자	코드
ü	00FC		

벨라루스어 문자 집합

벨라루스어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
a	0430	c	0441
б	0431	т	0442
в	0432	y	0443
г	0433	ф	0444
д	0434	x	0445
е	0435	ц	0446
ж	0436	ч	0447
з	0437	ш	0448
й	0439	ы	044B
к	043A	ь	044C
л	043B	э	044D
м	043C	ю	044E

문자	코드	문자	코드
н	043D	я	044F
о	043E	ë	0451
п	043F	і	0456
р	0440	ŷ	045E

벵골어 문자 집합

벵골어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

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

문자	코드	문자	코드
ঋ	098B	৳	09B0
এ	098F	৳	09B2
ঐ	0990	শ	09B6
ও	0993	ষ	09B7
ঔ	0994	স	09B8
ক	0995	হ	09B9
খ	0996	.	09BC
গ	0997	#	09BD
ঘ	0998	†	09BE
ঙ	0999	†	09BF
চ	099A	†	09C0
ছ	099B	৳	09C1
জ	099C	৳	09C2
ঝ	099D	৳	09C3
ঞ	099E	৳	09C4
ট	099F	৳	09C7
ঠ	09A0	৳	09C8
ড	09A1	৳	09CB
ঢ	09A2	৳	09CC
ণ	09A3	৳	09CD

문자	코드	문자	코드
ত	09A4	#	09CE
থ	09A5	৳	09D7

보스니아어 문자 집합

보스니아어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
ć	0107	đ	0111
č	010D	š	0161
ž	017E		

불가리아어 문자 집합

불가리아어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

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

카탈루냐어 문자 집합

카탈루냐어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
à	00E0	ï	00EF
ç	00E7	ò	00F2
è	00E8	ó	00F3
é	00E9	ú	00FA
í	00ED	ü	00FC
।	0140		

소라니어 문자 집합

소라니어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
ʾ	0626	ﻡ	0645
ا	0627	ن	0646
ب	0628	و	0648
ت	062A	پ	067E
ج	062C	چ	0686
ح	062D	ر	0695

문자	코드	문자	코드
خ	062E	ژ	0698
د	062F	ڦ	06A4
ر	0631	ک	06A9
ز	0632	گ	06AF
س	0633	چ	06B5
ش	0634	ھ	06BE
ع	0639	و	06C6
غ	063A	ؤ	06C7
ف	0641	ی	06CC
ق	0642	ئ	06CE
ل	0644	ه	06D5

중국어, 만다린(중국 본토), 간체 문자 집합

중국어(간체) 사용자 지정 어휘의 경우, Phrase 필드는 다음 파일에 나열된 문자를 사용할 수 있습니다.

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

SoundsLike 필드에는 다음 파일에 나열된 병음 음절이 포함될 수 있습니다.

- [pinyin-character-set](#)

SoundsLike 필드에서 병음 음절을 사용하는 경우 하이픈(-)으로 음절을 구분합니다.

Amazon Transcribe는 번호를 사용하여 중국어(간체)의 4개 톤을 나타냅니다. 다음 테이블은 톤 표시가 'ma' 단어에 매핑되는 방법을 보여줍니다.

톤	톤 표시	톤 번호
톤 1	mā	ma1
톤 2	má	ma2
톤 3	mǎ	ma3
톤 4	mà	ma4

Note

5번째(중간) 톤의 경우 톤 1을 사용할 수 있습니다. 단, 'er'는 톤 2에 매핑되어야 합니다. 예를 들어, 打转儿는 'da3-zhuan4-er2'로 표시됩니다.

중국어(간체) 사용자 지정 어휘는 IPA 필드를 사용하지 않지만 사용자 지정 어휘 테이블에 IPA 헤더를 계속 포함시켜야 합니다.

다음은 텍스트 형식의 입력 파일 예입니다. 이 예시에서는 스페이스를 사용하여 열을 정렬합니다. 입력 파일은 TAB 문자를 사용하여 열을 구분해야 합니다. DisplayAs 열에 스페이스를 포함합니다.

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

중국어, 만다린(대만), 한자 문자 집합

중국어(한자) 사용자 지정 어휘의 경우, Phrase 필드는 다음 파일에 나열된 문자를 사용할 수 있습니다.

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

SoundsLike 필드에는 다음 파일에 나열된 주음 음절이 포함될 수 있습니다.

- [zhuyin-character-set](#)

SoundsLike 필드에서 주음 음절을 사용하는 경우 하이픈(-)으로 음절을 구분합니다.

Amazon Transcribe는 번호를 사용하여 중국어(한자)의 4개 톤을 나타냅니다. 다음 테이블은 톤 표시가 ㄱ ㅏ 단어에 매핑되는 방법을 보여줍니다.

톤	톤 표시
톤 1	ㄱ ㅏ
톤 2	ㄱ ㅏ´
톤 3	ㄱ ㅏˇ
톤 4	ㄱ ㅏ`

중국어(한자) 사용자 지정 어휘는 IPA 필드를 사용하지 않지만 사용자 지정 어휘 테이블에 IPA 헤더를 계속 포함시켜야 합니다.

다음은 텍스트 형식의 입력 파일 예입니다. 이 예시에서는 스페이스를 사용하여 열을 정렬합니다. 입력 파일은 TAB 문자를 사용하여 열을 구분해야 합니다. DisplayAs 열에 스페이스를 포함합니다.

Phrase	SoundsLike	IPA	DisplayAs
##	###`-##		
##	###ˇ-##´		
####	###´-##´-##`-##ˇ		
#####	#`-###`-##´-##ˇ-###`	###	

중국어, 광둥어(홍콩), 한자 문자 집합

홍콩 중국어(광둥어) 사용자 지정 어휘의 경우, Phrase 필드는 다음 파일에 나열된 문자를 사용할 수 있습니다.

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

크로아티아어 문자 집합

크로아티아어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z

- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
ć	0107	ď	0111
č	010D	š	0161
ž	017E		

체코어 문자 집합

체코어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
á	00E1	ď	010F
é	00E9	ě	011B
í	00ED	ň	0148
ó	00F3	ř	0159
ú	00FA	š	0161
ý	00FD	td	0165

문자	코드	문자	코드
č	010D	ů	016F
ž	017E		

덴마크어 문자 집합

덴마크어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- A - Z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
Å	00C5	æ	00E6
Æ	00C6	é	00E9
Ø	00D8	ø	00F8
å	00E5		

네덜란드어 문자 집합

네덜란드어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- A - Z
- ' (아포스트로피)
- - (하이픈)

- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

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

영어 문자 집합

영어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- A - Z
- ' (아포스트로피)
- - (하이픈)
- . (마침표)

에스토니아어 문자 집합

에스토니아어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
ä	00E4	ü	00FC
õ	00F5	š	0161
ö	00F6	ž	017E

페르시아어 문자 집합

페르시아어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

문자	코드	문자	코드
ء	0621	ط	0638
آ	0622	ع	0639
إ	0623	غ	063A
ؤ	0624	ف	0641
ئ	0626	ق	0642
ل	0627	ج	0644
ب	0628	م	0645

문자	코드	문자	코드
ت	062A	ن	0646
ث	062B	ه	0647
ج	062C	و	0648
ح	062D	ـ	064E
خ	062E	ف	064F
د	062F	ـ	0650
ذ	0630	س	0651
ر	0631	پ	067E
ز	0632	چ	0686
س	0633	ژ	0698
ش	0634	ک	06A9
ص	0635	گ	06AF
ض	0636	ی	06CC
ط	0637		

핀란드어 문자 집합

핀란드어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
ä	00E4	ö	00F6
å	00E5	š	0161
ž	017E		

프랑스어 문자 집합

프랑스어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- A - Z
- ' (아포스트로피)
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
À	00C0	à	00E0
Â	00C2	â	00E2
Ç	00C7	ç	00E7
È	00C8	è	00E8
É	00C9	é	00E9
Ê	00CA	ê	00EA
Ë	00CB	ë	00EB
Î	00CE	î	00EE

문자	코드	문자	코드
İ	00CF	ï	00EF
Ô	00D4	ô	00F4
Ö	00D6	ö	00F6
Ù	00D9	ù	00F9
Û	00DB	û	00FB
Ü	00DC	ü	00FC

갈리시아어 문자 집합

갈리시아어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
á	00E1	ñ	00F1
é	00E9	ó	00F3
í	00ED	ú	00FA
ü	00FC		

조지아어 문자 집합

조지아어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
ა	10D0	რ	10E0
ბ	10D1	ს	10E1
გ	10D2	ტ	10E2
დ	10D3	ჯ	10E3
ე	10D4	ფ	10E4
ვ	10D5	ქ	10E5
ზ	10D6	ღ	10E6
თ	10D7	ყ	10E7
ო	10D8	შ	10E8
პ	10D9	ჩ	10E9
ც	10DA	ც	10EA
ძ	10DB	ძ	10EB
წ	10C	წ	10EC
ჭ	10DD	ჭ	10ED
ხ	10DE	ხ	10EE
ჯ	10DF	ჯ	10EF

문자	코드	문자	코드
3	10F0		

독일어 문자 집합

독일어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- A - Z
- ' (아포스트로피)
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
ä	00E4	Ä	00C4
ö	00F6	Ö	00D6
ü	00FC	Ü	00DC
ß	00DF		

그리스어 문자 집합

그리스어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
ά	03AC	ν	03BD
έ	03AD	ξ	03BE
ή	03AE	ο	03BF
ί	03AF	π	03C0
ϋ	03B0	ρ	03C1
α	03B1	ς	03C2
β	03B2	σ	03C3
γ	03B3	τ	03C4
δ	03B4	υ	03C5
ε	03B5	φ	03C6
ζ	03B6	χ	03C7
§	03B7	ψ	03C8
θ	03B8	ω	03C9
ι	03B9	ï	03CA
κ	03BA	ü	03CB
λ	03BB	ó	03CC
μ	03BC	ώ	03CE
ĩ	0390		

구자라트어 문자 집합

구자라트어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
ઁ	0A81	ઃ	0AA6
ઃ	0A82	઼	0AA7
ઃ	0A83	ઽ	0AA8
અ	0A85	઼	0AAA
અ	0A86	ઽ	0AAB
ઇ	0A87	઼	0AAC
ઇ	0A88	઼	0AAD
ઉ	0A89	઼	0AAE
ઊ	0A8A	઼	0AAF
ઋ	0A8B	ર	0AB0
ઁ	0A8D	લ	0AB2
એ	0A8F	ળ	0AB3
ઐ	0A90	વ	0AB5
ઑ	0A91	શ	0AB6
઒	0A93	ષ	0AB7
ઔ	0A94	સ	0AB8

문자	코드	문자	코드
ᄒ	0A95	ᄒ	0AB9
ᄒ	0A96	.	0ABC
ᄒ	0A97	ᄒ	0ABE
ᄒ	0A98	ᄒ	0ABF
ᄒ	0A99	ᄒ	0AC0
ᄒ	0A9A	ᄒ	0AC1
ᄒ	0A9B	ᄒ	0AC2
ᄒ	0A9C	ᄒ	0AC3
ᄒ	0A9D	ᄒ	0AC5
ᄒ	0A9E	ᄒ	0AC7
ᄒ	0A9F	ᄒ	0AC8
ᄒ	0AA0	ᄒ	0AC9
ᄒ	0AA1	ᄒ	0ACB
ᄒ	0AA2	ᄒ	0ACC
ᄒ	0AA3	ᄒ	0ACD
ᄒ	0AA4	ᄒ	0AD0
ᄒ	0AA5	ᄒ	0AE0

하우사어 문자 집합

하우사어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z

- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
k	0199	b	0253
y	01B4	d	0257
~	0303		

히브리어 문자 집합

히브리어 사용자 지정 어휘의 경우 Phrase 필드에 다음 유니코드 문자를 사용할 수 있습니다.

문자	코드	문자	코드
-	002D	א	05DD
כ	05D0	נ	05DE
כּ	05D1	ך	05DF
ל	05D2	ן	05E0
ט	05D3	ס	05E1
ה	05D4	ץ	05E2
ו	05D5	ף	05E3
ז	05D6	פּ	05E4
ח	05D7	ץ	05E5
ט	05D8	צ	05E6

문자	코드	문자	코드
ᳵ	05D9	ᳶ	05E7
᳷	05DA	᳸	05E8
᳹	05DB	ᳺ	05E9
᳻	05DC	᳼	05EA

힌디어 문자 집합

힌디어 사용자 지정 어휘의 경우 Phrase 필드에 다음 유니코드 문자를 사용할 수 있습니다.

문자	코드	문자	코드
-	002D	थ	0925
.	002E	द	0926
ॐ	0901	ध	0927
ॠ	0902	न	0928
:	0903	प	092A
अ	0905	फ	092B
आ	0906	ब	092C
इ	0907	भ	092D
ई	0908	म	092E
उ	0909	य	092F
ऊ	090A	र	0930
ऋ	090B	ल	0932

문자	코드	문자	코드
ए	090F	व	0935
ऐ	0910	श	0936
ऑ	0911	ष	0937
ओ	0913	स	0938
औ	0914	ह	0939
क	0915	ट	093E
ख	0916	ठ	093F
ग	0917	ड	0940
घ	0918	ढ	0941
ङ	0919	न	0942
च	091A	त	0943
छ	091B	थ	0945
ज	091C	द	0947
झ	091D	ध	0948
ञ	091E	न	0949
ट	091F	प	094B
ठ	0920	फ	094C
ड	0921	ब	094D
ढ	0922	भ	095B
ण	0923	म	095C

문자	코드	문자	코드
त	0924	ढ	095D

Amazon Transcribe는 다음 문자를 매핑합니다.

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

헝가리어 문자 집합

헝가리어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
á	00E1	ö	00F6
é	00E9	ú	00FA

문자	코드	문자	코드
í	00ED	ü	00FC
ó	00F3	ö	0151
ű	0171		

아이슬란드어 문자 집합

아이슬란드어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

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

인도네시아어 문자 집합

인도네시아어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- A - Z
- ' (아포스트로피)

- - (하이픈)
- . (마침표)

이탈리아어 문자 집합

이탈리아어 사용자 지정 어휘의 경우 `Phrase` 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- A - Z
- ' (아포스트로피)
- - (하이픈)
- . (마침표)

`Phrase` 필드에 다음 유니코드 문자도 사용할 수 있습니다.

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

일본어 문자 집합

일본어 사용자 지정 어휘의 경우 DisplayAs 필드는 모든 히라가나, 가타카나, 간지 문자와 전각 로마자 대문자를 지원합니다.

Phrase 필드는 다음 파일에 나열된 문자를 지원합니다.

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

커바일어 문자 집합

커바일어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

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

칸나다어 문자 집합

칸나다어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z

- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
◌	0C82	ᄒ	0CA7
◌◌	0C83	ᄔ	0CA8
ᄒ	0C85	ᄔ	0CAA
ᄒ	0C86	ᄔ	0CAB
ᄒ	0C87	ᄔ	0CAC
ᄒ	0C88	ᄔ	0CAD
ᄒ	0C89	ᄔ	0CAE
ᄒ	0C8A	ᄔ	0CAF
ᄒ	0C8B	ᄔ	0CB0
ᄒ	0C8E	ᄔ	0CB2
ᄒ	0C8F	ᄔ	0CB3
ᄒ	0C90	ᄔ	0CB5
ᄒ	0C92	ᄔ	0CB6
ᄒ	0C93	ᄔ	0CB7
ᄒ	0C94	ᄔ	0CB8
ᄒ	0C95	ᄔ	0CB9
ᄒ	0C96	#	0CBC

문자	코드	문자	코드
ᄀ	0C97	#	0CBD
ᄁ	0C98	ᄂ	0CBE
ᄃ	0C99	ᄄ	0CBF
ᄅ	0C9A	ᄆ	0CC0
ᄇ	0C9B	ᄈ	0CC1
ᄉ	0C9C	ᄊ	0CC2
ᄋ	0C9D	ᄌ	0CC3
ᄍ	0C9E	ᄎ	0CC6
ᄏ	0C9F	ᄐ	0CC7
ᄑ	0CA0	ᄒ	0CC8
ᄓ	0CA1	ᄔ	0CCA
ᄕ	0CA2	ᄖ	0CCB
ᄗ	0CA3	ᄘ	0CCC
ᄙ	0CA4	ᄚ	0CCD
ᄛ	0CA5	ᄜ	0CD5
ᄝ	0CA6	ᄞ	0CD6
ᄟ	0CE0		

카자흐어 문자 집합

카자흐어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z

- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
т	0442	ы	044B
б	0431	я	044F
о	043E	с	0441
п	043F	h	04BB
ш	0448	д	0434
и	0438	р	0440
ч	0447	г	0433
н	043D	ё	0451
қ	049B	й	0439
і	0456	ө	04E9
щ	0449	в	0432
е	0435	э	044D
ə	04D9	ң	04A3
ю	044E	л	043B
з	0437	ф	0444
х	0445	к	043A
ц	0446	у	0443

문자	코드	문자	코드
Ÿ	04AF	Ж	0436
М	043C	ƒ	0493
Ь	044C	а	0430
Ъ	044A	ყ	04B1

키냐르완다어 문자 집합

키냐르완다어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
á	00E1	ó	00F3
â	00E2	ô	00F4
ã	00E3	ú	00FA
ç	00E7	ü	00FC
è	00E8	ā	0101
é	00E9	-	0113
ê	00EA	ī	012B
ë	00EB	ō	014D
í	00ED	ū	016B

문자	코드	문자	코드
ï	00EF	'	0301

한국어 문자 집합

한국어 사용자 지정 어휘의 경우 Phrase 필드에 모든 한글 음절을 사용할 수 있습니다. 자세한 내용은 Wikipedia의 [Hangul Syllables](#)를 참조하세요.

키르기스어 문자 집합

키르기스어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
а	0430	ӡ	0459
б	0431	Ӣ	045A
в	0432	ӣ	045B
г	0433	Ӥ	045C
д	0434	Ӧ	045D
е	0435	ӧ	045E
ж	0436	Ө	045F
з	0437	Ӫ	0491
и	0438	Ӭ	0493

문자	코드	문자	코드
й	0439	ж	0497
к	043A	з	0499
л	043B	қ	049B
м	043C	к	049F
н	043D	к	04A1
о	043E	һ	04A3
п	043F	һ	04A5
р	0440	қ	04A9
с	0441	ç	04AB
т	0442	т	04AD
у	0443	ү	04AF
ф	0444	ү	04B1
х	0445	х	04B3
ц	0446	ц	04B5
ч	0447	ч	04B7
ш	0448	h	04BB
щ	0449	ё	04BD
ъ	044A	ё	04BF
ы	044B	#	04CA
ь	044C	ă	04D1

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

라트비아어 문자 집합

라트비아어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
ā	0101	ķ	0137

문자	코드	문자	코드
č	010D	į	013C
-	0113	ŋ	0146
ǵ	0123	š	0161
ī	012B	ū	016B
ž	017E		

리투아니아어 문자 집합

리투아니아어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

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

간다어 문자 집합

간다어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
ÿ	00FF	ŋ	014B

마케도니아어 문자 집합

마케도니아어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
a	0430	љ	0459
б	0431	њ	045A
в	0432	ћ	045B
г	0433	ќ	045C
д	0434	#	045D
е	0435	џ	045E
ж	0436	џ	045F
з	0437	ѓ	0491

문자	코드	문자	코드
и	0438	ѣ	0493
й	0439	ж	0497
к	043A	з	0499
л	043B	ќ	049B
м	043C	к	049F
н	043D	т	04A1
о	043E	ћ	04A3
п	043F	џ	04A5
р	0440	ѡ	04A9
с	0441	ѣ	04AB
т	0442	џ	04AD
у	0443	ѣ	04AF
ѣ	0444	ѣ	04B1
х	0445	х	04B3
ц	0446	ц	04B5
ч	0447	ч	04B7
ш	0448	h	04BB
щ	0449	ѣ	04BD
ъ	044A	ѣ	04BF
ы	044B	#	04CA

문자	코드	문자	코드
ь	044C	ă	04D1
э	044D	ä	04D3
ю	044E	ě	04D7
я	044F	ə	04D9
#	0450	з	04E1
ë	0451	й	04E3
ђ	0452	ö	04E7
í	0453	ө	04E9
є	0454	ȳ	04EF
s	0455	ÿ	04F1
i	0456	ý	04F3
ï	0457	#	04F7
j	0458	Ы	04F9

말레이어 문자 집합

말레이어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- A - Z
- ' (아포스트로피)
- - (하이픈)
- . (마침표)

말라얄람어 문자 집합

말라얄람어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
ഌ	0D02	ന	0D28
ഃ	0D03	പ	0D2A
അ	0D05	ഫ	0D2B
ആ	0D06	ബ	0D2C
ഇ	0D07	ഭ	0D2D
ഈ	0D08	മ	0D2E
ഉ	0D09	യ	0D2F
ഊ	0D0A	ര	0D30
ഋ	0D0B	റ	0D31
എ	0D0E	ല	0D32
ഏ	0D0F	ള	0D33
ഐ	0D10	ഴ	0D34
ഒ	0D12	വ	0D35
ഓ	0D13	ശ	0D36

문자	코드	문자	코드
ഓ	0D14	ഔ	0D37
ക	0D15	സ	0D38
ഖ	0D16	ഹ	0D39
ഗ	0D17	ഓ	0D3E
ഘ	0D18	റ	0D3F
ങ	0D19	ഴ	0D40
ച	0D1A	റ്റ	0D41
ഹര	0D1B	റ്റ	0D42
ജ	0D1C	ട	0D43
ഡ	0D1D	െ	0D46
ണ	0D1E	േ	0D47
ട	0D1F	ൈ	0D48
ഓ	0D20	ൊ	0D4A
ഡ	0D21	ോ	0D4B
ഡ	0D22	ൌ	0D4C
ണ	0D23	ഃ	0D4D
ത	0D24	#	0D7A
ഥ	0D25	#	0D7B
ദ	0D26	#	0D7C
ധ	0D27	#	0D7D

몰타어 문자 집합

몰타어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

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

마라티어 문자 집합

마라티어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
ॐ	0901	ॠ	0925
.	0902	ॡ	0926

문자	코드	문자	코드
:	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
एँ	090D	र	0930
ए	090F	ल	0932
ऐ	0910	ळ	0933
औ	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

문자	코드	문자	코드
छ	091B	उ	0941
ज	091C	२	0942
झ	091D	ॢ	0943
ञ	091E	ॣ	0945
ट	091F	।	0947
ठ	0920	॥	0948
ड	0921	०	0949
ढ	0922	१	094B
ण	0923	ॡ	094C
त	0924	ॢ	094D
ॐ	0950		

메도우 마리 문자 집합

메도우 마리 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
a	0430	љ	0459
б	0431	њ	045A

문자	코드	문자	코드
в	0432	ћ	045B
г	0433	ќ	045C
д	0434	#	045D
е	0435	ђ	045E
ж	0436	џ	045F
з	0437	ѓ	0491
и	0438	ѣ	0493
й	0439	жџ	0497
к	043A	џ	0499
л	043B	ќ	049B
м	043C	к	049F
н	043D	ќ	04A1
о	043E	ћ	04A3
п	043F	ѓ	04A5
р	0440	џ	04A9
с	0441	џ	04AB
т	0442	џ	04AD
у	0443	џ	04AF
ф	0444	џ	04B1
х	0445	х	04B3

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

몽골어 문자 집합

몽골어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
a	0430	ь	0459
б	0431	ъ	045A
в	0432	һ	045B
г	0433	ќ	045C
д	0434	#	045D
е	0435	ӧ	045E
ж	0436	ұ	045F
з	0437	ѓ	0491
и	0438	ґ	0493
й	0439	ж	0497
к	043A	з	0499
л	043B	қ	049B
м	043C	к	049F
н	043D	к	04A1

문자	코드	문자	코드
o	043E	Һ	04A3
п	043F	Һ	04A5
p	0440	Ɔ	04A9
c	0441	ç	04AB
т	0442	Ƨ	04AD
y	0443	Ƴ	04AF
ф	0444	ƴ	04B1
x	0445	х	04B3
ц	0446	ц	04B5
ч	0447	ч	04B7
ш	0448	h	04BB
щ	0449	ё	04BD
ъ	044A	ё	04BF
ы	044B	#	04CA
ь	044C	ă	04D1
э	044D	ä	04D3
ю	044E	ě	04D7
я	044F	ə	04D9
#	0450	з	04E1
ë	0451	й	04E3

문자	코드	문자	코드
ħ	0452	ö	04E7
í	0453	ø	04E9
€	0454	ÿ	04EF
s	0455	ÿ	04F1
i	0456	ÿ	04F3
ï	0457	#	04F7
j	0458	ÿ	04F9

노르웨이어(보크몰) 문자 집합

노르웨이어(보크몰) 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
å	00E5	æ	00E6
ø	00F8		

오디아/오리야어 문자 집합

오디아/오리야어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z

- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
ୠ	0B01	ୡ	0B26
ୢ	0B02	ୣ	0B27
୦	0B03	ୠ	0B28
ୡ	0B05	ୢ	0B2A
ୣ	0B06	୦	0B2B
ୠ	0B07	ୡ	0B2C
ୢ	0B08	ୣ	0B2D
ୠ	0B09	ୡ	0B2E
ୢ	0B0A	ୣ	0B2F
ୠ	0B0B	ୡ	0B30
ୣ	0B0F	ୠ	0B32
ୠୡ	0B10	ୠୡ	0B33
ୠୢ	0B13	ୠୢ	0B36
ୠୣ	0B14	ୠୣ	0B37
ୠୠ	0B15	ୠୠ	0B38
ୠୡ	0B16	ୠୡ	0B39
ୠୢ	0B17	ୠୢ	0B3C

문자	코드	문자	코드
௪	0B18		0B3E
௫	0B19	ˆ	0B3F
௶	0B1A	1	0B40
௷	0B1B	∞	0B41
௸	0B1C	˘	0B42
௹	0B1D	∞	0B43
௺	0B1E	6	0B47
௻	0B1F	6˘	0B48
௼	0B20	6#	0B4B
௽	0B21	6˘	0B4C
௾	0B22	˘	0B4D
௿	0B23	˘	0B56
௻	0B24	௹	0B5F
௼	0B25	௹	0B60
#	0B71		

파슈토어 문자 집합

파슈토어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
آ	0622	و	0648
أ	0623	ي	064A
ؤ	0624	ٲ	064B
ئ	0626	ٲٲ	064C
ا	0627	ٲٲ	064D
ب	0628	ٲ	064E
ت	062A	ٲٲ	064F
ث	062B	ٲ	0650
ج	062C	ٲٲ	0651
ح	062D	ٲ	0652
خ	062E	#	0654
د	062F	ٲ	0670
ذ	0630	ٲٲ	067C
ر	0631	ٲٲ	067E
ز	0632	خ	0681
س	0633	خ	0685
ش	0634	چ	0686
ص	0635	ٲ	0689
ض	0636	ر	0693

문자	코드	문자	코드
ط	0637	ڀ	0696
ظ	0638	ڙ	0698
ع	0639	ڻ	069A
غ	063A	ڪ	06A9
ف	0641	گ	06AB
ق	0642	گ	06AF
ج	0644	ڻ	06BC
م	0645	ي	06CC
ن	0646	ي	06CD
ه	0647	ڀ	06D0

폴란드어 문자 집합

폴란드어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
ó	00F3	ł	0142
ą	0105	ń	0144
ć	0107	ś	015B

문자	코드	문자	코드
ę	0119	ż	017A
ż	017C		

포르투갈어 문자 집합

포르투갈어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- A - Z
- ' (아포스트로피)
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
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

문자	코드	문자	코드
Ě	00CB	ě	00EB
Í	00CD	í	00ED
Ñ	00D1	ñ	00F1
Ó	00D3	ó	00F3
Ô	00D4	ô	00F4
Õ	00D5	õ	00F5
Ö	00D6	ö	00F6
Ú	00DA	ú	00FA
Ü	00DC	ü	00FC

편자브어 문자 집합

편자브어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
𑌀	0A05	𑌀	0A27
𑌁	0A06	𑌁	0A28
𑌂	0A07	𑌂	0A2A
𑌃	0A08	𑌃	0A2B

문자	코드	문자	코드
ੳ	0A09	ਬ	0A2C
ੴ	0A0A	ਭ	0A2D
ਏ	0A0F	ਮ	0A2E
ਐ	0A10	ਯ	0A2F
ੳ	0A13	ਰ	0A30
ੳੳ	0A14	ਲ	0A32
ਕ	0A15	ਵ	0A35
ਖ	0A16	ਸ	0A38
ਗ	0A17	ਹ	0A39
ਘ	0A18	.	0A3C
ਙ	0A19	ਟ	0A3E
ਚ	0A1A	ਫਿ	0A3F
ਛ	0A1B	ਫੀ	0A40
ਜ	0A1C	-	0A41
ਝ	0A1D	=	0A42
ਞ	0A1E	~	0A47
ਟ	0A1F	ਯ	0A48
ਠ	0A20	ਯ	0A4B
ਡ	0A21	ਯ	0A4C
ਢ	0A22	ਯ	0A4D

문자	코드	문자	코드
Ѡ	0A23	ѡ	0A5C
Ѣ	0A24	ѣ	0A70
Ѥ	0A25	ѥ	0A71
Ѧ	0A26	ѧ	0A72
Ѩ	0A73		

루마니아어 문자 집합

루마니아어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
ă	0103	#	0219
â	00E2	#	021B
î	00EE	ș	015F
ț	0163		

러시아어 문자 집합

러시아어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

문자	코드	문자	코드
'	0027	п	043F
-	002D	р	0440
.	002E	с	0441
а	0430	т	0442
б	0431	у	0443
в	0432	ф	0444
г	0433	х	0445
д	0434	ц	0446
е	0435	ч	0447
ж	0436	ш	0448
з	0437	щ	0449
и	0438	ъ	044A
й	0439	ы	044B
к	043A	ь	044C
л	043B	э	044D
м	043C	ю	044E
н	043D	я	044F
о	043E	ё	0451

세르비아어 문자 집합

세르비아어 사용자 지정 어휘의 경우 `Phrase` 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
ć	0107	ı	0456
č	010D	ï	0457
đ	0111	j	0458
š	0161	љ	0459
ž	017E	њ	045A
а	0430	ћ	045B
б	0431	ќ	045C
в	0432	#	045D
г	0433	ђ	045E
д	0434	џ	045F
е	0435	ѓ	0491
ж	0436	ѣ	0493
з	0437	ж	0497
и	0438	ѕ	0499
й	0439	ќ	049B
к	043A	к	049F

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

문자	코드	문자	코드
я	044F	й	04E3
#	0450	ö	04E7
ë	0451	ө	04E9
ђ	0452	ȳ	04EF
í	0453	ÿ	04F1
є	0454	ŷ	04F3
s	0455	#	04F7
İ	04F9		

싱할라어 문자 집합

싱할라어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
#	0D82	#	0DAF
#	0D83	#	0DB0
#	0D85	#	0DB1
#	0D86	#	0DB3
#	0D87	#	0DB4

문자	코드	문자	코드
#	0D88	#	0DB5
#	0D89	#	0DB6
#	0D8A	#	0DB7
#	0D8B	#	0DB8
#	0D8C	#	0DB9
#	0D8D	#	0DBA
#	0D91	#	0DB
#	0D92	#	0DBD
#	0D93	#	0DC0
#	0D94	#	0DC1
#	0D95	#	0DC2
#	0D96	#	0DC3
#	0D9A	#	0DC4
#	0D9B	#	0DC5
#	0D9C	#	0DC6
#	0D9D	#	0DCA
#	0D9E	#	0DCF
#	0D9F	#	0DD0
#	0DA0	#	0DD1
#	0DA1	#	0DD2

문자	코드	문자	코드
#	0DA2	#	0DD3
#	0DA3	#	0DD4
#	0DA4	#	0DD6
#	0DA5	#	0DD8
#	0DA7	#	0DD9
#	0DA8	##	0DDA
#	0DA9	#	0ddb
#	0DAA	##	0DDC
#	0DAB	###	0DDD
#	0DAC	##	0DDE
#	0DAD	#	0DDF
#	0DAE	#	0DF2

슬로바키아어 문자 집합

슬로바키아어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
á	00E1	ň	0148

문자	코드	문자	코드
ä	00E4	ó	00F3
č	010D	ô	00F4
d'	010F	í	0155
é	00E9	š	0161
í	00ED	td	0165
í	013A	ú	00FA
ĭ	013E	ý	00FD
ž	017E		

슬로베니아어 문자 집합

슬로베니아어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
č	010D	š	0161
ž	017E		

소말리아어 문자 집합

소말리아어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
s	0073	d	0064
t	0074	a	0061
a	0061	r	0072
n	006E	d	0064

스페인어 문자 집합

스페인어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- A - Z
- ' (아포스트로피)
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
Á	00C1	á	00E1
É	00C9	é	00E9
Í	00CD	í	00ED

문자	코드	문자	코드
Ó	00D3	ó	0XF3
Ú	00DA	ú	00FA
Ñ	00D1	ñ	0XF1
ü	00FC		

순다어 문자 집합

순다어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
s	0073	d	0064
t	0074	a	0061
a	0061	r	0072
n	006E	d	0064

스와힐리어 문자 집합

스와힐리어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
s	0073	d	0064
t	0074	a	0061
a	0061	r	0072
n	006E	d	0064

스웨덴어 문자 집합

스웨덴어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- A - Z
- ' (아포스트로피)
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
Ä	00C4	ä	00E4
Å	00C5	å	00E5
Ö	00D6	ö	00F6

타갈로그어/필리핀어 문자 집합

타갈로그어/필리핀어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z

- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드		
ñ	00F1		

타밀어 문자 집합

타밀어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

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

문자	코드	문자	코드
◌̣	0B83	◌̤	0BCD
ஃ	0B95	஄	0BBE
அ	0B99	ஆ	0BBF
இ	0B9A	ஈ	0BC0
ஊ	0B9E	஋	0BC1
஌	0B9F	஍	0BC2
எ	0BA3	ஏ	0BC6
஑	0BA4	ஒ	0BC7
ஒ	0BA8	ஓ	0BC8
ஐ	0BAA	ஐ	0BCA
ஓ	0BAE	ஔ	0BCB
ஔ	0BAF	க	0BCC

타타르어 문자 집합

타타르어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
a	0430	ӑ	0459

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

문자	코드	문자	코드
x	0445	х	04B3
ц	0446	ц	04B5
ч	0447	ч	04B7
ш	0448	h	04BB
щ	0449	е	04BD
ъ	044A	ё	04BF
ы	044B	#	04CA
ь	044C	ă	04D1
э	044D	ä	04D3
ю	044E	ě	04D7
я	044F	ə	04D9
#	0450	з	04E1
ë	0451	й	04E3
ђ	0452	ö	04E7
í	0453	ө	04E9
є	0454	ȳ	04EF
s	0455	ÿ	04F1
i	0456	ý	04F3
ï	0457	#	04F7
j	0458	Ы	04F9

텔루구어 문자 집합

텔루구어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

문자	코드	문자	코드
-	002D	త	0C24
ఁ	0C01	థ	0C25
ం	0C02	ద	0C26
ః	0C03	ధ	0C27
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
ఌ	0C30	ఎ	0C0E
఍	0C31	ఏ	0C0F
అ	0C32	ఐ	0C10
ఇ	0C33	ఒ	0C12
ఋ	0C35	ఓ	0C13
శ	0C36	ఔ	0C14
ష	0C37	క	0C15

문자	코드	문자	코드
స	0C38	ఖ	0C16
హ	0C39	గ	0C17
౧	0C3E	ఫ	0C18
౨	0C3F	బ	0C19
౩	0C40	చ	0C1A
౪	0C41	ఛ	0C1B
౫	0C42	జ	0C1C
౬	0C43	ఝ	0C1D
౭	0C44	ఞ	0C1E
౮	0C47	ట	0C1F
౯	0C48	థ	0C20
౧౦	0C4A	డ	0C21
౧౧	0C4B	ణ	0C22
౧౨	0C4C	త	0C23
౧౩	0C4D		

태국어 문자 집합

태국어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
ก	0E01	ล	0E25
ข	0E02	ฬ	0E26
ฃ	0E03	ว	0E27
ค	0E04	ศ	0E28
ฅ	0E05	ษ	0E29
ฆ	0E06	ส	0E2A
ง	0E07	ห	0E2B
จ	0E08	ฬ	0E2C
ฉ	0E09	อ	0E2D
ช	0E0A	ฮ	0E2E
ฌ	0E0B	ๆ	0E2F
ฌ	0E0C	ะ	0E30
ญ	0E0D	เ	0E31
ฎ	0E0E	า	0E32
ฏ	0E0F	อ	0E34
ฐ	0E10	อ	0E35
ฑ	0E11	อ	0E36
ฒ	0E12	อ	0E37
ณ	0E13	อ	0E38
ด	0E14	น	0E39

문자	코드	문자	코드
ඳ	0E15	.	0E3A
ඒ	0E16	෧	0E40
උ	0E17	෨	0E41
ඌ	0E18	෪	0E42
ඍ	0E19	෫	0E43
ඎ	0E1A	෬	0E44
ඏ	0E1B	෭	0E45
ඐ	0E1C	෮	0E46
එ	0E1D	෯	0E47
ඒ	0E1E	෰	0E48
උ	0E1F	෱	0E49
ඌ	0E20	෴	0E4A
ඍ	0E21	෵	0E4B
ඎ	0E22	෶	0E4C
ඏ	0E23	෷	0E4D
ඐ	0E24		

터키어 문자 집합

터키어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- A - Z

- ' (아포스트로피)
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
Ç	00C7	ö	00F6
Ö	00D6	û	00FB
Ü	00DC	ü	00FC
â	00E2	Ĝ	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		

우크라이나어 문자 집합

우크라이나어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

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

위구르어 문자 집합

위구르어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

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

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

문자	코드	문자	코드
o	0647	o	0693
o	0695		

우즈베키스탄어 문자 집합

우즈베키스탄어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

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

문자	코드	문자	코드
з	0437	л	043B
x	0445	ф	0444
ц	0446	к	043A
м	043C	у	0443
ь	044C	ж	0436
ъ	044A	ф	0493
a	0430		

베트남어 문자 집합

Amazon Transcribe는 번호를 사용하여 베트남어의 여섯 가지 톤을 나타냅니다. 다음 테이블은 톤 표시가 'ma' 단어에 매핑되는 방법을 보여줍니다.

톤 이름	톤 표시	톤 번호
ngang	ma	ma1
sắc	má	ma2
huyền	mà	ma3
hỏi	mả	ma4
ngã	mã	ma5
nặng	mạ	ma6

베트남어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- A - Z

- ' (아포스트로피)
- - (하이픈)
- . (마침표)
- & (앰퍼샌드)
- ; (세미콜론)
- _ (밑줄)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
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

문자	코드	문자	코드
ú	00FA	Ú	00DA
ý	00FD	Ý	00DD
ă	0103	Ă	0102
đ	0111	Đ	0110
ĩ	0129	Ĩ	0128
ũ	0169	Ũ	0168
ơ	01A1	Ơ	01A0
ư	01B0	Ư	01AF
ạ	1EA1	Ạ	1EA0
ă	1EA3	Ă	1EA2
ă	1EA5	ă	1EA4
ă	1EA7	Ă	1EA6
ă	1EA9	Ă	1EA8
ă	1EAB	Ă	1EAA
ậ	1EAD	Ậ	1EAC
ă	1EAF	Ă	1EAE
ă	1EB1	Ă	1EB0
ă	1EB3	Ă	1EB2
ă	1EB5	Ă	1EB4
ă	1EB7	Ă	1EB6

문자	코드	문자	코드
ẹ	1EB9	Ẹ	1EB8
ě	1EBB	Ě	1EBA
ẽ	1EBD	Ẽ	1EBC
ế	1EBF	Ế	1EBE
ề	1EC1	Ề	1EC0
ễ	1EC3	Ễ	1EC2
ẽ	1EC5	Ẽ	1EC4
ệ	1EC7	Ệ	1EC6
ỉ	1EC9	Ỉ	1EC8
ị	1ECB	Ị	1ECA
ọ	1ECD	Ọ	1ECC
ỏ	1ECF	Ỏ	1ECE
ố	1ED1	Ố	1ED0
ồ	1ED3	Ồ	1ED2
ỗ	1ED5	Ỗ	1ED4
ỗ	1ED7	Ỗ	1ED6
ộ	1ED9	Ộ	1ED8
ớ	1EDB	Ớ	1EDA
ờ	1EDD	Ờ	1EDC
ở	1EDF	Ở	1EDE

문자	코드	문자	코드
õ	1EE1	Õ	1EE0
ø	1EE3	Ø	1EE2
ұ	1EE5	Ұ	1EE4
ű	1EE7	Ű	1EE6
ú	1EE9	Ú	1EE8
ù	1EEB	Ù	1EEA
ű	1EED	Ű	1EEC
ű	1EEF	Ű	1EEE
ұ	1EF1	Ұ	1EF0
ỳ	1EF3	Ỳ	1EF2
ყ	1EF5	ყ	1EF4
ỷ	1EF7	Ỳ	1EF6
ỹ	1EF9	Ỳ	1EF8

웨일스어 문자 집합

웨일스어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
à	00E0	ò	00F2
á	00E1	ó	00F3
â	00E2	ô	00F4
ä	00E4	ö	00F6
è	00E8	ù	00F9
é	00E9	ú	00FA
ê	00EA	û	00FB
ë	00EB	ü	00FC
ì	00EC	ý	00FD
í	00ED	ÿ	00FF
î	00EE	ŵ	0175
ï	00EF	ÿ	0177
ÿ	1EF3		

월로프어 문자 집합

월로프어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
à	00E0	ê	00EA
ã	00E3	ë	00EB
ç	00E7	ñ	00F1
è	00E8	ó	00F3
é	00E9	ô	00F4
η	014B		

줄루어 문자 집합

줄루어 사용자 지정 어휘의 경우 Phrase 필드에 다음 문자를 사용할 수 있습니다.

- a - z
- - (하이픈)
- . (마침표)

Phrase 필드에 다음 유니코드 문자도 사용할 수 있습니다.

문자	코드	문자	코드
s	0073	d	0064
t	0074	a	0061
a	0061	r	0072
n	006E	d	0064

Amazon Transcribe 작동 방법

Amazon Transcribe는 기계 학습 모델을 사용하여 음성을 텍스트로 변환합니다.

트랜스크립트에는 트랜스크립션된 텍스트 외에도 트랜스크립션된 콘텐츠에 대한 데이터가 포함되며, 여기에는 각 단어 또는 구두점에 대한 신뢰도 점수 및 타임스탬프 등이 있습니다. 출력 예시를 보려면 [데이터 입력 및 출력](#) 섹션을 참조하세요. 트랜스크립션에 적용할 수 있는 전체 기능 목록은 [기능 요약](#)을 참조하세요.

트랜스크립션 메서드는 두 가지 주요 범주로 나눌 수 있습니다.

- 배치 트랜스크립션: Amazon S3 버킷에 업로드된 미디어 파일을 트랜스크립션합니다. 배치 트랜스크립션에는 [AWS CLI](#), [AWS Management Console](#) 및 다양한 [AWS SDK](#)를 사용할 수 있습니다.
- 스트리밍 트랜스크립션: 미디어 스트림을 실시간으로 트랜스크립션합니다. 스트리밍 트랜스크립션에는 [AWS Management Console](#), [HTTP/2](#), [WebSocket](#) 및 다양한 [AWS SDK](#)를 사용할 수 있습니다.

단, 배치 트랜스크립션과 스트리밍 트랜스크립션의 기능 및 언어 지원은 다릅니다. 자세한 내용은 [Amazon Transcribe features](#) 및 [지원되는 언어](#)를 참조하세요.

주제

- [데이터 입력 및 출력](#)
- [번호 및 구두점 트랜스크립션](#)

 시작하는 데 도움을 주는 API 작업

배치: [StartTranscriptionJob](#)

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

데이터 입력 및 출력

Amazon Transcribe는 오디오 데이터를 Amazon S3 버킷 또는 미디어 스트림의 미디어 파일로 가져와 텍스트 데이터로 변환합니다.

Amazon S3 버킷에 저장된 미디어 파일을 트랜스크립션하는 경우 배치 트랜스크립션을 수행하는 것입니다. 미디어 스트림을 트랜스크립션하는 경우 스트리밍 트랜스크립션을 수행하는 것입니다. 이 두 프로세스에는 다른 규칙과 요구 사항이 있습니다.

배치 트랜스크립션을 사용하면 모든 트랜스크립션 작업을 동시에 처리할 필요가 없는 경우 [작업 대기열](#)을 사용할 수 있습니다. 이렇게 하면 Amazon Transcribe에서 트랜스크립션 작업을 추적하고 슬롯을 사용할 수 있을 때 이를 처리할 수 있습니다.

Note

Amazon Transcribe는 분석 모델의 품질을 지속적으로 개선하기 위해 귀하의 콘텐츠를 일시적으로 저장할 수 있습니다. 자세한 정보는 [Amazon Transcribe FAQ](#)를 참조하세요. Amazon Transcribe에서 저장했을 수 있는 콘텐츠의 삭제를 요청하려면 [지원](#)에서 사례를 개설하시기 바랍니다.

주제

- [미디어 형식](#)
- [오디오 채널](#)
- [샘플 속도](#)
- [출력](#)

미디어 형식

지원되는 미디어 유형은 배치 트랜스크립션과 스트리밍 트랜스크립션 간에 다르지만 둘 다 무손실 형식을 사용하는 것이 좋습니다. 자세한 내용은 다음 테이블을 참조하세요.

	배치	스트리밍
지원되는 형식	• AMR • FLAC • M4A • MP3 • MP4 • Ogg • WebM • WAV	• FLAC • Ogg Opus • PCM 인코딩
권장 형식	• FLAC	• FLAC

	배치	스트리밍
	PCM 16비트 인코딩을 사용하는 WAV	PCM 서명된 16비트 리틀 엔디안 오디오(WAV는 포함되지 않음)

최상의 결과를 얻으려면 PCM 16비트 인코딩을 사용하는 무손실 형식(예: FLAC 또는 WAV)을 사용하세요.

Note

일부 언어에서는 스트리밍 트랜스크립션이 지원되지 않습니다. 자세한 내용은 [지원되는 언어 테이블](#)의 '데이터 입력' 열을 참조하세요.

오디오 채널

Amazon Transcribe는 단일 채널 및 이중 채널 미디어를 지원합니다. 채널이 2개가 넘는 미디어는 현재 지원되지 않습니다.

오디오의 한 채널에 여러 화자가 포함되어 있으며 트랜스크립션 출력에서 각 화자를 파티셔닝하여 레이블을 지정하려는 경우 [화자 파티셔닝\(분할\)](#)을 사용할 수 있습니다.

오디오의 별도의 두 채널에 음성이 포함되어 있는 경우 [채널 식별](#)을 사용하여 트랜스크립트 내에서 각 채널을 개별적으로 트랜스크립션할 수 있습니다.

이 두 옵션 모두 하나의 트랜스크립트 파일을 생성합니다.

Note

[화자 파티셔닝](#) 또는 [채널 식별](#)을 활성화하지 않으면 트랜스크립트 텍스트가 하나의 연속 섹션으로 제공됩니다.

샘플 속도

배치 트랜스크립션 작업의 경우 샘플 속도를 제공하도록 선택할 수 있지만 이 파라미터는 선택 사항입니다. 요청에 포함하는 경우 제공하는 값이 오디오의 실제 샘플 속도와 일치하는지 확인하세요. 오디오와 일치하지 않는 샘플 속도를 제공하면 작업이 실패할 수 있습니다.

스트리밍 트랜스크립션의 경우 요청에 샘플 속도를 포함해야 합니다. 배치 트랜스크립션 작업과 마찬가지로 제공하는 값이 오디오의 실제 샘플 속도와 일치하는지 확인하세요.

전화 녹음과 같은 저충실도 오디오의 샘플 속도는 일반적으로 8,000Hz를 사용합니다. 고충실도 오디오의 경우 Amazon Transcribe는 16,000Hz에서 48,000Hz 사이의 값을 지원합니다.

출력

트랜스크립션 출력은 JSON 형식입니다. 스크립트의 첫 부분에는 트랜스크립트 자체가 단락 형식으로 포함되며 그 뒤에 모든 단어와 구두점에 대한 추가 데이터가 포함됩니다. 제공되는 데이터는 요청에 포함된 기능에 따라 달라집니다. 트랜스크립트에는 최소한 모든 단어의 시작 시간, 종료 시간 및 신뢰도 점수가 포함됩니다. [다음 섹션](#)에는 추가 옵션이나 기능이 포함되지 않은 기본 트랜스크립션 요청의 출력 예시가 나와 있습니다.

모든 배치 트랜스크립트는 Amazon S3 버킷에 저장됩니다. 스크립트를 자체 Amazon S3 버킷에 저장하거나 Amazon Transcribe에서 보안 기본 버킷을 사용하도록 선택할 수 있습니다. Amazon S3 버킷 생성 및 사용에 대해 자세히 알아보려면 [버킷 작업](#)을 참조하세요.

소유한 Amazon S3 버킷에 트랜스크립트를 저장하려면 트랜스크립션 요청에 버킷의 URI를 지정하세요. 배치 트랜스크립션 작업을 시작하기 전에 이 버킷에 대한 Amazon Transcribe 쓰기 권한을 부여해야 합니다. 자체 버킷을 지정하는 경우 버킷을 제거할 때까지 트랜스크립트가 해당 버킷에 남아 있습니다.

Amazon S3 버킷을 지정하지 않는 경우 Amazon Transcribe는 보안 서비스 관리형 버킷을 사용하며 스크립트를 다운로드하는 데 사용할 수 있는 임시 URI를 제공합니다. 임시 URI는 15분간 유효하다는 점에 유의하세요. 제공된 URI를 사용할 때 AccessDenied 오류가 발생하는 경우 트랜스크립트에 사용할 새 임시 URI를 GetTranscriptionJob 요청하세요.

기본 버킷을 선택하면 작업이 완료될 때(90일) 트랜스크립트가 삭제됩니다. 이 만료일을 지나 트랜스크립트를 보관하려면 다운로드해야 합니다.

스트리밍 스크립트는 스트림에 사용하는 것과 동일한 메서드를 통해 반환됩니다.

Tip

JSON 출력을 Word 형식의 단계별 트랜스크립트로 변환하려면 이 [GitHub 예시\(Python3용\)](#)를 참조하세요. 이 스크립트는 통화 후 분석 트랜스크립트 및 분할이 활성화된 표준 배치 트랜스크립트와 함께 작동합니다.

출력 예시

트랜스크립트는 단락 형식으로 완전한 트랜스크립션을 제공하며, 이어 등장하는 단어별 분석을 통해 모든 단어와 구두점에 대한 데이터를 제공합니다. 여기에는 시작 시간, 종료 시간, 신뢰도 점수 및 유형 (pronunciation 또는 punctuation)이 포함됩니다.

다음 예시는 [추가 기능](#)이 포함되지 않은 단순한 배치 트랜스크립션 작업의 예입니다. 트랜스크립션 요청에 추가 기능을 적용할 때마다 트랜스크립션 출력 파일에 추가 데이터가 추가됩니다.

기본 배치 트랜스크립트에는 두 개의 메인 섹션이 있습니다.

1. transcripts: 전체 트랜스크립트가 하나의 텍스트 블록에 들어 있습니다.
2. items: transcripts 섹션의 각 단어 및 구두점에 대한 정보가 들어 있습니다.
3. audio_segments: 오디오 세그먼트는 중단 없는 음성 언어가 포함된 오디오 녹음의 특정 부분으로, 일시 중지 또는 중단이 최소화됩니다. 이 세그먼트는 자연적인 음성 흐름을 캡처하고 시작 시간 및 종료 시간으로 audio_segments에 포착됩니다. 오디오 세그먼트 내의 items 요소는 세그먼트 내의 각 항목에 해당하는 식별자 시퀀스입니다.

트랜스크립션 요청에 포함시키는 각 추가 기능은 트랜스크립트에 추가 정보를 생성합니다.

```
{
  "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"
}

```

번호 및 구두점 트랜스크립션

Amazon Transcribe는 지원되는 모든 언어에 자동으로 구두점을 추가하고 표기 체계에서 대/소문자를 구분하는 언어에 맞게 단어를 대문자로 표시합니다.

대부분의 언어에서 숫자는 단어 형태로 트랜스크립션됩니다. 그러나 번호 트랜스크립션을 지원하는 언어의 경우는 사용되는 컨텍스트에 따라 번호를 다르게 Amazon Transcribe 처리합니다.

예를 들어 화자가 "Meet me at eight-thirty AM on June first at one-hundred Main Street with three-dollars-and-fifty-cents and one-point-five chocolate bars,"라고 말하면 다음과 같이 트랜스크립션됩니다.

- 숫자 트랜스크립션이 지원되는 언어: Meet me at 8:30 a.m. on June 1st at 100 Main Street with \$3.50 and 1.5 chocolate bars
- 기타 모든 언어: 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

번호 트랜스크립션을 지원하는 언어를 보려면 [지원되는 언어 및 언어별 기능](#) 섹션을 참조하세요.

시작하기 Amazon Transcribe

트랜스크립션을 생성하려면 다음과 같은 몇 가지 사전 조건이 있습니다.

- [AWS 계정에 가입](#)
- [AWS CLI 및 SDKs 설치](#)(트랜스크립션에 AWS Management Console 를 사용하는 경우 이 단계를 건너뛸 수 있음)
- [IAM 보안 인증 구성](#)
- [Amazon S3 버킷 설정](#)
- [IAM 정책 생성](#)

이러한 사전 조건을 완료하면 트랜스크립션할 준비가 된 것입니다. 시작하려면 다음 목록에서 원하는 트랜스크립션 메서드를 선택합니다.

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

Tip

를 처음 Amazon Transcribe 사용하거나 기능을 탐색하려는 경우를 사용하는 것이 좋습니다 [AWS Management Console](#). 컴퓨터 마이크를 사용하여 스트림을 시작하려는 경우 가장 쉬운 방법이기도 합니다.

HTTP/2 및 WebSocket을 사용한 스트리밍은 다른 트랜스크립션 메서드보다 복잡하므로 이러한 메서드를 시작하기 전에 [스트리밍 트랜스크립션 설정](#) 섹션을 검토하는 것이 좋습니다. 스트리밍 트랜스크립션에는 SDK를 사용하는 것이 좋다는 점에 유의하세요.

에 가입 AWS 계정

[프리 티어](#) 계정 또는 [유료 계정](#)에 가입할 수 있습니다. 두 옵션 모두 모든에 액세스할 수 있습니다 AWS 서비스. 프리 티어에는 사용량을 탐색 AWS 서비스 하고 추정할 수 있는 평가판 기간이 있습니다. 평가

판 기간이 만료되면 유료 계정으로 마이그레이션할 수 있습니다. 요금은 사용량에 따라 지불하는 방식으로 부과됩니다. 자세한 내용은 [Amazon Transcribe 요금](#)을 참조하세요.

 Tip

계정을 설정할 때 IAM 엔터티를 생성하는 데 필요하므로 AWS 계정 ID를 기록해 둡니다.

AWS CLI 및 SDKs 설치

Amazon Transcribe API를 사용하려면 먼저 설치해야 합니다 AWS CLI. 현재 AWS CLI 는 버전 2입니다. [Linux](#), [Mac](#), [Windows](#) 및 [Docker](#)에 대한 설치 지침은 사용 [AWS Command Line Interface 설명서](#)에서 찾을 수 있습니다.

를 AWS CLI 설치한 후에는 보안 자격 증명 및에 맞게 [구성](#)해야 합니다 AWS 리전.

SDK와 Amazon Transcribe 함께를 사용하려면 설치 지침을 위해 원하는 언어를 선택합니다.

- [.NET](#)
- [C++](#)
- [Go](#)
- [Java V2](#)
- [JavaScript](#)
- [PHP V3](#)
- [AWS SDK for Python \(Boto3\)](#)(배치 트랜스크립션)
- [Python](#)(스트리밍 트랜스크립션)
- [Ruby V3](#)
- [Rust](#)(배치 트랜스크립션)
- [Rust](#)(스트리밍 트랜스크립션)

IAM 자격 증명 구성

를 생성할 때 계정의 모든 AWS 서비스 및 리소스에 대한 완전한 액세스 권한이 있는 하나의 로그인 자격 증명으로 AWS 계정시작합니다. 이 자격 증명을 AWS 계정 루트 사용자라고 하며 계정을 생성하는 데 사용한 이메일 주소와 암호로 로그인하여 액세스합니다.

일상적인 작업에 루트 사용자를 사용하지 않을 것을 강력히 권장합니다. 루트 사용자 보안 인증 정보를 보호하고 루트 사용자만 수행할 수 있는 작업을 수행하는 데 사용합니다.

가장 좋은 방법은 관리자 액세스가 필요한 사용자를 포함한 사용자에게 자격 증명 공급자와의 페더레이션을 사용하여 임시 자격 증명을 사용하여 AWS 서비스에 액세스하도록 요구하는 것입니다.

페더레이션 자격 증명은 자격 증명 소스를 통해 제공된 자격 증명을 사용하여 AWS 서비스에 액세스하는 모든 사용자입니다. 페더레이션 ID는 AWS 계정에 액세스할 때 역할을 수임하고 역할은 임시 자격 증명을 제공합니다.

중앙 집중식 액세스 관리를 위해 [AWS IAM Identity Center](#)을 사용하는 것이 좋습니다. IAM Identity Center에서 사용자 및 그룹을 생성할 수 있습니다. 또는 모든 및 애플리케이션에서 사용할 수 있도록 자체 ID 소스의 사용자 AWS 계정 및 그룹 집합에 연결하고 동기화할 수 있습니다. 자세한 내용은 [용 자격 증명 및 액세스 관리 Amazon Transcribe](#) 단원을 참조하십시오.

IAM 모범 사례에 대해 자세히 알아보려면 [의 보안 모범 사례를 IAM](#) 참조하세요.

Amazon S3 버킷 생성

Amazon S3 는 안전한 객체 스토리지 서비스입니다. 는 파일(객체라고 함)을 컨테이너(버킷이라고 함)에 Amazon S3 저장합니다.

배치 트랜스크립션을 실행하려면 먼저 미디어 파일을 Amazon S3 버킷에 업로드해야 합니다. 트랜스크립션 출력에 Amazon S3 버킷을 지정하지 않으면 임시 AWS관리형 Amazon S3 버킷에 트랜스크립트를 Amazon Transcribe 입력합니다. AWS관리형 버킷의 트랜스크립션 출력은 90일 후에 자동으로 삭제됩니다.

[첫 S3 버킷 생성](#) 및 [버킷에 객체 업로드](#) 방법을 알아보세요.

IAM 정책 생성

에서 액세스를 관리하려면 정책을 생성하고 IAM 자격 증명(사용자, 그룹 또는 역할) 또는 AWS 리소스에 연결해야 AWS합니다. 정책은 연결된 엔터티의 권한을 정의합니다. 예를 들어, 액세스 권한을 부여하는 정책을 해당 역할에 연결한 경우에만 역할은 Amazon S3 버킷에 있는 미디어 파일에 액세스할 수 있습니다. 해당 역할을 추가로 제한하려는 경우 버킷 내의 특정 파일에 대한 액세스를 제한할 수 있습니다 Amazon S3 .

AWS 정책 사용에 대한 자세한 내용은 다음을 참조하세요.

- [의 정책 및 권한 IAM](#)

- [IAM 정책 생성](#)
- [Amazon Transcribe 에서 IAM을 사용하는 방법](#)

와 함께 사용할 수 있는 정책 예제는 단원을 Amazon Transcribe참조하십시오 [Amazon Transcribe 자격 증명 기반 정책 예제](#). 사용자 지정 정책을 생성하려면 [AWS 정책 생성기](#)를 사용해 보세요.

AWS Management Console AWS CLI또는 AWS SDK를 사용하여 정책을 추가할 수 있습니다. 지침은 자격 [IAM 증명 권한 추가 및 제거](#)를 참조하세요.

정책 형식은 다음과 같습니다.

JSON

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

Amazon 리소스 이름(ARNs Amazon S3 버킷과 같은 모든 AWS 리소스를 고유하게 식별합니다. 정책에서 ARN을 사용하여 특정 작업에 특정 리소스를 사용할 권한을 부여할 수 있습니다. 예를 들어 Amazon S3 버킷 및 하위 폴더에 대한 읽기 액세스 권한을 부여하려면 신뢰 정책의 Statement 섹션에 다음 코드를 추가할 수 있습니다.

```
{
  "Effect": "Allow",
  "Action": [
    "s3:GetObject",
    "s3:ListBucket"
  ]
}
```

```

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

```

다음은 Amazon S3 버킷, amzn-s3-demo-bucket 및 하위 폴더에 Amazon Transcribe 읽기 (GetObject, ListBucket) 및 쓰기(PutObject) 권한을 부여하는 정책의 예입니다.

JSON

```

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

```

로 트랜스크립션 AWS Management Console

AWS 콘솔을 사용하여 배치 및 스트리밍 트랜스크립션을 수행할 수 있습니다. Amazon S3 버킷에 있는 미디어 파일을 트랜스크립션하는 경우 배치 트랜스크립션을 수행하는 것입니다. 오디오 데이터의 실시간 스트림을 트랜스크립션하는 경우 스트리밍 트랜스크립션을 수행하는 것입니다.

배치 트랜스크립션을 시작하기 전에 먼저 미디어 파일을 Amazon S3 버킷에 업로드해야 합니다. 를 사용하여 트랜스크립션을 스트리밍 AWS Management Console하려면 컴퓨터 마이크를 사용해야 합니다.

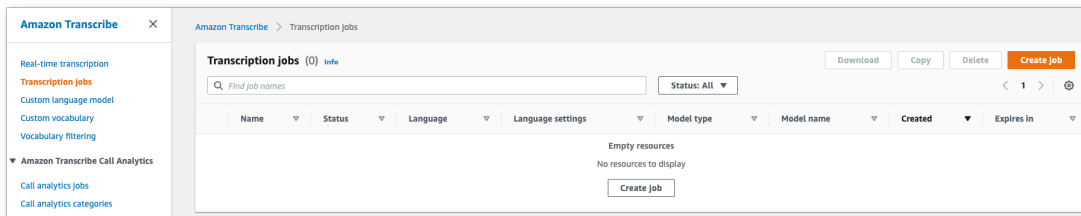
지원되는 미디어 형식과 기타 미디어 요구 사항과 제약 조건을 보려면 [데이터 입력 및 출력](#)을 참조하세요.

각 트랜스크립션 메서드에 대한 간략한 설명을 보려면 다음 섹션을 확장하세요.

배치 트랜스크립션

먼저 트랜스크립션하려는 미디어 파일을 Amazon S3 버킷에 업로드했는지 확인합니다. 방법을 잘 모르겠으면 Amazon S3 사용 설명서: [버킷에 객체 업로드](#)를 참조하세요.

1. [AWS Management Console](#)에서 왼쪽 탐색 창의 훈련 작업을 선택합니다. 이렇게 하면 트랜스크립션 작업 목록으로 이동합니다.



작업 생성을 선택합니다.

2. 작업 세부 정보 지정 페이지의 필드를 작성합니다.

Specify job details [Info](#)

Job settings

Name

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

Model type [Info](#)

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

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

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

Language settings

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

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

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

Language

Choose the language of the input audio.

► **Additional settings**

입력 위치는 Amazon S3 버킷 내의 객체여야 합니다. 출력 위치의 경우 안전한 Amazon S3 서비스 관리형 버킷을 선택하거나 자체 Amazon S3 버킷을 지정할 수 있습니다.

서비스 관리형 버킷을 선택하면 AWS Management Console 에서 트랜스크립트 미리보기를 볼 수 있고 작업 세부 정보 페이지에서 트랜스크립트를 다운로드할 수 있습니다 (아래 참조).

자체 Amazon S3 버킷을 선택하면에서 미리 보기를 볼 수 AWS Management Console 없으며 트랜스크립트를 다운로드하려면 Amazon S3 버킷으로 이동해야 합니다.

Input data [Info](#)

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

Browse S3

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

Output data

Output data location type info [Info](#)

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

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

Subtitle file format [Info](#)

☐ SRT (SubRip)

☐ VTT (WebVTT)

Tags - optional

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

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel

Next

다음을 선택합니다.

- 작업 구성 페이지에서 원하는 옵션을 선택합니다. 트랜스크립션과 함께 [사용자 지정 어휘](#) 또는 [사용자 지정 언어 모델](#)을 사용하려는 경우 트랜스크립션 작업을 시작하기 전에 생성해야 합니다.

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

Audio settings

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

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

Content removal

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

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

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

Customization

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

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

작업 생성을 선택합니다.

- 이제 트랜스크립션 작업 페이지가 열렸습니다. 여기에서 트랜스크립션 작업의 상태를 확인할 수 있습니다. 완료되면 트랜스크립션을 선택합니다.

Transcription jobs (5) Info

Find job names

Status: All

Download

Copy

Delete

Create Job

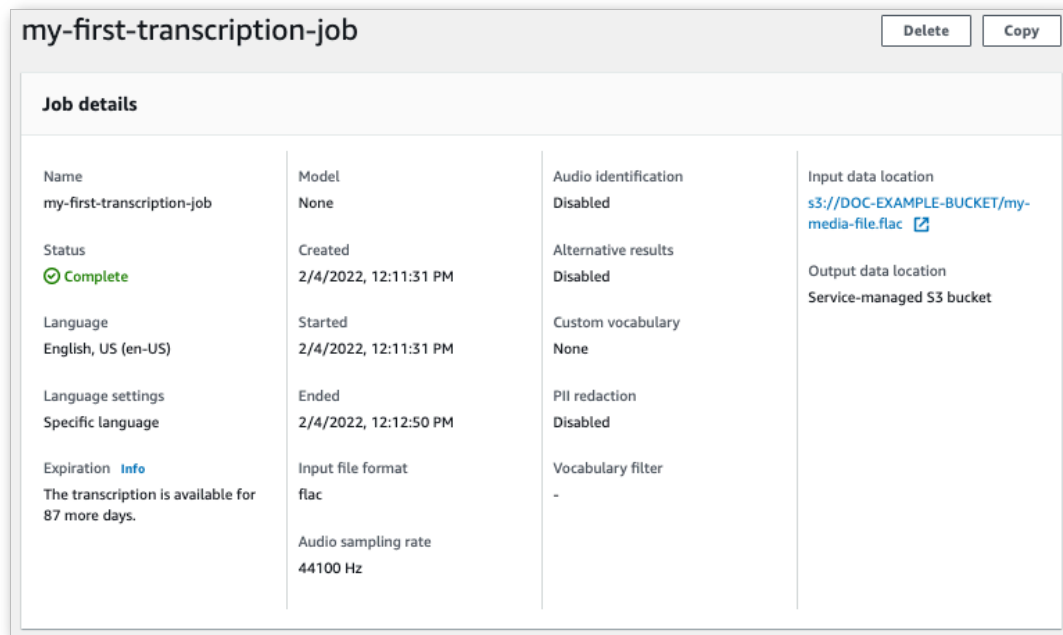
<1>

⌕

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

- 이제 트랜스크립션에 대한 작업 세부 정보 페이지를 볼 수 있습니다. 여기에서 트랜스크립션 작업을 설정할 때 지정한 모든 옵션을 볼 수 있습니다.

트랜스크립트를 보려면 출력 데이터 위치 아래의 오른쪽 열에서 연결된 파일 경로를 선택합니다. 그러면 지정한 Amazon S3 출력 폴더로 이동합니다. 이제 확장명이 .json인 출력 파일을 선택합니다.



6. 트랜스크립트를 다운로드하는 방법은 서비스 관리형 Amazon S3 버킷을 선택했는지 아니면 자체 Amazon S3 버킷을 선택했는지에 따라 달라집니다.
 - a. 서비스 관리형 버킷을 선택한 경우 트랜스크립션 작업 정보 페이지에 다운로드 버튼과 함께 트랜스크립션 미리 보기 창이 표시됩니다.

my-first-transcription-job

Delete

Copy

Job details

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

Transcription preview

Download ▾

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

Text

Audio identification

Subtitles

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

다운로드를 선택하고 트랜스크립트 다운로드를 선택합니다.

- b. 자체 Amazon S3 버킷을 선택한 경우 트랜스크립션 작업 정보 페이지의 트랜스크립션 미리 보기 창에 텍스트가 표시되지 않습니다. 대신 선택한 Amazon S3 버킷에 대한 링크가 포함된 파란색 정보 상자가 표시됩니다.

my-first-transcription-job

DeleteCopy

Job details

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

Transcription preview

Select download to save a local copy of the transcription.

Download

Text

Audio identification

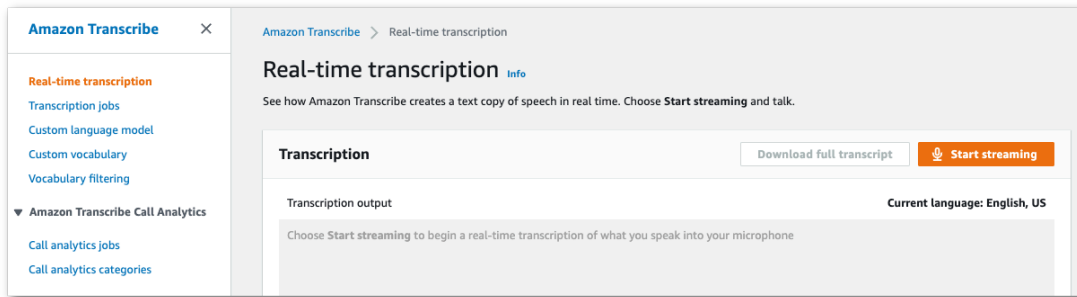
Subtitles

When you use your own S3 bucket for transcription output, Amazon Transcribe does not show the output in the console. You open the output file from your S3 Bucket.

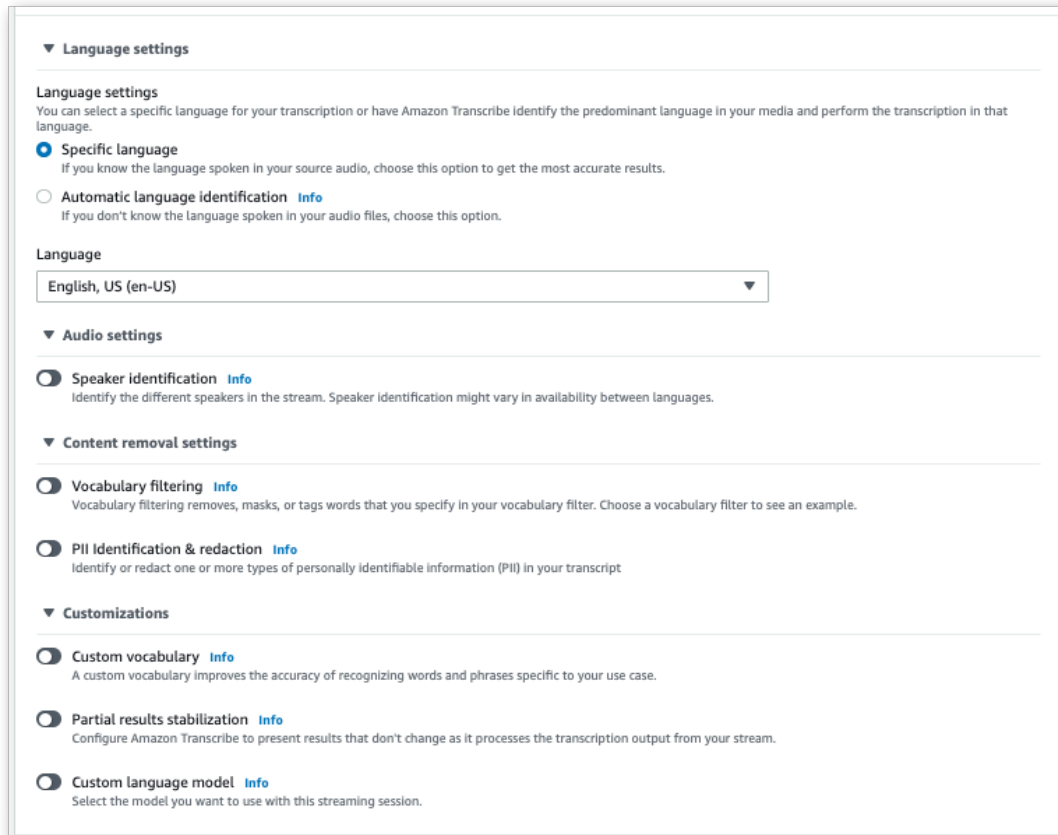
트랜스크립트에 액세스하려면 작업 세부 정보 창의 출력 데이터 위치에 있는 링크 또는 트랜스크립션 미리 보기 창의 파란색 정보 상자 내의 S3 버킷 링크를 Amazon S3 사용하여 지정된 버킷으로 이동합니다.

스트리밍 트랜스크립션

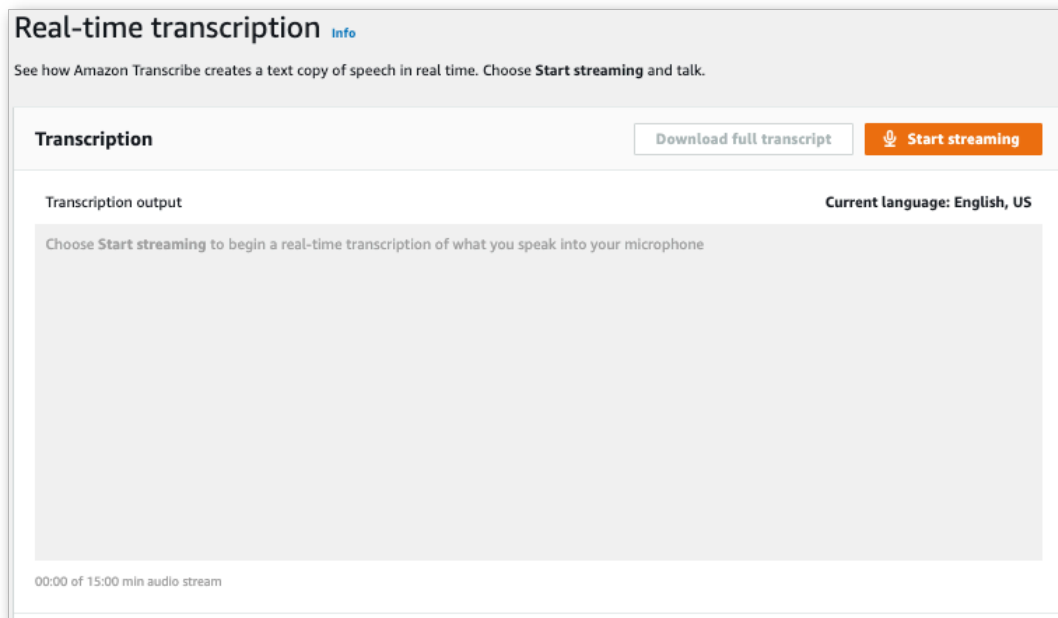
1. [AWS Management Console](#)에서 왼쪽 탐색 창의 실시간 트랜스크립션을 선택합니다. 그러면 스트리밍을 시작하기 전에 옵션을 선택할 수 있는 메인 스트리밍 페이지로 이동합니다.



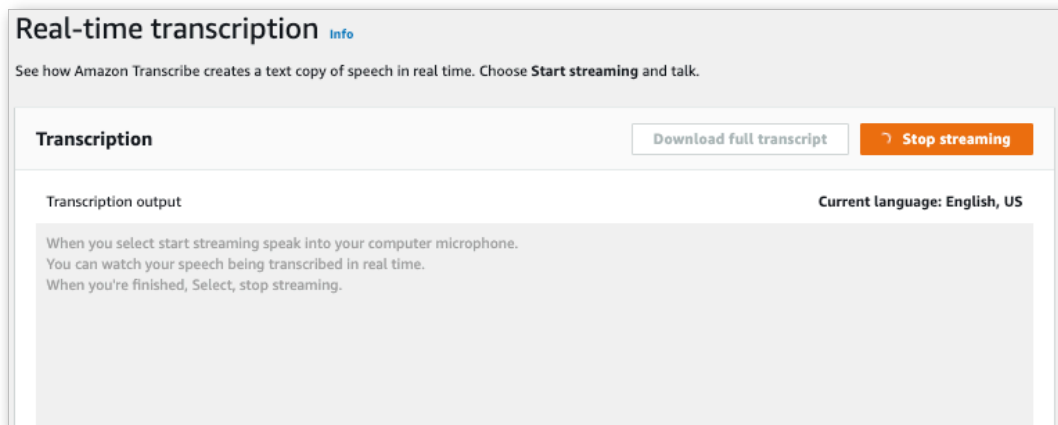
2. 트랜스크립션 출력 상자 아래에는 다양한 언어 및 오디오 설정을 선택할 수 있는 옵션이 있습니다.



3. 적절한 설정을 선택한 후 페이지 상단으로 스크롤하여 스트리밍 시작을 선택한 다음 컴퓨터 마이크에 대고 말하기 시작합니다. 음성이 트랜스크립션되는 것을 실시간으로 볼 수 있습니다.



4. 마쳤으면 스트리밍 중지를 선택합니다.



이제 전체 트랜스크립트 다운로드를 선택하여 트랜스크립트를 다운로드할 수 있습니다.

로 트랜스크립션 AWS CLI

AWS CLI 를 사용하여 트랜스크립션을 시작할 때 CLI 수준에서 모든 명령을 실행할 수 있습니다. 또는 사용하려는 명령을 실행한 다음 요청 본문이 포함된 JSON 파일의 AWS 리전 및 위치를 차례로 실행할 수 있습니다. 이 설명서의 예시에서는 두 가지 메서드를 모두 보여 주지만 이 섹션에서는 전자의 메서드를 중점적으로 다룹니다.

AWS CLI 는 스트리밍 트랜스크립션을 지원하지 않습니다.

계속하기 전에 다음을 완료했는지 확인하세요.

- 미디어 파일을 Amazon S3 버킷에 업로드했습니다. Amazon S3 버킷을 생성하거나 파일을 업로드하는 방법을 잘 모르는 경우 [첫 번째 Amazon S3 버킷 생성](#) 및 [버킷에 객체 업로드를 참조하세요](#).
- [AWS CLI](#)를 설치했습니다.

AWS CLI 명령 [AWS CLI 참조](#) Amazon Transcribe 에서에 대한 모든 명령을 찾을 수 있습니다.

새 트랜스크립션 작업 시작

새 트랜스크립션을 시작하려면 start-transcription-job 명령을 사용합니다.

1. 터미널 창에서 다음을 입력합니다.

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

다음 줄에 '>'가 표시되며, 이제 다음 단계에서 설명하는 대로 필수 파라미터를 계속 추가할 수 있습니다.

'\'를 생략하고 모든 파라미터를 스페이스로 구분하여 추가할 수도 있습니다.

2. start-transcription-job 명령의 경우 region, transcription-job-name, media 및 language-code 또는 identify-language 중 하나를 포함해야 합니다.

출력 위치를 지정하려면 요청에 output-bucket-name을 포함하고, 지정된 출력 버킷의 하위 폴더를 지정하려면 output-key도 포함합니다.

```
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
```

모든 파라미터를 추가하는 경우 이 요청은 다음과 같습니다.

```
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
```

output-bucket-name을 사용하여 출력 버킷을 지정하지 않기로 선택한 경우, Amazon Transcribe 는 트랜스크립션 출력을 서비스 관리형 버킷에 위치시킵니다. 서비스 관리형 버킷에 저장된 트랜스크립트는 90일 후에 만료됩니다.

Amazon Transcribe 는 다음과 같이 응답합니다.

```
{
  "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"
  }
}
```

[TranscriptionJobStatus](#)가 IN_PROGRESS에서 COMPLETED로 변경되면 트랜스크립션 작업이 성공한 것입니다. 업데이트된 [TranscriptionJobStatus](#)를 보려면 다음 섹션과 같이 get-transcription-job 또는 list-transcription-job 명령을 사용합니다.

트랜스크립션 작업의 상태 가져오기

트랜스크립션 작업에 대한 정보를 얻으려면 get-transcription-job 명령을 사용합니다.

이 명령에 필요한 유일한 파라미터는 작업이 AWS 리전 위치한과 작업 이름입니다.

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

Amazon Transcribe 는 다음과 같이 응답합니다.

```
{
  "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
    }
  }
}

```

트랜스크립션 출력에 대해 자체 Amazon S3 버킷을 선택한 경우 이 버킷은와 함께 나열됩니다 TranscriptFileUri. 서비스 관리형 버킷을 선택한 경우 임시 URI가 제공됩니다. 이 URI를 사용하여 트랜스크립트를 다운로드합니다.

Note

서비스 관리형 Amazon S3 버킷URIs는 15분 동안만 유효합니다. URI를 사용할 때 AccesDenied 오류가 발생하는 경우 get-transcription-job 요청을 다시 실행하여 새 임시 URI를 얻습니다.

트랜스크립션 작업 나열

지정된 모든 트랜스크립션 작업을 나열하려면 list-transcription-jobs 명령을 AWS 리전사 용합니다.

이 명령에 필요한 유일한 파라미터는 트랜스크립션 작업이 있는 AWS 리전 입니다.

```

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

```

Amazon Transcribe 는 다음과 같이 응답합니다.

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

트랜스크립션 작업 삭제

트랜스크립션 작업을 삭제하려면 `delete-transcription-job` 명령을 사용합니다.

이 명령에 필요한 유일한 파라미터는 작업이 AWS 리전 위치한과 작업 이름입니다.

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

`list-transcription-jobs` 명령을 실행하여 삭제 요청이 성공했는지 확인할 수 있습니다. 더 이상 목록에 작업이 표시되지 않아야 합니다.

AWS SDKs

배치 및 스트리밍 트랜스크립션 모두를 위해 SDK를 사용할 수 있습니다. Amazon S3 버킷에 있는 미디어 파일을 트랜스크립션하는 경우 배치 트랜스크립션을 수행합니다. 오디오 데이터의 실시간 스트림을 트랜스크립션하는 경우 스트리밍 트랜스크립션을 수행하는 것입니다.

사용할 수 있는 프로그래밍 언어 목록은 [섹션을 Amazon Transcribe참조하세요](#) [지원되는 프로그래밍 언어](#). 스트리밍 트랜스크립션은 모든 AWS SDKs에서 지원되지 않습니다. 지원되는 미디어 형식과 기타 미디어 요구 사항과 제약 조건을 보려면 [데이터 입력 및 출력](#)을 참조하세요.

사용 가능한 모든 AWS SDKs 및 빌더 도구에 대한 자세한 내용은 [빌드 기반 도구를 AWS](#) 참조하세요.

Tip

기능별, 시나리오 및 교차 서비스 예제를 포함하여 AWS SDKs를 사용하는 추가 예제는 [AWS SDKs를 사용한 Amazon Transcribe의 코드 예제](#)장을 참조하세요.

SDK 코드 샘플은 다음 GitHub 리포지토리에서도 찾을 수 있습니다.

- [AWS 코드 예제](#)
- [Amazon Transcribe 예시](#)

배치 트랜스크립션

Amazon S3 버킷에 있는 미디어 파일의 URI를 사용하여 배치 트랜스크립션을 생성할 수 있습니다.

Amazon S3 버킷을 생성하거나 파일을 업로드하는 방법을 잘 모르는 경우 [첫 번째 S3 버킷 생성 및 버킷에 객체 업로드를 참조하세요](#).

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()

```

스트리밍 트랜스크립션

스트리밍된 미디어 파일 또는 라이브 미디어 스트림을 사용하여 스트리밍 트랜스크립션을 생성할 수 있습니다.

스트리밍에는 표준 AWS SDK for Python (Boto3) 이 지원되지 Amazon Transcribe 않습니다. Python을 사용하여 스트리밍 트랜스크립션을 시작하려면 [비동기 Python SDK를 Amazon Transcribe](#) 사용합니다.

Java

다음 예시는 스트리밍 오디오를 트랜스크립션하는 Java 프로그램입니다.

이 예를 실행하려면 다음을 유의하세요.

- [Java 2.x용AWS SDK](#)를 사용해야 합니다.
- 클라이언트는 자바 1.8을 사용해야 [Java 2.x용AWS SDK](#)와 호환됩니다.
- 지정하는 샘플 속도는 오디오 스트림의 실제 샘플 속도와 일치해야 합니다.

[Amazon Transcribe 스트리밍을 위한 클라이언트 재시도\(Java SDK\)](#)도 참조하세요. 이 코드는 Amazon Transcribe 에 대한 연결을 관리하고, 연결 오류 발생 시 데이터 전송을 재시도합니다. 예를 들어 네트워크에 일시적인 오류가 발생한 경우 이 클라이언트는 실행한 요청을 다시 전송합니다.

```
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

다음 예시는 스트리밍 오디오를 트랜스크립션하는 Python 프로그램입니다.

이 예를 실행하려면 다음을 유의하세요.

- 이 [Python용 SDK](#)를 사용해야 합니다.
- 지정하는 샘플 속도는 오디오 스트림의 실제 샘플 속도와 일치해야 합니다.

```
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++

[스트리밍 C++ SDK 예시](#)는 코드 예시 장을 참조하세요.

AWS SDK에서 이 서비스 사용

AWS 소프트웨어 개발 키트(SDKs)는 널리 사용되는 많은 프로그래밍 언어에 사용할 수 있습니다. 각 SDK는 개발자가 선호하는 언어로 애플리케이션을 쉽게 구축할 수 있도록 하는 API, 코드 예시 및 설명서를 제공합니다.

SDK 설명서	코드 예제
AWS SDK for C++	AWS SDK for C++ 코드 예제
AWS CLI	AWS CLI 코드 예제
AWS SDK for Go	AWS SDK for Go 코드 예제
AWS SDK for Java	AWS SDK for Java 코드 예제

SDK 설명서	코드 예제
AWS SDK for JavaScript	AWS SDK for JavaScript 코드 예제
AWS SDK for Kotlin	AWS SDK for Kotlin 코드 예제
AWS SDK for .NET	AWS SDK for .NET 코드 예제
AWS SDK for PHP	AWS SDK for PHP 코드 예제
AWS Tools for PowerShell	AWS Tools for PowerShell 코드 예제
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) 코드 예제
AWS SDK for Ruby	AWS SDK for Ruby 코드 예제
AWS SDK for Rust	AWS SDK for Rust 코드 예제
AWS SDK for SAP ABAP	AWS SDK for SAP ABAP 코드 예제
AWS SDK for Swift	AWS SDK for Swift 코드 예제

이 서비스 관련 예시는 [AWS SDKs를 사용한 Amazon Transcribe의 코드 예제](#)를 참조하세요.

예제 사용 가능 여부

필요한 예제를 찾을 수 없습니까? 이 페이지 하단의 피드백 제공 링크를 사용하여 코드 예시를 요청하세요.

HTTP 또는 WebSocket을 통한 트랜스크립션

Amazon Transcribe 는 배치(HTTP/1.1) 및 스트리밍(HTTP/2) 트랜스크립션 모두에 대해 HTTP를 지원합니다. 스트리밍 트랜스크립션에는 WebSocket이 지원됩니다.

Amazon S3 버킷에 있는 미디어 파일을 트랜스크립션하는 경우 배치 트랜스크립션을 수행하는 것입니다. 오디오 데이터의 실시간 스트림을 트랜스크립션하는 경우 스트리밍 트랜스크립션을 수행하는 것입니다.

HTTP와 WebSocket 모두 AWS 서명 버전 4 헤더를 사용하여 요청을 인증해야 합니다. 자세한 내용은 [AWS API 요청 서명을](#) 참조하세요.

배치 트랜스크립션

다음 헤더를 사용하여 배치 HTTP 요청을 할 수 있습니다.

- host
- x-amz-target
- content-type
- x-amz-content-sha256
- x-amz-date
- authorization

다음은 StartTranscriptionJob 요청의 예입니다.

```
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/"
}
```

추가 작업 및 파라미터는 [API 참조](#)에 나열되어 있으며, 모든 AWS API 작업에 공통적인 파라미터는 [공통 파라미터](#) 섹션에 나열되어 있습니다. 다른 서명 요소는 [AWS 서명 버전 4 요청의 요소에 자세히 설명되어 있습니다](#).

스트리밍 트랜스크립션

HTTP/2 및 WebSocket을 사용한 스트리밍 트랜스크립션은 SDK를 사용하는 것보다 더 복잡합니다. 첫 스트림을 설정하기 전에 [스트리밍 트랜스크립션 설정](#) 섹션을 검토하는 것이 좋습니다.

이러한 메서드에 대한 자세한 내용은 [HTTP/2 스트림 설정](#) 또는 [WebSocket 스트림 설정](#)을 참조하세요.

Note

스트리밍 트랜스크립션에는 SDK를 사용하는 것이 좋습니다. 지원되는 SDK 목록은 [지원되는 프로그래밍 언어](#)를 참조하세요.

스트리밍 오디오 트랜스크립션

Amazon Transcribe 스트리밍을 사용하면 미디어 콘텐츠에 대한 실시간 트랜스크립션을 생성할 수 있습니다. 미디어 파일 업로드를 포함하는 배치 트랜스크립션과 달리 스트리밍 미디어는 실시간으로 Amazon Transcribe에 전달 Amazon Transcribe된 다음 트랜스크립트도 실시간으로 반환합니다.

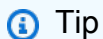
스트리밍에는 사전 녹화된 미디어(영화, 음악, 팟캐스트)와 실시간 미디어(실시간 뉴스 방송)가 포함될 수 있습니다. 의 일반적인 스트리밍 사용 사례에 Amazon Transcribe는 스포츠 이벤트에 대한 라이브 자막 및 콜센터 오디오의 실시간 모니터링이 포함됩니다.

스트리밍 콘텐츠는 Amazon Transcribe가 순식간에 변환하는 일련의 순차적 데이터 패킷 또는 '청크'로 전달됩니다. 배치보다 스트리밍을 사용하면 애플리케이션의 실시간 음성-텍스트 변환 기능과 더 빠른 트랜스크립션 시간이 장점입니다. 그러나 이러한 속도 향상으로 인해 경우에 따라 정확도가 제한될 수 있습니다.

Amazon Transcribe는 스트리밍을 위해 다음 옵션을 제공합니다.

- [SDK\(권장\)](#)
- [HTTP/2](#)
- [WebSocket](#)
- [AWS Management Console](#)

에서 스트리밍 오디오를 트랜스크립션하려면 컴퓨터 마이크에 AWS Management Console말합니다.



Tip

SDK 코드 예시는 GitHub의 [AWS 샘플 리포지토리](#)를 참조하세요.

스트리밍 트랜스크립션에 지원되는 오디오 형식은 다음과 같습니다.

- FLAC
- Ogg 컨테이너의 OPUS 인코딩 오디오
- PCM(WAV를 포함하지 않는 서명된 16비트 리틀 엔디안 오디오 형식만 해당)

무손실 형식(FLAC 또는 PCM)을 사용하는 것이 좋습니다.

Note

일부 언어에서는 스트리밍 트랜스크립션이 지원되지 않습니다. 자세한 내용은 [지원되는 언어 테이블](#)의 '데이터 입력' 열을 참조하세요.

스트리밍 트랜스크립션의 Amazon Transcribe 리전 가용성을 보려면 [Amazon Transcribe 엔드포인트 및 할당량을 참조하세요](#).

모범 사례

다음 권장 사항은 스트리밍 트랜스크립션 효율성을 개선합니다.

- 가능하면 PCM으로 인코딩된 오디오를 사용하세요.
- 스트림을 최대한 실시간에 가깝게 재생하세요.
- 지연 시간은 오디오 청크의 크기에 따라 달라집니다. 오디오 유형(예: PCM 사용)에 따라 청크 크기를 지정할 수 있는 경우 각 청크를 50ms에서 200ms 사이로 설정하세요. 다음 수식으로 오디오 청크 크기를 계산할 수 있습니다.

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

- 균일한 청크 크기를 사용하세요.
- 오디오 채널 수를 올바르게 지정해야 합니다.
- 단일 채널 PCM 오디오의 경우 각 샘플은 2바이트로 구성되므로 각 청크는 짝수 바이트로 구성되어야 합니다.
- 듀얼 채널 PCM 오디오의 경우 각 샘플은 4바이트로 구성되므로 각 청크는 4바이트의 배수여야 합니다.
- 오디오 스트림에 음성이 없는 경우 동일한 양의 무음을 인코딩하여 전송하세요. 예를 들어 PCM의 무음은 0바이트의 스트림입니다.
- 오디오에 올바른 샘플링 속도를 지정해야 합니다. 가능하면 16,000Hz의 샘플링 속도로 녹음하세요. 이렇게 하면 네트워크를 통해 전송되는 음질과 데이터 볼륨 간에 최상의 균형을 맞출 수 있습니다. 대부분의 하이엔드 마이크는 44,100Hz 또는 48,000Hz로 녹음합니다.

스트리밍 및 부분 결과

스트리밍은 실시간으로 작동하므로 트랜스크립트는 부분적인 결과로 생성됩니다.는 화자 변경 또는 오디오 일시 중지 및 같은 자연 음성 세그먼트를 기반으로 수신 오디오 스트림을 Amazon Transcribe 분할합니다. 트랜스크립션은 트랜스크립션 이벤트 스트림으로 애플리케이션에 반환되며, 전체 세그먼트가 트랜스크립션될 때까지 각 응답에는 트랜스크립션된 더 많은 음성이 포함됩니다.

이에 대한 근사치는 다음 코드 블록에 나와 있습니다. [AWS Management Console](#)에 로그인하고 실시간 트랜스크립션을 선택한 다음 마이크에 대고 말하면 이 프로세스가 실제로 진행되는 것을 볼 수 있습니다. 말하는 동안 트랜스크립션 출력 창을 보세요.

이 예시에서 각 라인은 오디오 세그먼트의 부분 결과입니다.

```
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.
```

이러한 부분 결과는 [Results](#) 객체 내 트랜스크립션 출력에 표시됩니다. 또한 이 객체 블록에는 `IsPartial` 필드가 있습니다. 이 필드가 참이면 트랜스크립션 세그먼트가 아직 완성되지 않은 것입니다. 불완전한 세그먼트와 완료된 세그먼트의 차이를 아래에서 확인할 수 있습니다.

```
"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
```

완료된 세그먼트 내의 각 단어에는 0과 1 사이의 값인 관련 신뢰도 점수가 있습니다. 값이 클수록 해당 단어를 올바르게 트랜스크립션할 가능성이 커집니다.

Tip

오디오 세그먼트의 StartTime 및 EndTime을 사용하여 트랜스크립션 출력을 비디오 대화와 동기화할 수 있습니다.

자연 시간이 짧아야 하는 애플리케이션을 실행하는 경우 [부분 결과 안정화](#)를 사용하는 것이 좋습니다.

부분 결과 안정화

Amazon Transcribe는 오디오 스트리밍을 시작하는 즉시 트랜스크립션 결과를 반환하기 시작합니다. 자연 음성 세그먼트 수준에서 완성된 결과를 생성할 때까지 이러한 부분 결과를 점진적으로 반환합니다. 자연 음성 세그먼트는 화자의 일시 중지 또는 변경이 포함된 연속 음성입니다.

Amazon Transcribe는 음성 세그먼트에 대한 최종 트랜스크립션 결과를 생성할 때까지 부분 결과를 계속 출력합니다. 음성 인식은 더 많은 컨텍스트를 얻을수록 단어를 수정할 수 있으므로 스트리밍 트랜스크립션은 부분 결과가 새로 출력될 때마다 약간씩 변경될 수 있습니다.

이 프로세스에는 음성 세그먼트별로 두 가지 옵션이 있습니다.

- 세그먼트가 완성될 때까지 대기
- 세그먼트의 부분 결과 사용

부분 결과 안정화가 각 전체 세그먼트에 대한 최종 트랜스크립션 결과를 Amazon Transcribe 생성하는 방식을 변경합니다. 이 기능을 활성화하면 부분 결과 중 마지막 몇 단어만 변경될 수 있습니다. 이로 인해 트랜스크립션 정확도가 영향을 받을 수 있습니다. 하지만 트랜스크립트는 부분 결과 안정화 기능이 없는 경우보다 더 빨리 반환됩니다. 이렇게 자연 시간을 줄이면 비디오 자막을 만들거나 실시간 스트리밍 자막을 생성할 때 유용할 수 있습니다.

다음 예는 부분 결과 안정화를 활성화하지 않은 경우와 활성화한 경우에 동일한 오디오 스트림이 처리되는 방식을 보여줍니다. 안정성 수준은 낮음, 중간 또는 높음으로 설정할 수 있습니다. 안정성이 낮으

면 정확도가 가장 높습니다. 안정성이 높으면 트랜스크립션이 더 빨라지지만 정확도는 약간 떨어집니다.

"트랜스크립트":	"EndTime":	"IsPartial":

부분 결과 안정화를 활성화하지 않음

The	0.545	true
The	1.045	true
The Amazon.	1.545	true
The Amazon is	2.045	true
The Amazon is the law.	2.545	true
The Amazon is the	3.045	true
largest	3.545	true
The Amazon is the	4.045	true
largest ray	4.545	true
The Amazon is the	5.045	true
largest rain for	5.545	true
The Amazon is the	6.025	true
largest rainforest.	6.025	false
The Amazon is the		
largest rainforest on		
the		
The Amazon is the		
largest rainforest on		
the planet.		
The Amazon is the		
largest rainforest on		
the planet.		
The Amazon is the		
largest rainforest on		
the planet.		

부분 결과 안정화 활성화(안정성 높음)

The	0.515	true
The	1.015	true
The Amazon.	1.515	true
The Amazon is	2.015	true
The Amazon is the large	2.515	true
The Amazon is the	3.015	true
largest	3.515	true

"트랜스크립트":	"EndTime":	"IsPartial":
The Amazon is the	4.015	true
largest rainfall.	4.515	true
The Amazon is the	5.015	true
largest rain forest.	5.515	true
The Amazon is the	6.015	true
largest rain forest on	6.335	true
The Amazon is the	6.335	false
largest rain forest on		
the planet.		
The Amazon is the		
largest rain forest on		
the planet.		
The Amazon is the		
largest rain forest on		
the planet.		
The Amazon is the		
largest rain forest on		
the planet.		
The Amazon is the		
largest rain forest on		
the planet.		

부분 결과 안정화를 활성화하면 Stable 필드를 Amazon Transcribe 사용하여 항목이 안정적인지 여부를 나타냅니다. 여기서 'item'은 트랜스크립션된 단어 또는 구두점 표시를 나타냅니다. Stable의 값은 true 또는 false입니다. false(안정적이지 않음)로 플래그가 지정된 항목은 세그먼트를 트랜스크립션함에 따라 변경될 가능성이 더 높습니다. 반대로 true(안정적)로 플래그가 지정된 항목은 변경되지 않습니다.

안정적이지 않은 단어를 렌더링하여 자막을 음성에 맞게 정렬할 수 있습니다. 컨텍스트가 추가되면서 자막이 약간 변경되더라도 음성에 맞을 수도 있고 그렇지 않을 수도 있는 주기적인 텍스트 연속 재생보다 사용자 경험이 개선됩니다.

또한 기울임꼴과 같이 불안정한 단어를 다른 형식으로 표시하여 시청자에게 이러한 단어가 변경될 수 있음을 알릴 수도 있습니다. 부분 결과를 표시하면 해당 시간에 표시되는 텍스트의 양이 제한됩니다. 이는 비디오 자막과 같이 공간 제약이 있는 경우에 중요할 수 있습니다.

AWS Machine Learning 블로그로 더 자세히 알아보기

실시간 트랜스크립션으로 정확도를 높이는 방법에 대해 자세히 알아보려면 다음을 참조하세요.

- [Amazon Transcribe 부분 결과 안정화를 통해 스트리밍 트랜스크립션 경험 개선](#)
- [“그게 뭐였죠?” Amazon Transcribe를 사용한 라이브 브로드캐스트의 자막 정확도 향상](#)

부분 결과 안정화 예시 출력

다음 예시 출력은 불완전한 세그먼트("IsPartial": true)에 대한 Stable 플래그를 보여줍니다. "to" 및 "Amazon"이라는 단어는 안정적이지 않으므로 세그먼트가 확정되기 전에 변경될 수 있음을 알 수 있습니다.

```
"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
}
]
}

```

스트리밍 트랜스크립션 설정

이 섹션은 메인 [스트리밍](#) 섹션의 확장판으로, AWS SDK가 아닌 HTTP/2 또는 WebSockets으로 스트림을 직접 설정하려는 사용자에게 정보를 제공하기 위한 것입니다. 이 섹션의 정보를 사용하여 자체 SDK를 빌드할 수도 있습니다.

Important

HTTP/2 및 WebSocket을 직접 사용하는 것보다 SDK를 사용하는 것을 적극 권장합니다. SDK는 데이터 스트림을 트랜스크립션하는 가장 간단하고 신뢰할 수 있는 방법입니다. AWS SDK를 사용하여 스트리밍을 시작하려면 [섹션을 참조하세요](#) [AWS SDKs](#).

HTTP/2 스트림 설정

를 사용하여 트랜스크립션 요청을 스트리밍하기 위한 [HTTP/2 프로토콜](#)의 주요 구성 요소는 다음과 Amazon Transcribe 같습니다.

- 헤더 프레임. 여기에는 요청에 대한 HTTP/2 헤더와를 시드 서명으로 Amazon Transcribe 사용하여 데이터 프레임에 서명하는 권한 부여 헤더의 서명이 포함됩니다.
- 메타데이터와 원시 오디오 바이트가 포함된 [이벤트 스트림 인코딩](#) 내 하나 이상의 메시지 프레임.
- 종료 프레임. [이벤트 스트림 인코딩](#)의 본문이 비어 있는 서명된 메시지입니다.

Note

Amazon Transcribe 는 HTTP/2 세션당 하나의 스트림만 지원합니다. 스트림을 여러 개 사용하려고 하면 트랜스크립션 요청이 실패합니다.

1. 요청을 수행하는 IAM 역할에 다음 정책을 연결합니다. 자세한 내용은 [IAM 정책 추가](#)를 참조하세요.

JSON

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

2. 세션을 시작하려면 Amazon Transcribe로 HTTP/2 요청을 전송합니다.

```
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
```

추가 작업 및 파라미터는 [API 참조](#)에 나열되어 있으며, 모든 AWS API 작업에 공통적인 파라미터는 [공통 파라미터](#) 섹션에 나열되어 있습니다.

Amazon Transcribe 는 다음 응답을 전송합니다.

```
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
```

- 오디오 데이터가 포함된 오디오 이벤트를 만듭니다. 다음 테이블에 설명된 헤더를 오디오 바이트 청크와 결합하여 이벤트 코딩된 메시지를 만듭니다. 이벤트 메시지에 대한 페이로드를 만들려면 원시 바이트 형식의 버퍼를 사용합니다.

헤더 이름 바이트 길이	헤더 이름(문자열)	헤더 값 유형	값 문자열 바이트 길이	값 문자열 (UTF-8)
13	:content-type	7	24	application/octet-stream
11	:event-type	7	10	AudioEvent
13	:message-type	7	5	event

이 예시의 이진 데이터는 base64로 인코딩됩니다. 실제 요청에서 데이터는 원시 바이트입니다.

```
:content-type: "application/vnd.amazon.eventstream"
:event-type: "AudioEvent"
:message-type: "event"
Uk1GRjZxPQBxQVZFZm10IBAAAAABAAEAgD4AAAB9AAACABAAZGF0YVVTwPQAAAAAAAAAAAAAAAAAAD//wIA/f8EAA==
```

4. 오디오 데이터가 포함된 오디오 메시지를 만듭니다.

- a. 오디오 메시지 데이터 프레임에는 오디오 청크 및 오디오 이벤트에 대한 현재 날짜와 서명이 포함된 이벤트 인코딩 헤더가 포함됩니다.

헤더 이름 바이트 길이	헤더 이름(문자열)	헤더 값 유형	값 문자열 바이트 길이	값
16	:청크 서명	6	varies	생성된 서명
5	:날짜	8	8	timestamp

이 요청의 이진 데이터는 base64로 인코딩됩니다. 실제 요청에서 데이터는 원시 바이트입니다.

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

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

```
AAAA0gAAAIKVoRFcTTcjb250ZW50LXR5cGUHABhhcHBsaWNhdGlvbi9vY3RldC1zdHJlYW0L0mV2ZW50LXR5cGUHAAPBdWRpb0V2ZW50DTptZXNzYWdlLXR5cGUHAAPV1dmVudAxDb256ZW50LVR5cGUHABphcHBsaWNhdGlvbi94LWFtei1qc29uLTEuMVJJRkY88T0AV0FWRWZtdCAQAAAAAQAABAIA
+AAAAfQAAAgAQAGRhdGFU8D0AAAAA
AAAAAAAAAAAA//8CAP3/BAC7QLFf
```

- b. [서명 버전 4의 서명할 문자열 만들기](#)에 설명된 대로 서명할 문자열을 구성합니다. 문자열은 다음 형식을 따릅니다.

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

- **DateTime**: 서명이 생성된 날짜 및 시간. 형식은 YYYYMMDDTHHMMSSZ입니다. 여기서 YYYY=년, MM=월, DD=일, HH=시간, MM=분, SS=초이고, 'T'와 'Z'는 고정 문자입니다. 자세한 내용은 [서명 버전 4의 날짜 처리](#)를 참조하세요.
 - **Keypath**: 서명은 date/region/service/aws4_request 형식으로 표시됩니다. 예를 들어 20220127/us-west-2/transcribe/aws4_request입니다.
 - **Hex**: 입력을 16진수 형식으로 인코딩하는 함수.
 - **priorSignature**: 이전 프레임의 서명. 첫 데이터 프레임에는 헤더 프레임의 서명을 사용합니다.
 - **HexHash**: 먼저 입력의 SHA-256 해시를 생성한 후 함수를 사용하여 해시를 인코딩하는 함수.
 - **nonSignatureHeaders**: 문자열로 인코딩된 DateTime 헤더.
 - **payload**: 오디오 이벤트 데이터가 포함된 바이트 버퍼.
- c. AWS 보안 액세스 키에서 서명 키를 생성하고 이를 사용하여 서명합니다 `stringToSign`. 생성된 키는 더 높은 수준의 보호를 위해 날짜, 서비스 및 AWS 리전에 고유합니다. 자세한 내용은 [AWS서명 버전 4의 서명 계산](#)을 참조하세요.

서명 키를 추출하는 `GetSignatureKey` 함수를 구현했는지 확인합니다. 서명 키를 아직 추출하지 않은 경우 [서명 버전 4에 대한 서명 키 생성 방법을 보여주는 예](#)를 참조하세요.

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

- **HMACSHA256**: SHA-256 해시 함수를 사용하여 서명을 만드는 함수.
- **derivedSigningKey**: 서명 버전 4 서명 키.
- **stringToSign**: 데이터 프레임에 대해 계산된 문자열.

데이터 프레임의 서명을 계산한 후 날짜, 서명 및 오디오 이벤트 페이로드가 포함된 바이트 버퍼를 구성합니다. 바이트 배열을 트랜스크립션을 수행할 Amazon Transcribe 로 보냅니다.

5. 오디오 스트림의 완료를 알리기 위해 날짜와 서명만 포함된 빈 데이터 프레임을 전송합니다. 이 종료 프레임은 데이터 프레임을 구성한 것과 동일한 방식으로 구성합니다.

Amazon Transcribe 는 애플리케이션으로 전송되는 트랜스크립션 이벤트 스트림으로 응답합니다. 이 응답은 이벤트 스트림 인코딩됩니다. 여기에는 표준 서문과 다음 헤더가 포함됩니다.

헤더 이름 바이트 길이	헤더 이름(문자열)	헤더 값 유형	값 문자열 바이트 길이	값 문자열 (UTF-8)
13	:content-type	7	16	application/json
11	:event-type	7	15	TranscriptEvent
13	:message-type	7	5	event

이벤트는 원시 바이트 형식으로 전송됩니다. 이 예에서 바이트는 base64로 인코딩됩니다.

```
AAAAUwAAAEp1RHpYBTpkYXR1CAAAAWiXUkMLEDpjaHVuay1zaWduYXR1cmUGACct6Zy+uymwEK2SrLp/
zVBI
5eGn83jdBwCaRUBJA+eaDafqjqI=
```

트랜스크립션 결과를 보려면 이벤트 스트림 인코딩을 사용하여 원시 바이트를 디코딩합니다.

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

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

6. 스트리밍을 종료하려면 Amazon Transcribe로 빈 오디오 이벤트를 전송합니다. 빈 페이로드를 제외하고 오디오 이벤트는 다른 오디오 이벤트와 동일하게 만듭니다. 이벤트를 서명하고 서명을 다음과 같이 :chunk-signature 헤더에 포함합니다.

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

HTTP/2 스트리밍 오류 처리

미디어 스트림을 처리할 때 오류가 발생하면 예외 응답을 Amazon Transcribe 보냅니다. 응답은 이벤트 스트림 인코딩됩니다.

응답에는 표준 서문과 다음 헤더가 포함됩니다.

헤더 이름 바이트 길이	헤더 이름(문자열)	헤더 값 유형	값 문자열 바이트 길이	값 문자열(UTF-8)
13	:content-type	7	16	application/json
11	:event-type	7	19	BadRequestException
13	:message-type	7	9	exception

예외 응답이 디코딩되면 다음 정보가 포함됩니다.

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

Exception message

WebSocket 스트림 설정

를 사용하여 트랜스크립션 요청을 스트리밍하기 위한 [WebSocket 프로토콜](#)의 주요 구성 요소는 다음과 Amazon Transcribe 같습니다.

- 업그레이드 요청. 여기에는 요청에 대한 쿼리 파라미터와를 시드 서명으로 Amazon Transcribe 사용하여 데이터 프레임에 서명하는 서명이 포함됩니다.
- 메타데이터와 원시 오디오 바이트가 포함된 [이벤트 스트림 인코딩](#) 내 하나 이상의 메시지 프레임.
- 종료 프레임. [이벤트 스트림 인코딩](#)의 본문이 비어 있는 서명된 메시지입니다.

Note

Amazon Transcribe 는 WebSocket 세션당 하나의 스트림만 지원합니다. 스트림을 여러 개 사용하려고 하면 트랜스크립션 요청이 실패합니다.

1. 요청을 수행하는 IAM 역할에 다음 정책을 연결합니다. 자세한 내용은 [IAM 정책 추가](#)를 참조하세요.

JSON

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

2. 세션을 시작하려면 다음 형식으로 미리 서명된 URL을 만듭니다. 가독성을 높이기 위해 줄바꿈이 추가되었습니다.

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


Note

X-Amz-Expires의 최대값은 300(5분)입니다.

추가 작업 및 파라미터는 [API 참조](#)에 나열되어 있으며, 모든 AWS API 작업에 공통적인 파라미터는 [공통 파라미터](#) 섹션에 나열되어 있습니다.

요청에 대한 URL과 [서명 버전 4 서명](#)을 생성하려면 다음 단계를 참조하세요. 유사 코드의 예시입니다.

- a. 표준 요청을 생성합니다. 표준 요청은 표준화된 형식으로 된 요청의 정보가 포함된 문자열입니다. 이렇게 하면가 요청을 AWS 수신할 때 URL에 대해 생성한 것과 동일한 서명을 계산할 수 있습니다. 자세한 내용은 [서명 버전 4에 대한 표준 요청 생성](#)을 참조하세요.

```
# 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
timestamp = YYYYMMDD
```

- b. 도메인과 쿼리 문자열 간의 URI 부분인 표준 URI를 생성합니다.

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

- c. 표준 헤더 및 서명된 헤더를 생성합니다. 표준 헤더의 후행 \n에 유의하세요.
 - 소문자 헤더 이름과 콜론(:)을 차례대로 추가합니다.
 - 헤더에 대한 쉼표로 구분된 값 목록을 추가합니다. 여러 값을 가진 헤더에서 값을 정렬하지 마세요.
 - 새 줄(\n)을 추가합니다.

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

- d. 이 알고리즘을 해싱 알고리즘과 일치시킵니다. SHA-256를 사용합니다.

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

- e. 생성된 키의 범위를 날짜, AWS 리전 및 서비스로 지정하는 자격 증명 범위를 생성합니다. 예를 들어 `20220127/us-west-2/transcribe/aws4_request`입니다.

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

- f. 표준 쿼리 문자열을 생성합니다. 쿼리 문자열 값은 URL로 인코딩되어야 하며 이름을 기준으로 정렬되어야 합니다.
- 문자 코드 포인트를 기준으로 파라미터 이름을 오름차순으로 정렬합니다. 중복된 이름을 가진 파라미터는 값별로 정렬해야 합니다. 예를 들어 대문자 F로 시작하는 파라미터 이름 앞에 소문자 b로 시작하는 파라미터 이름이 옵니다.
 - 예약되지 않은 문자는 URI로 인코딩하지 않습니다. RFC 3986에 정의된 예약되지 않은 문자는 A-Z, a-z, 0-9, 하이픈(-), 밑줄(_), 마침표(.) 및 물결표(~)입니다.
 - %XY와 같이 모든 기타 문자를 퍼센트 인코딩합니다. 여기서 X 및 Y는 16진 문자(0~9 및 대문자 A~F)입니다. 예를 들어 공백 문자는 %20(일부 인코딩 구조처럼 +를 넣지 말 것)로 인코딩되고, 확장 UTF-8 문자는 %XY%ZA%BC 형식이어야 합니다.
 - 매개변수 값에서 등호(=) 문자를 두 번 인코딩합니다.

```
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. 페이로드의 해시를 생성합니다. GET 요청의 경우 페이로드는 빈 문자열입니다.

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

- h. 다음 요소를 결합하여 표준 요청을 생성합니다.

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

3. 서명할 문자열을 생성합니다. 여기에는 요청에 대한 메타 정보가 포함됩니다. 요청 서명을 계산할 때 다음 단계의 서명할 문자열을 사용합니다. 자세한 내용은 [서명 버전 4에 대한 서명할 문자열 생성](#)을 참조하세요.

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

4. 서명을 계산합니다. 이렇게 하려면 AWS 보안 액세스 키에서 서명 키를 파생합니다. 생성된 키는 더 높은 수준의 보호를 위해 낱파, 서비스 및 AWS 리전에 고유합니다. 생성된 키를 사용하여 요청에 서명합니다. 자세한 내용은 [서명 버전 4의 AWS 서명 계산을 참조하세요](#).

서명 키를 추출하는 GetSignatureKey 함수를 구현했는지 확인합니다. 서명 키를 아직 추출하지 않은 경우 [서명 버전 4에 대한 서명 키 생성 방법을 보여주는 예](#)를 참조하세요.

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

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

HMAC(key, data) 함수는 결과를 이진 형식으로 반환하는 HMAC-SHA256 함수를 나타냅니다.

5. 요청에 서명 정보를 추가하고 요청 URL을 생성합니다.

서명을 계산한 후 쿼리 문자열에 추가합니다. 자세한 내용은 [요청에 서명 추가](#)를 참조하세요.

첫째, 쿼리 문자열에 인증 정보를 추가합니다.

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

둘째, 요청에 대한 URL을 생성합니다.

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

요청 URL을 WebSocket 라이브러리와 함께 사용하여 Amazon Transcribe에 대한 요청을 수행합니다.

6. 에 대한 요청에는 다음 헤더가 포함되어야 Amazon Transcribe 합니다. 일반적으로 이러한 헤더는 WebSocket 클라이언트 라이브러리에서 관리합니다.

```
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. 가 WebSocket 요청을 Amazon Transcribe 수신하면 WebSocket 업그레이드 응답으로 응답합니다. 일반적으로 WebSocket 라이브러리는이 응답을 관리하고 통신을 위한 소켓을 설정합니다 Amazon Transcribe.

다음은의 응답입니다 Amazon Transcribe. 가독성을 높이기 위해 줄바꿈이 추가되었습니다.

```
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. WebSocket 스트리밍을 요청합니다.

WebSocket 연결이 설정되면 클라이언트는 각각 [이벤트 스트림 인코딩](#)을 사용하여 인코딩한 오디오 프레임 시퀀스 전송을 시작할 수 있습니다.

각 데이터 프레임에는 원시 오디오 바이트의 청크와 결합된 세 개의 헤더가 있으며, 다음 테이블은 이러한 헤더를 설명한 것입니다.

헤더 이름 바이트 길이	헤더 이름(문자열)	헤더 값 유형	값 문자열 바이트 길이	값 문자열 (UTF-8)
13	:content-type	7	24	application/octet-stream
11	:event-type	7	10	AudioEvent
13	:message-type	7	5	event

9. 데이터 스트림을 종료하려면 이벤트 스트림 인코딩 메시지에 빈 오디오 청크를 보냅니다.

응답에는 페이로드의 이벤트 스트림 인코딩 원시 바이트가 포함되어 있습니다. 여기에는 표준 서문과 다음 헤더가 포함됩니다.

헤더 이름 바이트 길이	헤더 이름(문자열)	헤더 값 유형	값 문자열 바이트 길이	값 문자열 (UTF-8)
13	:content-type	7	16	application/json
11	:event-type	7	15	TranscriptEvent
13	:message-type	7	5	event

이진 응답을 디코딩할 때 트랜스크립션 결과가 포함된 JSON 구조로 종료합니다.

WebSocket 스트리밍 오류 처리

요청을 처리하는 동안 예외가 발생하면 이벤트 스트림 인코딩 Amazon Transcribe 응답이 포함된 터미널 WebSocket 프레임으로 응답합니다. 이 응답에는 다음 테이블에 설명된 헤더가 포함되며, 응답 본문에는 서술적인 오류 메시지가 포함되어 있습니다. 예외 응답을 전송한 후는 닫힌 프레임을 Amazon Transcribe 전송합니다.

헤더 이름 바이트 길이	헤더 이름(문자열)	헤더 값 유형	값 문자열 바이트 길이	값 문자열(UTF-8)
13	:content-type	7	16	application/json
15	:exception-type	7	varies	다양함, 아래 참조
13	:message-type	7	9	exception

exception-type 헤더에는 다음 값 중 하나가 포함됩니다.

- **BadRequestException**: 스트림을 생성할 때 클라이언트 오류가 발생했거나 데이터를 스트리밍하는 동안 오류가 발생했습니다. 클라이언트가 데이터를 수락할 준비가 되어 있는지 확인하고 요청을 다시 시도하세요.
- **InternalFailureException**: 클라이언트와의 핸드셰이크 중에 문제가 Amazon Transcribe 발생했습니다. 요청을 다시 시도하세요.
- **LimitExceededException**: 클라이언트가 동시 스트림 제한을 초과했습니다. 자세한 내용은 [Amazon Transcribe 제한](#)을 참조하세요. 트랜스크립션하는 스트림의 수를 줄입니다.
- **UnrecognizedClientException**: WebSocket 업그레이드 요청이 잘못된 액세스 키 또는 보안 암호 키로 서명되었습니다. 액세스 키를 올바르게 생성하고 있는지 확인하고 요청을 다시 시도하세요.

Amazon Transcribe 는 일반적인 서비스 오류를 반환할 수도 있습니다. 목록은 [일반 오류](#)를 참조하세요.

이벤트 스트림 인코딩

Amazon Transcribe 는 스트리밍 트랜스크립션에 이벤트 스트림 인코딩이라는 형식을 사용합니다.

이벤트 스트림 인코딩은 클라이언트와 서버 간 양방향 통신 기능을 제공합니다. Amazon Transcribe 스트리밍 서비스로 전송된 데이터 프레임은 이 형식으로 인코딩됩니다. 의 응답도 이 인코딩을 Amazon Transcribe 사용합니다.

각 메시지는 두 섹션(서문 및 데이터)으로 구성됩니다. 서문의 구성 요소는 다음과 같습니다.

1. 메시지의 총 바이트 길이
2. 모든 헤더의 바이트 길이 합계

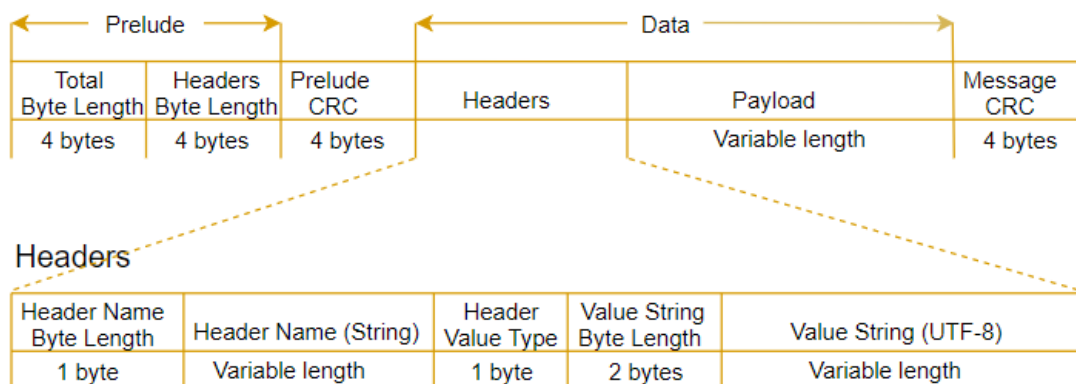
데이터 섹션의 구성 요소는 다음과 같습니다.

1. 헤더
2. Payload

각 섹션은 4바이트 빅 엔디안 정수 CRC(순환 중복 검사) 체크섬으로 끝납니다. 메시지 CRC 체크섬은 서문 섹션과 데이터 섹션 모두에 대한 것입니다. Amazon Transcribe 는 CRC32(GZIP CRC32 라고도 함)를 사용하여 두 CRC를 모두 계산합니다. CRC32에 대한 자세한 내용은 [GZIP file format specification version 4.3](#)을 참조하세요.

총 메시지 오버헤드(서문과 두 체크섬 포함)는 16바이트입니다.

다음 다이어그램은 메시지와 헤더를 구성하는 구성 요소를 보여줍니다. 메시지별로 여러 개의 헤더가 있습니다.



각 메시지는 다음 구성 요소를 포함합니다.

- 서문: 두 개의 4바이트 필드로 구성되어 있으며 총 8바이트는 고정됩니다.
 - 첫 번째 4바이트: 전체 메시지의 빅 엔디안 정수 바이트 길이이며 4바이트 길이의 이 필드는 포함됩니다.

- 두 번째 4바이트: 메시지 헤더 부분의 빅 엔디안 정수 바이트 길이이며 헤더의 길이 필드 자체는 제외됩니다.
- 서문 CRC: 메시지 서문 부분의 4바이트 CRC 체크섬이며 CRC 자체는 제외됩니다. 서문에는 메시지 CRC와 별도의 CRC가 있습니다. 이렇게 하면 버퍼 오버런과 같은 오류를 일으키지 않고 손상된 바이트 길이 정보를 즉시 감지 Amazon Transcribe 할 수 있습니다.
- 헤더: 메시지 유형, 콘텐츠 유형 등의 메시지 주석 메타데이터. 메시지는 키:값 페어인 헤더가 여러 개 있으며, 여기서 키는 UTF-8 문자열입니다. 헤더는 메시지의 헤더 부분에 순서에 상관없이 표시되며 각 헤더는 한 번만 표시될 수 있습니다.
- 페이로드: 트랜스크립션할 오디오 콘텐츠.
- 메시지 CRC: 메시지 시작 부분부터 체크섬 시작 부분까지의 4바이트 CRC 체크섬. 즉 CRC 자체를 제외한 메시지의 전체입니다.

헤더 프레임은 스트리밍 트랜스크립션에 대한 권한 부여 프레임입니다.는 권한 부여 헤더의 값을 요청의 데이터 프레임에 대한 권한 부여 헤더 체인을 생성하기 위한 시드로 Amazon Transcribe 사용합니다.

각 헤더에는 다음과 같은 구성 요소가 포함되며, 프레임당 여러 개의 헤더가 있습니다.

- 헤더 이름 바이트 길이: 헤더 이름의 바이트 길이.
- 헤더 이름: 헤더 유형을 나타내는 헤더 이름. 유효한 값은 다음 프레임 설명을 참조하세요.
- 헤더 값 유형: 헤더 값 유형을 나타내는 숫자. 다음 목록은 헤더에 사용할 수 있는 값과 해당 값의 의미를 보여줍니다.
 - 0 – TRUE(참)
 - 1 – FALSE(거짓)
 - 2 – BYTE(바이트)
 - 3 – SHORT(짧음)
 - 4 – INTEGER(정수)
 - 5 – LONG(길)
 - 6 – BYTE ARRAY(바이트 배열)
 - 7 – STRING(문자열)
 - 8 – TIMESTAMP(타임스탬프)
 - 9 – UUID
- 값 문자열 바이트 길이: 헤더 값 문자열의 바이트 길이.

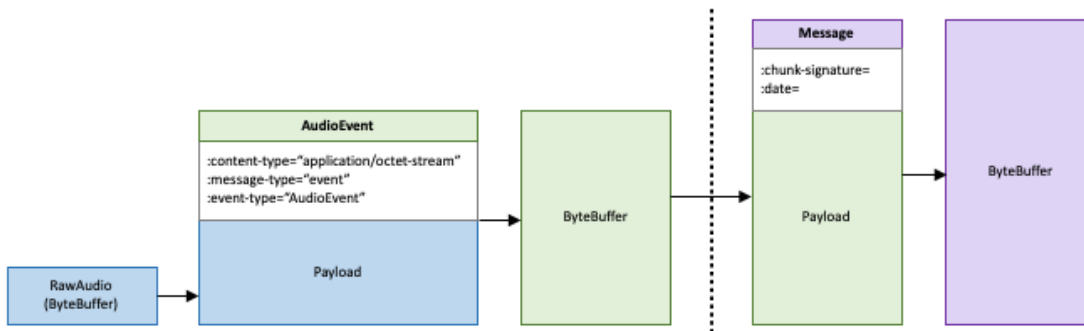
- **헤더 값:** 헤더 문자열의 값. 이 필드의 유효한 값은 헤더 유형에 따라 달라집니다. 자세한 내용은 [HTTP/2 스트림 설정](#) 또는 [WebSocket 스트림 설정](#) 섹션을 참조하세요.

데이터 프레임

각 스트리밍 요청에는 하나 이상의 데이터 프레임이 포함됩니다. 데이터 프레임을 생성하기 위해서는 두 가지 단계가 필요합니다.

1. 원시 오디오 데이터를 메타데이터와 결합하여 요청의 페이로드를 생성합니다.
2. 페이로드를 서명과 결합하여 Amazon Transcribe로 전송되는 이벤트 메시지를 만듭니다.

다음 다이어그램은 이 과정을 보여 줍니다.



작업 대기열

작업 대기열을 사용하면 동시에 처리할 수 있는 것보다 더 많은 트랜스크립션 작업 요청을 제출할 수 있습니다. 작업 대기열을 사용하지 않고 허용된 동시 요청 할당량에 도달하면 하나 이상의 요청이 완료 될 때까지 기다렸다가 새 요청을 제출해야 합니다.

작업 대기열은 트랜스크립션 작업 및 통화 후 분석 작업 요청 모두에 대해 선택 사항입니다.

작업 대기열을 활성화하면 Amazon Transcribe는 한도를 초과하는 모든 요청을 포함하는 대기열을 생성합니다. 요청이 완료되는 즉시 대기열에서 새 요청을 가져와 처리합니다. 대기 상태인 요청은 FIFO(선입선출)에 따라 처리됩니다.

대기열에 최대 10,000개의 작업을 추가할 수 있습니다. 이 한도를 초과할 경우 `LimitExceededConcurrentJobException` 오류가 발생합니다. 최적의 성능을 유지하기 위해 Amazon Transcribe는 대기 상태인 작업을 처리하는 데 할당량의 최대 90%(대역폭 비율 0.9)까지만 사용합니다. 요청 시 높일 수 있는 기본값이라는 점에 유의하세요.

Tip

[AWS 일반 참조](#)에서 Amazon Transcribe 리소스의 기본 한도 및 할당량 목록을 찾을 수 있습니다. 요청 시 이러한 기본값 중 일부를 높일 수 있습니다.

작업 대기열을 활성화했지만 동시 요청 할당량을 초과하지 않은 경우 모든 요청이 동시에 처리됩니다.

작업 대기열 활성화

AWS Management Console, AWS CLI 또는 AWS SDK를 사용하여 작업 대기열을 활성화할 수 있습니다. 예를 보려면 다음을 참조하세요. 예를 보려면 다음을 참조하세요.

AWS Management Console

1. 에 로그인합니다..[AWS Management Console](#)
2. 탐색 창에서 트랜스크립션 작업을 선택한 다음 작업 생성(오른쪽 상단)을 선택합니다. 그러면 작업 세부 정보 지정 페이지가 열립니다.
3. 작업 설정 상자에는 추가 설정 패널이 있습니다. 이 패널을 확장하면 작업 대기열에 추가 상자를 선택하여 작업 대기열을 활성화할 수 있습니다.

Specify job details Info

Job settings

Name

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

Language settings

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

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

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

Language
Choose the language of the input audio.

English, US (en-US) ▼

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

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

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

▼ **Additional settings**

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

☐ Add to job queue

- 작업 세부 정보 지정 페이지에 포함하려는 다른 필드를 모두 채운 후 다음을 선택합니다. 그러면 작업 구성 - 선택 사항 페이지로 이동합니다.
- 작업 생성을 선택하여 트랜스크립션 작업을 실행합니다.

AWS CLI

이 예시에서는 [start-transcription-job](#) 명령 및 job-execution-settings 파라미터를 AllowDeferredExecution 하위 파라미터와 함께 사용합니다. 단, 요청에 AllowDeferredExecution을 포함시킬 때는 DataAccessRoleArn 역시 포함해야 합니다.

자세한 내용은 [StartTranscriptionJob](#) 및 [JobExecutionSettings](#)을(를) 참조하세요.

```
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
```

다음은 [start-transcription-job](#) 명령을 사용하는 또 다른 예 및 대기열을 활성화하는 요청 본문입니다.

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

my-first-queueing-request.json 파일에는 다음과 같은 요청 본문이 포함되어 있습니다.

```
{
  "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 for Python (Boto3)

이 예시에서는 AWS SDK for Python (Boto3)을 사용하여 [start_transcription_job](#) 메서드의 AllowDeferredExecution 인수를 사용하여 작업 대기열을 활성화합니다. 단, 요청에 AllowDeferredExecution을 포함시킬 때는 DataAccessRoleArn 역시 포함해야 합니다. 자세한 내용은 [StartTranscriptionJob](#) 및 [JobExecutionSettings](#)을(를) 참조하세요.

기능별, 시나리오 및 교차 서비스 예를 포함하여 AWS SDK를 사용하는 추가 예시는 [AWS SDKs를 사용한 Amazon Transcribe의 코드 예제](#) 장을 참조하세요.

```

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)

```

AWS Management Console을 통해 또는 [GetTranscriptionJob](#) 요청을 제출하여 대기 상태인 작업의 진행 상황을 볼 수 있습니다. 작업이 대기열에 있으면 Status는 QUEUED입니다. 작업에서 처리가 시작되면 상태가 IN_PROGRESS로 변경되고 처리가 완료되면 COMPLETED 또는 FAILED로 변경됩니다.

리소스에 태그 지정

태그는 리소스를 좀 더 쉽게 식별하고 정리하고 검색하기 위해 리소스에 추가할 수 있는 사용자 지정 메타데이터 레이블입니다. 태그는 태그 키와 태그 값이라는 두 개의 개별 부분으로 구성됩니다. 이를 키:값 페어라고 합니다.

태그 키는 일반적으로 더 큰 범주를 나타내고 태그 값은 해당 범주의 하위 집합을 나타냅니다. 예를 들어 태그 키=색상 및 태그 값=파란색을 지정하면 키:값 페어 Color:Blue가 생성됩니다. 태그의 값을 빈 문자열로 설정할 수 있지만 태그의 값을 Null로 설정할 수는 없습니다. 태그 값을 생략하는 것은 빈 문자열을 사용하는 것과 같습니다.

Tip

AWS 결제 및 비용 관리는 태그를 사용하여 청구서를 동적 범주로 구분할 수 있습니다. 예를 들어 회사 내 여러 부서를 나타내는 태그를 추가하는 경우(예: Department:Sales 또는 Department:Legal) AWS는 부서별 비용 분배를 제공할 수 있습니다.

Amazon Transcribe에서 다음 리소스에 태그를 지정할 수 있습니다.

- 트랜스크립션 작업
- 의료 트랜스크립션 작업
- 통화 분석 통화 후 트랜스크립션 작업
- 사용자 지정 어휘
- 사용자 지정 의학 어휘
- 사용자 지정 어휘 필터
- 통화 분석 범주
- 사용자 지정 언어 모델

태그 키는 최대 길이가 128자이고 태그 값은 최대 길이가 256자일 수 있으며 둘 다 대소문자를 구분합니다. Amazon Transcribe는 리소스당 최대 50개의 태그를 지원합니다. 지정된 리소스에 대해 각 태그 키는 하나의 값만 가질 수 있습니다. AWS는 시스템 생성 태그에 대해 이 접두사를 예약하기 때문에 태그는 aws:로 시작할 수 없습니다. aws:* 태그는 추가, 수정 또는 삭제할 수 없으며 리소스당 태그 수 제한에 포함되지 않습니다.

리소스 태깅과 관련된 API 작업

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

태그 지정 API를 사용하려면 Amazon 리소스 이름(ARN)을 요청에 포함해야 합니다. ARN의 형식은 `arn:partition:service:region:account-id:resource-type/resource-id`입니다. 예를 들어, 트랜스크립션 작업과 관련된 ARN은 `arn:aws:transcribe:us-west-2:111122223333:transcription-job/my-transcription-job-name`과 비슷한 모습일 수 있습니다.

모범 사례를 포함한 태깅에 대한 자세한 내용은 [AWS 리소스 태깅](#)을 참조하세요.

태그 기반 액세스 제어

태그를 사용하여 AWS 계정 내의 액세스를 제어할 수 있습니다. 태그를 기반으로 액세스를 제어하려면 IAM 정책의 조건 요소에 태그 정보를 제공합니다. 그런 다음 태그 및 연관된 태그 조건 키를 사용하여 다음에 대한 액세스를 제어할 수 있습니다.

- 리소스: 리소스에 할당한 태그를 기반으로 Amazon Transcribe 리소스에 대한 액세스를 제어합니다.
 - 이를 위해 `aws:ResourceTag/key-name` 조건 키를 사용하여 리소스에 연결해야 하는 태그 키: 값 페어를 지정합니다.
- 요청: 어떤 태그가 요청에 전달될 수 있는지 제어합니다.
 - `aws:RequestTag/key-name` 조건 키를 사용하여 어떤 태그를 IAM 사용자 또는 역할에서 추가, 수정 또는 제거할 수 있는지 지정합니다.
- 권한 부여 프로세스: 권한 부여 프로세스의 모든 부분에 대한 태그 기반 액세스를 제어합니다.
 - `aws:TagKeys/` 조건 키를 사용하여 리소스나 요청에서 또는 주체가 특정 태그 키를 사용할 수 있는지 여부를 제어합니다. 이 경우 키 값은 중요하지 않습니다.

태그 기반 액세스 제어 정책 예시는 [태그를 기준으로 트랜스크립션 작업 보기](#) 섹션을 참조하세요.

태그 기반 액세스 제어에 대한 자세한 내용은 [태그를 사용한 AWS 리소스 액세스 제어](#)를 참조하세요.

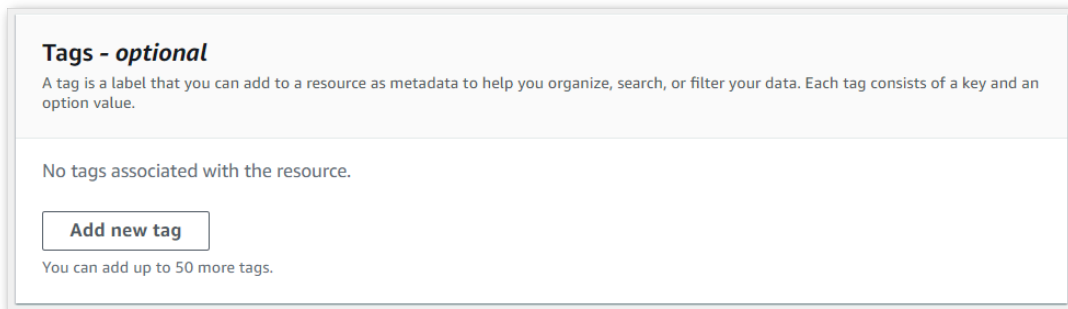
Amazon Transcribe 리소스에 태그 추가

Amazon Transcribe 작업을 실행하기 전이나 후에 태그를 추가할 수 있습니다. 기존 Create* 및 Start* API를 사용하여 트랜스크립션 요청에 태그를 추가할 수 있습니다.

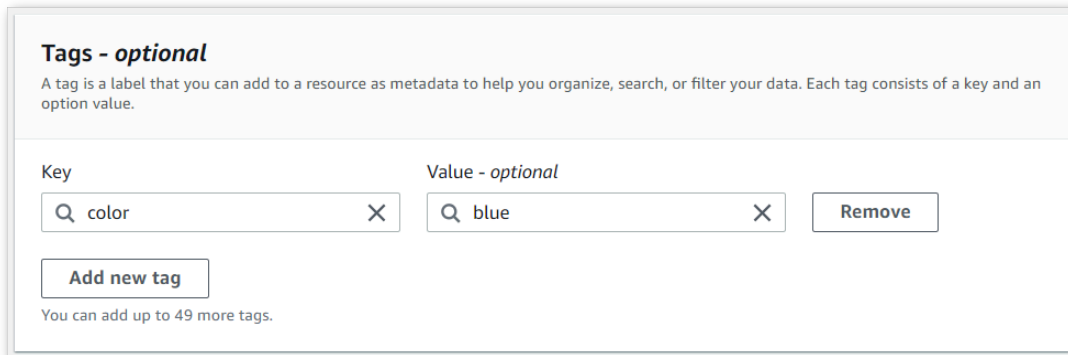
AWS Management Console, AWS CLI 또는 AWS SDK를 사용하여 태그를 추가, 수정 또는 삭제할 수 있습니다, 다음의 예시를 참조하세요.

AWS Management Console

1. 에 로그인합니다..[AWS Management Console](#)
2. 탐색 창에서 트랜스크립션 작업을 선택한 다음 작업 생성(오른쪽 상단)을 선택합니다. 그러면 작업 세부 정보 지정 페이지가 열립니다.
3. 작업 세부 정보 지정 페이지 하단으로 스크롤하여 태그 - 선택 사항 상자를 찾은 다음 새 태그 추가를 선택합니다.



4. 키 필드 및 값 필드(선택 사항)에 대한 정보를 입력합니다.



5. 작업 세부 정보 지정 페이지에 포함하려는 다른 필드를 모두 채운 후 다음을 선택합니다. 그러면 작업 구성 - 선택 사항 페이지로 이동합니다.

작업 생성을 선택하여 트랜스크립션 작업을 실행합니다.

6. 트랜스크립션 작업 페이지로 이동하여 트랜스크립션 작업을 선택한 다음 해당 작업의 정보 페이지 하단으로 스크롤하면 트랜스크립션 작업과 관련된 태그를 볼 수 있습니다. 태그를 편집하려는 경우 태그 관리를 선택하면 됩니다.

Tags (2)		Manage Tags
Key	Value	
color	blue	

AWS CLI

이 예시에서는 [start-transcription-job](#) 명령 및 Tags 파라미터를 사용합니다. 자세한 내용은 [StartTranscriptionJob](#) 및 [Tag](#)을(를) 참조하세요.

```
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
```

다음은 [start-transcription-job](#) 명령을 사용하는 또 다른 예 및 해당 작업에 태그를 추가하는 요청 본문입니다.

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

my-first-tagging-job.json 파일에는 다음과 같은 요청 본문이 포함되어 있습니다.

```
{
  "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 for Python (Boto3)

다음 예시에서는 AWS SDK for Python (Boto3)를 통해 [start_transcription_job](#) 메서드의 Tags 인수를 사용하여 태그를 추가합니다. 자세한 내용은 [StartTranscriptionJob](#) 및 [Tag](#)을(를) 참조하세요.

기능별, 시나리오 및 교차 서비스 예를 포함하여 AWS SDK를 사용하는 추가 예시는 [AWS SDKs를 사용한 Amazon Transcribe의 코드 예제](#) 장을 참조하세요.

```

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)
```

화자 분할(분할)

화자 분할을 사용하면 트랜스크립션 출력에서 서로 다른 화자를 구분할 수 있습니다.는 최대 30개의 고유한 화자를 구분할 Amazon Transcribe 수 있으며 각 고유한 화자의 텍스트에 고유한 값(spk_0~)으로 레이블을 지정할 수 있습니다 spk_9.

화자 분할이 활성화된 요청에는 [표준 트랜스크립트 섹션](#)(transcripts 및 items) 외에도 speaker_labels 섹션이 포함됩니다. 이 섹션은 화자별로 그룹화되어 있으며 화자 레이블과 타임스탬프를 포함하여 각 발화에 대한 정보를 포함합니다.

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

화자 분할(화자 2명)이 포함된 전체 트랜스크립트 예시를 보려면 [분할 출력 예시\(배치\)](#)를 참조하세요.

배치 트랜스크립션의 화자 분할

배치 트랜스크립션에서 화자를 분할하려면 다음 예를 참조하세요.

AWS Management Console

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창에서 트랜스크립션 작업을 선택한 다음 작업 생성(오른쪽 상단)을 선택합니다. 그러면 작업 세부 정보 지정 페이지가 열립니다.

Specify job details [Info](#)

Job settings

Name

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

Model type [Info](#)

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

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

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

Language settings

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

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

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

Language

Choose the language of the input audio.

► **Additional settings**

3. 작업 세부 정보 지정 페이지에 포함하려는 필드를 모두 채운 후 다음을 선택합니다. 그러면 작업 구성 - 선택 사항 페이지로 이동합니다.

화자 분할을 활성화하려면 오디오 설정에서 오디오 식별을 선택합니다. 그런 다음 화자 분할을 선택하고 화자 수를 지정합니다.

Audio settings

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

Audio identification type

☐ Channel identification

☒ Speaker partitioning

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

The maximum number of speakers is 10.

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

4. 작업 생성을 선택하여 트랜스크립션 작업을 실행합니다.

AWS CLI

이 예시에서는 [start-transcription-job](#)을 사용합니다. 자세한 내용은 [StartTranscriptionJob](#) 단원을 참조하십시오.

```
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
```

다음은 [start-transcription-job](#) 명령을 사용하는 또 다른 예 및 해당 작업과 함께 화자 분할을 활성화하는 요청 본문입니다.

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

my-first-transcription-job.json 파일에는 다음과 같은 요청 본문이 포함되어 있습니다.

```
{
  "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 for Python (Boto3)

이 예제에서는 AWS SDK for Python (Boto3)를 사용하여 [start_transcription_job](#) 메서드를 사용하여 채널을 식별합니다. 자세한 내용은 [StartTranscriptionJob](#)을 참조하세요.

```
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)
```

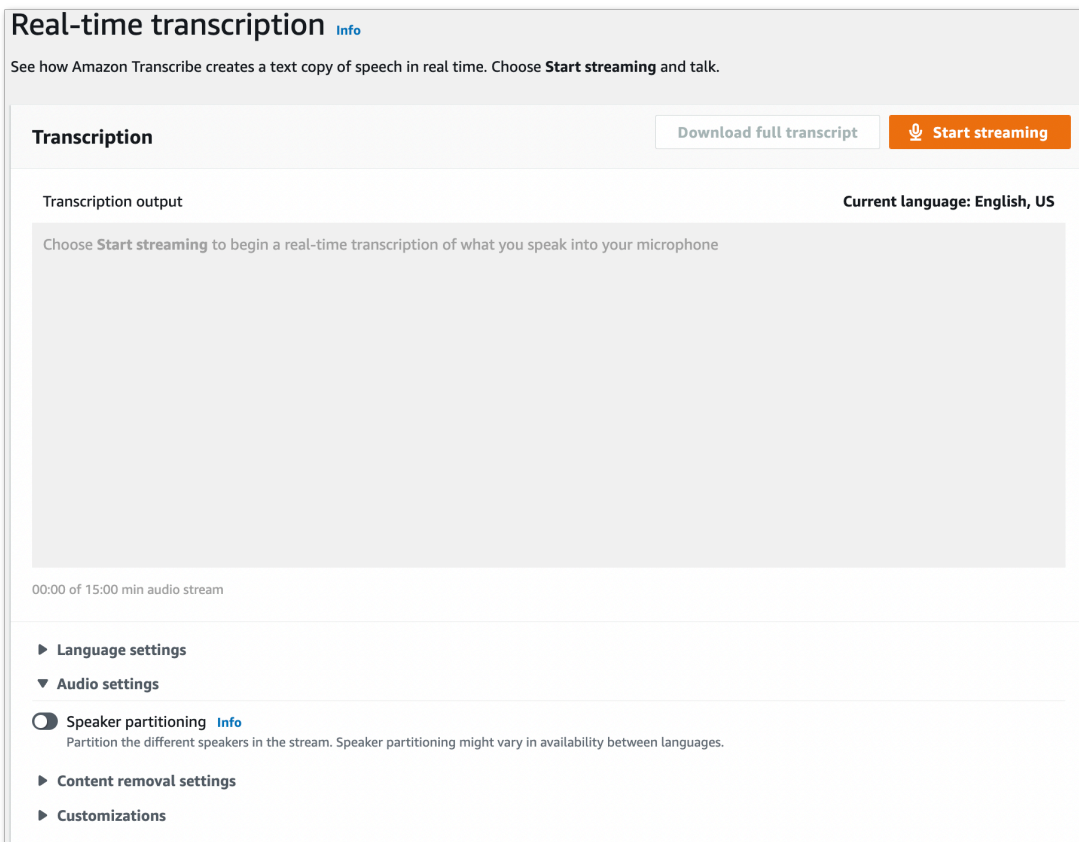
```
print(status)
```

스트리밍 트랜스크립션의 화자 분할

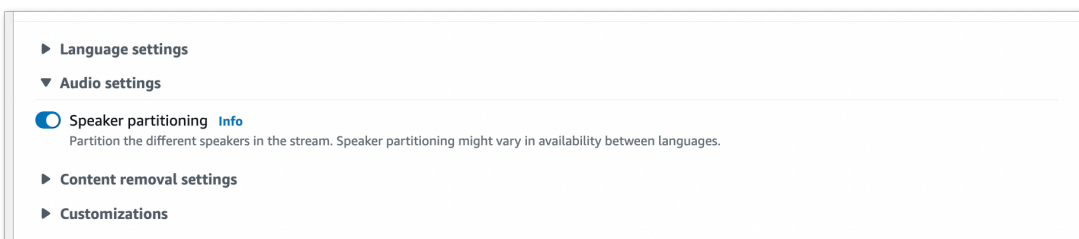
스트리밍 트랜스크립션에서 화자를 분할하려면 다음 예를 참조하세요.

스트리밍 트랜스크립션

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창에서 실시간 트랜스크립션을 선택합니다. 오디오 설정까지 아래로 스크롤하고 최소화된 경우 이 필드를 확장합니다.



3. 화자 분할을 켭니다.



- 이제 스트림을 트랜스크립션할 준비가 되었습니다. 스트리밍 시작을 선택하고 말하기 시작합니다. 구술을 끝내려면 스트리밍 중지를 선택합니다.

HTTP/2 스트림

이 예시에서는 트랜스크립션 출력에서 화자를 분할하는 HTTP/2 요청을 생성합니다. 에서 HTTP/2 스트리밍을 사용하는 방법에 대한 자세한 내용은 섹션을 Amazon Transcribe참조하세요[HTTP/2 스트림 설정](#). Amazon Transcribe 관련 파라미터 및 헤더에 대한 자세한 내용은 [StartStreamTranscription](#)을 참조하세요.

```
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
```

파라미터 정의는 [API 참조](#)에서 찾을 수 있습니다. 모든 AWSAPI 작업에 공통적인 파라미터는 [공통 파라미터](#) 섹션에 나열되어 있습니다.

WebSocket 스트림

이 예시에서는 트랜스크립션 출력에서 화자를 구분하는 미리 서명된 URL을 생성합니다. 가독성을 높이기 위해 줄바꿈이 추가되었습니다. 에서 WebSocket 스트림을 사용하는 방법에 대한 자세한 내용은 섹션을 Amazon Transcribe참조하세요[WebSocket 스트림 설정](#). 파라미터에 대한 자세한 내용은 [StartStreamTranscription](#)을 참조하세요.

```
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
```

파라미터 정의는 [API 참조](#)에서 찾을 수 있습니다. 모든 AWSAPI 작업에 공통적인 파라미터는 [공통 파라미터](#) 섹션에 나열되어 있습니다.

분할 출력 예시(배치)

다음은 분할이 활성화된 배치 트랜스크립션의 출력 예시입니다.

```
{
  "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"  
}
```


다중 채널 오디오 트랜스크립션

오디오에 두 개의 채널이 있는 경우 채널 식별을 사용하여 각 채널의 스피치를 개별적으로 트랜스크립션할 수 있습니다. Amazon Transcribe는 현재 두 개 이상의 채널이 있는 오디오를 지원하지 않습니다.

트랜스크립트에서 채널에는 `ch_0` 및 `ch_1` 레이블이 지정됩니다.

채널 식별이 활성화된 요청에는 [표준 트랜스크립트 섹션](#)(`transcripts` 및 `items`) 외에도 `channel_labels` 섹션이 포함됩니다. 이 섹션에는 채널별로 그룹화된 각 발화 또는 구두점 및 관련 채널 레이블, 타임스탬프 및 신뢰도 점수가 포함되어 있습니다.

```
"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  
  },
```

한 채널에 있는 사람이 별도의 채널에 있는 사람과 동시에 발언하는 경우 각 채널의 타임스탬프는 개인이 서로 말하는 동안 겹칩니다.

채널 식별을 사용한 전체 트랜스크립트 예시를 보려면 [채널 식별 출력 예시\(배치\)](#)를 참조하세요.

배치 트랜스크립션에서 채널 식별 사용

배치 트랜스크립션에서 채널을 식별하려면 AWS Management Console, AWS CLI 또는 AWS SDK를 사용할 수 있습니다. 예를 보려면 다음을 참조하세요.

AWS Management Console

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창에서 트랜스크립션 작업을 선택한 다음 작업 생성(오른쪽 상단)을 선택합니다. 그러면 작업 세부 정보 지정 페이지가 열립니다.

Specify job details [Info](#)

Job settings

Name

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

Model type [Info](#)

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

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

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

Language settings

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

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

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

Language

Choose the language of the input audio.

► **Additional settings**

- 작업 세부 정보 지정 페이지에 포함하려는 필드를 모두 채운 후 다음을 선택합니다. 그러면 작업 구성 - 선택 사항 페이지로 이동합니다.

오디오 설정 패널에서 채널 식별('오디오 식별 유형' 제목 아래)을 선택합니다.

Audio settings

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

Audio identification type

☒ **Channel identification**
☐ **Speaker partitioning**

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

4. 작업 생성을 선택하여 트랜스크립션 작업을 실행합니다.

AWS CLI

이 예시에서는 [start-transcription-job](#)을 사용합니다. 자세한 내용은 [StartTranscriptionJob](#) 단원을 참조하십시오.

```
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
```

다음은 [start-transcription-job](#) 명령을 사용하는 또 다른 예 및 해당 작업과 함께 채널 식별을 활성화하는 요청 본문입니다.

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

my-first-transcription-job.json 파일에는 다음과 같은 요청 본문이 포함되어 있습니다.

```
{  
  "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 for Python (Boto3)

이 예제에서는 AWS SDK for Python (Boto3)를 사용하여 [start_transcription_job](#) 메서드를 사용하여 채널을 식별합니다. 자세한 내용은 [StartTranscriptionJob](#)을 참조하세요.

```
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)
```

스트리밍 트랜스크립션에서 채널 식별 사용

스트리밍 트랜스크립션에서 채널을 식별하려면 HTTP/2 또는 WebSocket을 사용할 수 있습니다. 예를 보려면 다음을 참조하세요.

HTTP/2 스트림

이 예시에서는 트랜스크립션 출력에서 채널을 구분하는 HTTP/2 요청을 생성합니다. 에서 HTTP/2 스트리밍을 사용하는 방법에 대한 자세한 내용은 섹션을 Amazon Transcribe참조하세요[HTTP/2 스트림](#)

설정. Amazon Transcribe 관련 파라미터 및 헤더에 대한 자세한 내용은 [StartStreamTranscription](#)을 참조하세요.

```
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
```

파라미터 정의는 [API 참조](#)에서 찾을 수 있습니다. 모든 AWSAPI 작업에 공통적인 파라미터는 [공통 파라미터](#) 섹션에 나열되어 있습니다.

WebSocket 스트림

이 예시에서는 트랜스크립션 출력에서 채널을 구분하는 미리 서명된 URL을 생성합니다. 가독성을 높이기 위해 줄바꿈이 추가되었습니다. 에서 WebSocket 스트림을 사용하는 방법에 대한 자세한 내용은 섹션을 Amazon Transcribe참조하세요[WebSocket 스트림 설정](#). 파라미터에 대한 자세한 내용은 [StartStreamTranscription](#)을 참조하세요.

```
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
```

파라미터 정의는 [API 참조](#)에서 찾을 수 있습니다. 모든 AWSAPI 작업에 공통적인 파라미터는 [공통 파라미터](#) 섹션에 나열되어 있습니다.

채널 식별 출력 예시(배치)

다음은 채널 식별이 활성화된 배치 트랜스크립션의 출력 예시입니다.

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

미디어의 지배적 언어 식별

Amazon Transcribe를 사용하면 언어 코드를 지정하지 않고도 미디어에서 사용되는 언어를 자동으로 식별할 수 있습니다.

[배치 언어 식별](#)은 미디어 파일에서 사용되는 지배적 언어를 식별할 수 있으며, 미디어에 여러 언어가 포함된 경우 사용되는 모든 언어를 식별할 수 있습니다. 언어 식별 정확도를 높이기 위해 미디어에 있을 것으로 생각되는 두 개 이상의 언어 목록을 선택적으로 제공할 수 있습니다.

[스트리밍 언어 식별](#)은 채널당 하나의 언어(최대 2개 채널 지원)를 식별할 수 있으며, 스트림에 여러 언어가 포함된 경우 사용되는 모든 언어를 식별할 수 있습니다. 스트리밍 요청에는 요청에 최소 두 개의 추가 언어 옵션이 포함되어야 합니다. 언어 옵션을 제공하면 언어 식별이 더 빨라집니다. Amazon Transcribe가 언어를 더 빨리 식별할수록 스트림의 처음 몇 초 동안 발생하는 데이터 손실의 가능성이 더 줄어듭니다.

Important

배치 트랜스크립션과 스트리밍 트랜스크립션은 서로 다른 언어를 지원합니다. 자세한 내용은 [지원되는 언어 테이블](#)의 데이터 입력 열을 참조하세요. 참고로 스웨덴어와 베트남어는 현재 언어 식별이 지원되지 않습니다.

언어 식별을 통한 모니터링 및 이벤트에 대한 자세한 내용은 [언어 식별 이벤트](#)를 참조하세요.

배치 트랜스크립션 작업을 통한 언어 식별

배치 언어 식별을 사용하여 미디어 파일의 언어를 자동으로 식별합니다.

미디어에 한 가지 언어만 포함된 경우 [단일 언어 식별](#)을 활성화하여 미디어 파일에서 사용되는 지배적 언어를 식별하고 이 언어만 사용하여 트랜스크립트를 생성할 수 있습니다.

미디어에 두 개 이상의 언어가 포함된 경우 [다국어 식별](#)을 활성화하여 미디어 파일에서 사용되는 모든 언어를 식별하고 식별된 각 언어를 사용하여 트랜스크립트를 생성할 수 있습니다. 다국어 트랜스크립트가 생성된다는 점에 유의하세요. 와 같은 다른 서비스를 사용하여 트랜스크립트 Amazon Translate를 번역할 수 있습니다.

지원되는 언어 및 관련 언어 코드의 전체 목록은 [지원되는 언어](#) 테이블을 참조하세요.

최상의 결과를 얻으려면 미디어 파일에 30초 이상의 음성이 포함되어 있어야 합니다.

AWS Management Console AWS CLI 및 AWS Python SDK를 사용한 사용 예제는 섹션을 참조하세요. [요배치 트랜스크립션 작업을 통한 언어 식별 사용](#).

다국어 오디오의 언어 식별

다국어 식별은 다국어 미디어 파일을 위한 것으로 미디어에서 사용되는 [지원되는 언어](#) 모두를 반영하는 트랜스크립트를 제공합니다. 즉, 화자가 대화 도중에 언어를 바꾸거나 각 참가자가 서로 다른 언어를 사용하는 경우 트랜스크립션 출력은 각 언어를 올바르게 감지하고 트랜스크립션합니다. 예를 들어 미디어에 미국 영어(en-US)와 힌디어(hi-IN)를 번갈아 사용하는 이중 언어 화자가 포함되어 있는 경우 다국어 식별을 통해 미국 영어는 en-US로 힌디어는 hi-IN으로 식별하고 트랜스크립션할 수 있습니다.

이는 한 가지 지배적 언어만 사용하여 트랜스크립트를 작성하는 단일 언어 식별과는 다릅니다. 이 경우 지배적 언어가 아닌 음성 언어는 부정확하게 트랜스크립션됩니다.

Note

현재 다국어 식별에서는 교정 및 사용자 지정 언어 모델이 지원되지 않습니다.

Note

현재 다국어 식별이 지원되는 언어는 다음과 같습니다. en-AB, en-AU, en-GB, en-IE, en-IN, en-NZ, en-US, en-WL, en-ZA, es-ES, es-US, fr-CA, fr-FR, zh-CN, zh-TW, pt-BR, pt-PT, de-CH, de-DE, af-ZA, ar-AE, da-DK, he-IL, hi-IN, id-ID, fa-IR, it-IT, ja-JP, ko-KR, ms-MY, nl-NL, ru-RU, ta-IN, te-IN, th-TH, tr-TR

다국어 트랜스크립트에는 감지된 언어 및 미디어에서 각 언어가 사용된 총 시간이 요약되어 있습니다. 다음은 그 예입니다.

```
"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
      }
    ]
  }
}

```

언어 식별 정확도 향상

언어 식별을 사용하면 미디어에 있을 것으로 생각되는 언어 목록을 포함할 수 있습니다. 언어 옵션(LanguageOptions)을 포함하면 오디오를 올바른 언어와 일치시킬 때 지정한 언어만 Amazon Transcribe 사용하도록 제한되므로 언어 식별 속도가 빨라지고 올바른 언어 언어 할당과 관련된 정확도가 향상될 수 있습니다.

언어 코드를 포함하려면 두 개 이상 포함해야 합니다. 포함할 수 있는 언어 코드 수에는 제한이 없지만 최적의 효율성과 정확도를 위해 2~5개 사이를 사용하는 것이 좋습니다.

Note

요청에 언어 코드를 포함하고 제공한 언어 코드가 오디오에서 식별된 언어와 일치하지 않는 경우는 지정된 언어 코드에서 가장 가까운 언어 일치를 Amazon Transcribe 선택합니다. 그런 다음 해당 언어로 트랜스크립트를 생성합니다. 예를 들어 미디어가 미국 영어(en-US)이고 언어 코드 zh-CN, fr-FR를 Amazon Transcribe 제공하는 경우 미디어를 독일어(de-DE)와 일치시키고 독일어 de-DE Amazon Transcribe 트랜스크립션을 생성할 수 있습니다. 언어 코드와 음성 언어가 일치하지 않으면 트랜스크립트가 정확하지 않을 수 있으므로 언어 코드를 포함할 때는 주의를 기울이는 것이 좋습니다.

언어 식별을 다른 Amazon Transcribe 기능과 결합

배치 언어 식별을 다른 Amazon Transcribe 기능과 함께 사용할 수 있습니다. 언어 식별을 다른 기능과 결합하는 경우 해당 기능에서 지원되는 언어로 제한됩니다. 예를 들어 콘텐츠 편집과 함께 언어 식별을 사용하는 경우, 편집이 가능한 유일한 언어이므로 미국 영어(en-US) 또는 미국 스페인어(es-US)로 제한됩니다. 자세한 내용은 [지원되는 언어 및 언어별 기능](#) 섹션을 참조하세요.

Important

콘텐츠 편집이 활성화된 상태에서 자동 언어 식별을 사용하고 오디오에 미국 영어(en-US) 또는 미국 스페인어(es-US) 이외의 언어가 포함된 경우 미국 영어 혹은 미국 스페인어 콘텐츠만 트랜스크립트에서 편집됩니다. 다른 언어는 편집할 수 없으며 경고나 작업 실패도 없습니다.

사용자 지정 언어 모델, 사용자 지정 어휘 및 사용자 지정 어휘 필터

언어 식별 요청에 사용자 지정 언어 모델, 사용자 지정 어휘 또는 사용자 지정 어휘 필터를 하나 이상 추가하려면 [LanguageIdSettings](#) 파라미터를 포함해야 합니다. 그런 다음 해당하는 사용자 지정 언어 모델, 사용자 지정 어휘 및 사용자 지정 어휘 필터를 사용하여 언어 코드를 지정할 수 있습니다. 다국어 식별은 사용자 지정 언어 모델을 지원하지 않는다는 점에 유의하세요.

올바른 언어 방언을 식별할 수 있도록 [LanguageIdSettings](#)을 사용할 때 `LanguageOptions`을 포함하는 것이 좋습니다. 예를 들어 사용자 en-US 지정 어휘를 지정했지만 미디어에서 사용되는 언어가 라고 Amazon Transcribe 판단되면 en-AU사용자 지정 어휘가 트랜스크립션에 적용되지 않습니다. `LanguageOptions`을 포함하고 유일한 영어 방언으로 en-US를 지정하는 경우 사용자 지정 어휘가 트랜스크립션에 적용됩니다.

요청의 [LanguageIdSettings](#) 예시는 [배치 트랜스크립션 작업을 통한 언어 식별 사용](#) 섹션의 AWS CLI 및 AWS SDK 드롭다운 패널에서 옵션 2를 참조하세요.

배치 트랜스크립션 작업을 통한 언어 식별 사용

AWS Management Console, AWS CLI 또는 AWS SDK를 사용하여 배치 트랜스크립션 작업에서 자동 언어 식별을 사용할 수 있습니다. 예를 보려면 다음을 참조하세요.

AWS Management Console

1. [AWS Management Console](#)에 로그인합니다.

2. 탐색 창에서 트랜스크립션 작업을 선택한 다음 작업 생성(오른쪽 상단)을 선택합니다. 그러면 작업 세부 정보 지정 페이지가 열립니다.
3. 작업 설정 패널에서 언어 설정 섹션을 찾아 자동 언어 식별 또는 자동 다국어 식별을 선택합니다.

오디오 파일에 어떤 언어가 있는지 알고 있으면 (언어 선택 드롭다운 상자에서) 다국어 옵션을 선택할 수 있습니다. 언어 옵션을 제공하면 정확도를 높일 수 있지만 필수는 아닙니다.

Specify job details [Info](#)

Job settings

Name

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

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

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

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

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

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

Select languages

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

4. 작업 세부 정보 지정 페이지에 포함하려는 다른 필드를 모두 채운 후 다음을 선택합니다. 그러면 작업 구성 - 선택 사항 페이지로 이동합니다.

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

Audio settings

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

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

Content removal

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

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

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

Customization

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

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

5. 작업 생성을 선택하여 트랜스크립션 작업을 실행합니다.

AWS CLI

이 예시에서는 [start-transcription-job](#) 명령 및 `IdentifyLanguage` 파라미터를 사용합니다. 자세한 내용은 [StartTranscriptionJob](#) 및 [LanguageIdSettings](#) 섹션을 참조하세요.

옵션 1: `language-id-settings` 파라미터 없음. 요청에 사용자 지정 언어 모델, 사용자 지정 어휘 또는 사용자 지정 어휘 필터를 포함하지 않는 경우 이 옵션을 사용합니다. `language-options`은 선택 사항이며, 권장 사항은 아닙니다.

```
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"
```

옵션 2: language-id-settings 파라미터 있음. 요청에 사용자 지정 언어 모델, 사용자 지정 어휘 또는 사용자 지정 어휘 필터를 포함하는 경우 이 옵션을 사용합니다.

```
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
```

다음은 [start-transcription-job](#) 명령을 사용하는 또 다른 예 및 언어를 식별하는 요청 본문입니다.

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

my-first-language-id-job.json 파일에는 다음과 같은 요청 본문이 포함되어 있습니다.

옵션 1: LanguageIdSettings 파라미터 없음. 요청에 사용자 지정 언어 모델, 사용자 지정 어휘 또는 사용자 지정 어휘 필터를 포함하지 않는 경우 이 옵션을 사용합니다. LanguageOptions은 선택 사항이며, 권장 사항은 아닙니다.

```
{
  "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"
]
}

```

옵션 2: LanguageIdSettings 파라미터 있음. 요청에 사용자 지정 언어 모델, 사용자 지정 어휘 또는 사용자 지정 어휘 필터를 포함하는 경우 이 옵션을 사용합니다.

```

{
  "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 for Python (Boto3)

이 예제에서는 AWS SDK for Python (Boto3) 를 사용하여 [start_transcription_job](#) 메서드에 대한 IdentifyLanguage 인수를 사용하여 파일의 언어를 식별합니다. 자세한 내용은 [StartTranscriptionJob](#) 및 [LanguageIdSettings](#) 섹션을 참조하세요.

기능별, 시나리오 및 교차 서비스 예제 AWS SDKs를 사용하는 추가 예제는 [AWS SDKs를 사용한 Amazon Transcribe의 코드 예제](#)장을 참조하세요.

옵션 1: LanguageIdSettings 파라미터 없음. 요청에 사용자 지정 언어 모델, 사용자 지정 어휘 또는 사용자 지정 어휘 필터를 포함하지 않는 경우 이 옵션을 사용합니다. LanguageOptions은 선택 사항이며, 권장 사항은 아닙니다.

```
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)
```

옵션 2: LanguageIdSettings 파라미터 있음. 요청에 사용자 지정 언어 모델, 사용자 지정 어휘 또는 사용자 지정 어휘 필터를 포함하는 경우 이 옵션을 사용합니다.

```
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)

```

스트리밍 트랜스크립션을 통한 언어 식별

스트리밍 언어 식별은 미디어 스트림에서 사용되는 지배적 언어를 식별할 수 있습니다. 는 언어를 식별하려면 최소 1초의 음성이 Amazon Transcribe 필요합니다.

스트림에 한 가지 언어만 포함된 경우 단일 언어 식별을 활성화하여 미디어 파일에서 사용되는 지배적 언어를 식별하고 이 언어만 사용하여 트랜스크립트를 생성할 수 있습니다.

스트림에 두 개 이상의 언어가 포함된 경우 다국어 식별을 활성화하여 스트림에서 사용되는 모든 언어를 식별하고 식별된 각 언어를 사용하여 트랜스크립트를 생성할 수 있습니다. 다국어 트랜스크립트가

생성된다는 점에 유의하세요. 와 같은 다른 서비스를 사용하여 트랜스크립트 Amazon Transcribe를 번역할 수 있습니다.

스트리밍 언어 식별을 사용하려면 언어 코드를 두 개 이상 제공해야 하며 스트림당 언어당 언어 방언을 하나만 선택할 수 있습니다. 즉, 동일한 트랜스크립션에 대해 en-US 및 en-AU를 언어 옵션으로 선택할 수 없습니다.

또한 제공하는 언어 코드 세트에서 기본 언어를 선택할 수도 있습니다. 기본 언어를 추가하면 언어 식별 프로세스의 속도를 높일 수 있으므로 짧은 오디오 클립에 유용합니다.

Important

제공한 언어 코드가 오디오에서 식별된 언어와 일치하지 않는 경우는 지정된 언어 코드에서 가장 가까운 언어 일치를 Amazon Transcribe 선택합니다. 그런 다음 해당 언어로 트랜스크립트를 생성합니다. 예를 들어 미디어가 미국 영어(en-US)이고 언어 코드 zh-CN, fr-FR 및 Amazon Transcribe 제공하는 경우 de-DE Amazon Transcribe 는 미디어를 독일어(de-DE)와 일치시키고 독일어 트랜스크립션을 생성할 수 있습니다. 언어 코드와 음성 언어가 일치하지 않으면 트랜스크립트가 정확하지 않을 수 있으므로 언어 코드를 포함할 때는 주의를 기울이는 것이 좋습니다.

미디어에 두 개의 채널이 포함된 경우는 각 채널에서 사용되는 지배적 언어를 식별할 Amazon Transcribe 수 있습니다. 이 경우 [ChannelIdentification](#) 파라미터를 true로 설정하면 각 채널이 개별적으로 트랜스크립션됩니다. 이 파라미터의 기본값은 false입니다. 변경하지 않으면 첫 번째 채널만 트랜스크립션되고 하나의 언어만 식별됩니다.

스트리밍 언어 식별은 사용자 지정 언어 모델 또는 편집과 결합할 수 없습니다. 언어 식별을 다른 기능과 결합하는 경우 해당 기능과 스트리밍 트랜스크립션에서 지원되는 언어로 제한됩니다. [지원되는 언어](#)를 참조하세요.

Note

스트리밍 언어 식별에 지원되는 오디오 형식은 PCM 및 FLAC뿐입니다. 다국어 식별의 경우 PCM만 지원됩니다.

다국어 오디오의 언어 식별

다국어 식별은 다국어 스트림을 위한 것으로 스트림에서 사용되는 지원되는 언어 모두를 반영하는 트랜스크립트를 제공합니다. 즉, 화자가 대화 도중에 언어를 바꾸거나 각 참가자가 서로 다른 언어를 사용하는 경우 트랜스크립션 출력은 각 언어를 올바르게 감지하고 트랜스크립션합니다.

예를 들어 스트림에 미국 영어(en-US)와 힌디어(hi-IN)를 번갈아 사용하는 이중 언어 화자가 포함되어 있는 경우 다국어 식별을 통해 미국 영어는 en-US로 힌디어는 hi-IN으로 식별하고 트랜스크립션할 수 있습니다. 이는 한 가지 지배적 언어만 사용하여 트랜스크립트를 작성하는 단일 언어 식별과는 다릅니다. 이 경우 지배적 언어가 아닌 음성 언어는 부정확하게 트랜스크립션됩니다.

Note

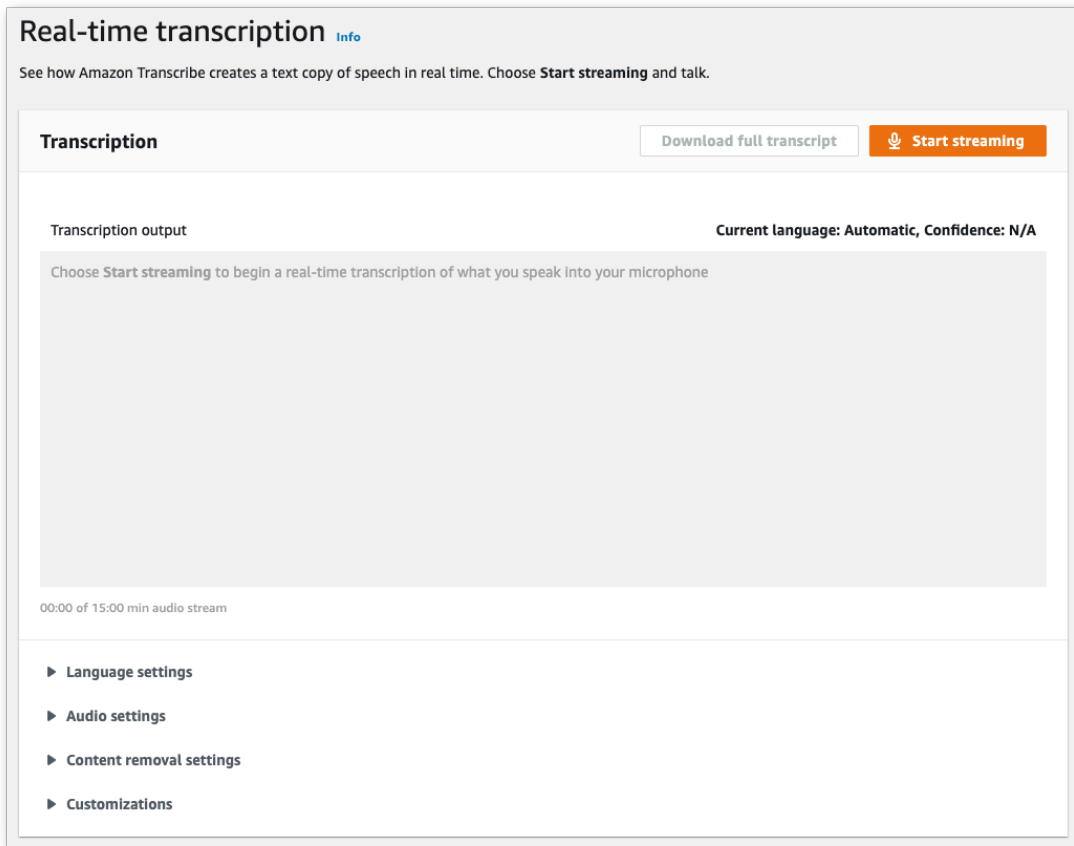
현재 다국어 식별에서는 교정 및 사용자 지정 언어 모델이 지원되지 않습니다.

스트리밍 미디어를 통한 언어 식별 사용

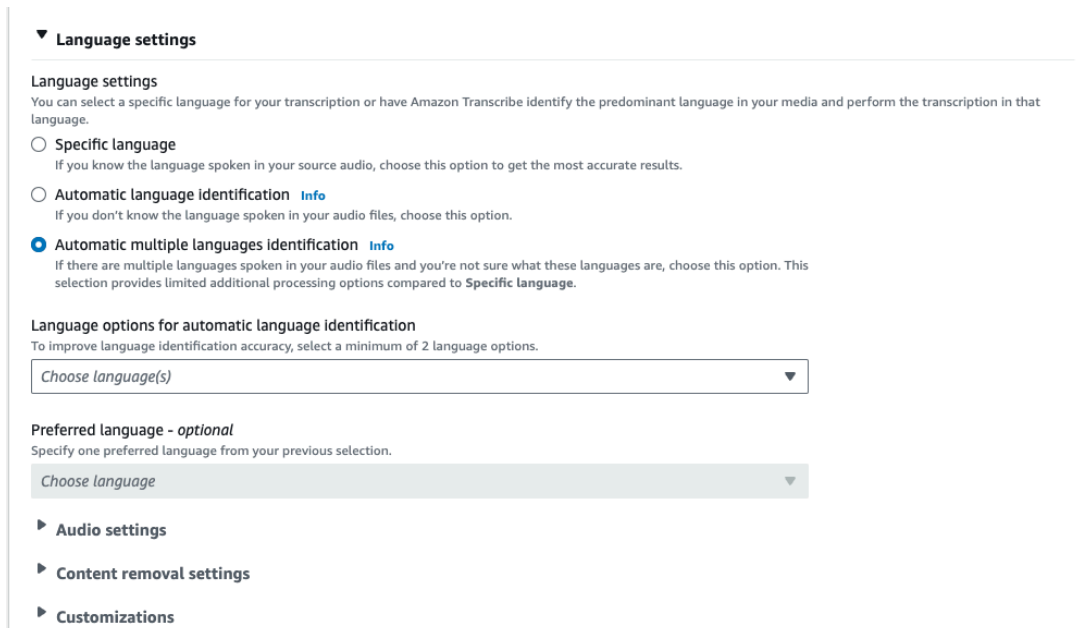
AWS Management Console, HTTP/2 또는 WebSockets를 사용하여 스트리밍 트랜스크립션에서 자동 언어 식별을 사용할 수 있습니다. 예를 보려면 다음을 참조하세요.

AWS Management Console

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창에서 실시간 트랜스크립션을 선택합니다. 언어 설정까지 아래로 스크롤하고 최소화된 경우 이 필드를 확장합니다.



3. 자동 언어 식별 또는 자동 다국어 식별을 선택합니다.



4. 트랜스크립션에 사용할 언어 코드를 최소 두 개 제공합니다. 언어당 방언을 하나만 제공할 수 있다는 점에 유의하세요. 예를 들어 동일한 트랜스크립션에 대해 en-US 및 en-GB 모두를 언어 옵션으로 선택할 수 없습니다.

▼ Language settings

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

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

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

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

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

Choose language(s) ▼

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

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

Choose language ▲

Q

None

English, US (en-US)

French, CA (fr-CA)

5. (선택 사항) 이전 단계에서 선택한 언어 하위 집합 중에서 트랜스크립트에 사용할 기본 언어를 선택할 수 있습니다.

▼ Language settings

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

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

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

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

Choose language(s) ▼

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

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

Choose language ▲

None

English, US (en-US)

French, CA (fr-CA)

► Customizations

6. 이제 스트림을 트랜스크립션할 준비가 되었습니다. 스트리밍 시작을 선택하고 말하기 시작합니다. 구슬을 끝내려면 스트리밍 중지를 선택합니다.

HTTP/2 스트림

이 예시에서는 언어 식별이 활성화된 HTTP/2 요청을 생성합니다. 에서 HTTP/2 스트리밍을 사용하는 방법에 대한 자세한 내용은 섹션을 [Amazon Transcribe참조하세요HTTP/2 스트림 설정](#). 관련 파라미터 및 헤더에 대한 자세한 내용은 섹션을 [Amazon Transcribe참조하세요StartStreamTranscription](#).

```
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
```

이 예시에서는 다국어 식별이 활성화된 HTTP/2 요청을 생성합니다. 에서 HTTP/2 스트리밍을 사용하는 방법에 대한 자세한 내용은 섹션을 [Amazon Transcribe참조하세요HTTP/2 스트림 설정](#). 관련 파라미터 및 헤더에 대한 자세한 내용은 섹션을 [Amazon Transcribe참조하세요StartStreamTranscription](#).

```
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
```

요청에 `identify-language` 또는 `identify-multiple-languages`를 사용하는 경우 `language-options` 역시 포함해야 합니다. 동일한 요청에서 `language-code` 및 `identify-language` 둘 다 사용할 수는 없습니다.

파라미터 정의는 [API 참조](#)에서 찾을 수 있습니다. 모든 AWS API 작업에 공통적인 파라미터는 [공통 파라미터](#) 섹션에 나열되어 있습니다.

WebSocket 스트림

이 예시에서는 WebSocket 스트림에서 언어 식별을 사용하는 미리 서명된 URL을 만듭니다. 가독성을 높이기 위해 줄바꿈이 추가되었습니다. 에서 WebSocket 스트림을 사용하는 방법에 대한 자세한 내용은 섹션을 Amazon Transcribe참조하세요 [WebSocket 스트림 설정](#). 파라미터에 대한 자세한 내용은 [StartStreamTranscription](#)을 참조하세요.

```
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
```

이 예시에서는 WebSocket 스트림에서 다국어 식별을 사용하는 미리 서명된 URL을 만듭니다. 가독성을 높이기 위해 줄바꿈이 추가되었습니다. 에서 WebSocket 스트림을 사용하는 방법에 대한 자세한 내용은 섹션을 Amazon Transcribe참조하세요 [WebSocket 스트림 설정](#). 파라미터에 대한 자세한 내용은 [StartStreamTranscription](#)을 참조하세요.

```
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
```

요청에 `identify-language` 또는 `identify-multiple-languages`를 사용하는 경우 `language-options` 역시 포함해야 합니다. 동일한 요청에서 `language-code` 및 `identify-language` 둘 다 사용할 수는 없습니다.

파라미터 정의는 [API 참조](#)에서 찾을 수 있습니다. 모든 AWS API 작업에 공통적인 파라미터는 [공통 파라미터](#) 섹션에 나열되어 있습니다.

대체 트랜스크립션

가 오디오 Amazon Transcribe를 트랜스크립션하면 동일한 트랜스크립트의 다른 버전을 생성하고 각 버전에 신뢰도 점수를 할당합니다. 일반적인 트랜스크립션에서는 신뢰도 점수가 가장 높은 버전만 받아볼 수 있습니다.

대체 트랜스크립션을 켜면는 신뢰도 수준이 낮은 다른 버전의 트랜스크립트를 Amazon Transcribe 반환합니다. 최대 10개의 대체 트랜스크립션을 반환하도록 선택할 수 있습니다. 가 Amazon Transcribe 식별하는 것보다 더 많은 수의 대안을 지정하면 실제 대안 수만 반환됩니다.

모든 대체는 동일한 트랜스크립션 출력 파일에 있으며 세그먼트 수준에서 표시됩니다. 세그먼트는 화자 변경 또는 오디오의 일시 중지처럼, 말하기 도중 자연스럽게 쉬는 지점입니다.

대체 트랜스크립션은 배치 트랜스크립션에만 사용할 수 있습니다.

트랜스크립션 출력의 구조는 다음과 같습니다. 코드 예시의 줄임표(...)는 간결성을 위해 콘텐츠가 제거된 위치를 나타냅니다.

1. 지정된 세그먼트에 대한 완전한 최종 트랜스크립션.

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

```

2. 이전 transcript 섹션의 각 단어에 대한 신뢰도 점수.

```
"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. 대체 트랜스크립트는 트랜스크립션 출력의 `segments` 부분에 있습니다. 각 세그먼트에 대한 대체는 신뢰도 점수를 내림차순으로 정렬됩니다.

```

"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. 트랜스크립션 출력 종료 시 상태.

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

대체 트랜스크립션 요청

AWS Management Console, AWS CLI 또는 AWS SDK를 사용하여 대체 트랜스크립션을 요청할 수 있습니다. 예를 보려면 다음을 참조하세요.

AWS Management Console

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창에서 트랜스크립션 작업을 선택한 다음 작업 생성(오른쪽 상단)을 선택합니다. 그러면 작업 세부 정보 지정 페이지가 열립니다.

Specify job details [Info](#)

Job settings

Name

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

Model type [Info](#)

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

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

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

Language settings

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

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

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

Language

Choose the language of the input audio.

► **Additional settings**

- 작업 세부 정보 지정 페이지에 포함하려는 필드를 모두 채운 후 다음을 선택합니다. 그러면 작업 구성 - 선택 사항 페이지로 이동합니다.

대체 결과를 선택하고 트랜스크립트에 포함할 대체 트랜스크립션 결과의 최대 수를 지정합니다.

Configure job - optional [Info](#)

Audio settings

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

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

Maximum alternatives
Provide the number of alternative transcription to provide in the text output.

The maximum number of alternative results is 10.

4. 작업 생성을 선택하여 트랜스크립션 작업을 실행합니다.

AWS CLI

이 예시에서는 [start-transcription-job](#) 명령 및 ShowAlternatives 파라미터를 사용합니다. 자세한 내용은 [StartTranscriptionJob](#) 및 [ShowAlternatives](#) 섹션을 참조하세요.

단, 요청에 ShowAlternatives=true를 포함하는 경우 MaxAlternatives 역시 포함해야 합니다.

```
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
```

다음은 [start-transcription-job](#) 명령을 사용하는 또 다른 예 및 대체 트랜스크립션을 포함하는 요청 본문입니다.

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

my-first-alt-transcription-job.json 파일에는 다음과 같은 요청 본문이 포함되어 있습니다.

```
{
  "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 for Python (Boto3)

다음 예제에서는 AWS SDK for Python (Boto3)를 사용하여 [start_transcription_job](#) 메서드에 대한 ShowAlternatives 인수를 사용하여 대체 트랜스크립션을 요청합니다. 자세한 내용은 [StartTranscriptionJob](#) 및 [ShowAlternatives](#) 섹션을 참조하세요.

기능별, 시나리오 및 교차 서비스 예제를 포함하여 AWSSDKs를 사용하는 추가 예제는 [AWS SDKs를 사용한 Amazon Transcribe의 코드 예제](#)장을 참조하세요.

단, 요청에 'ShowAlternatives':True을 포함하는 경우 MaxAlternatives 역시 포함해야 합니다.

```
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)
```

사용자 지정 어휘 및 사용자 지정 언어 모델을 통한 트랜스크립션 정확도 향상

미디어에 브랜드 이름, 두문자어, 기술적 단어 및 전문 용어와 같은 도메인별 또는 비표준 용어가 포함된 경우 트랜스크립션 출력에서 이러한 용어를 올바르게 캡처하지 못할 Amazon Transcribe 수 있습니다.

트랜스크립션의 부정확성을 수정하고 특정 사용 사례에 맞게 출력을 사용자 지정하려면 [사용자 지정 어휘](#) 및 [사용자 지정 언어 모델](#)을 만들 수 있습니다.

- [사용자 지정 어휘](#)는 모든 컨텍스트에서 특정 단어의 인식 및 형식 모두를 조정하고 향상시키도록 설계되었습니다. 여기에는 단어를 Amazon Transcribe에 제공하는 것은 물론 선택적으로 발음 및 표시 형식을 제공하는 것도 포함됩니다.

Amazon Transcribe가 트랜스크립트에서 특정 용어를 올바르게 렌더링하지 않는 경우 이러한 용어를 표시할 Amazon Transcribe 방법을 알려주는 사용자 지정 어휘 파일을 생성할 수 있습니다. 이 단어별 접근 방식은 브랜드 이름 및 두문자어와 같은 용어를 수정하는 데 가장 적합합니다.

- [사용자 지정 언어 모델](#)은 용어와 관련된 컨텍스트를 캡처하도록 설계되었습니다. 여기에는 Amazon Transcribe 대량의 도메인별 텍스트 데이터를 제공하는 작업이 포함됩니다.

Amazon Transcribe가 기술 용어를 올바르게 렌더링하지 않거나 트랜스크립트에서 잘못된 호모폰을 사용하는 경우 도메인별 언어를 가르치는 사용자 지정 언어 모델을 생성할 수 Amazon Transcribe 있습니다. 예를 들어 사용자 지정 언어 모델은 'floe'(부빙)와 'flow'(선형 흐름)를 언제 사용해야 하는지 학습할 수 있습니다.

이 컨텍스트 인식 접근 방식은 대량의 도메인별 음성을 트랜스크립션하는 데 가장 적합합니다. 사용자 지정 언어 모델을 사용하면 사용자 지정 어휘만 사용할 때보다 정확도가 크게 향상될 수 있습니다. 배치 트랜스크립션을 사용하는 경우 요청에 사용자 지정 언어 모델과 사용자 지정 어휘를 모두 포함할 수 있습니다.

Tip

트랜스크립션 정확도를 극대화하려면 사용자 지정 어휘를 사용자 지정 언어 모델과 함께 사용하세요.

를 사용하여 사용자 지정 어휘를 생성하는 방법에 대한 비디오 데모는 [사용자 지정 어휘 사용을 AWS Management Console](#) 참조하세요.

사용자 지정 언어 모델을 만들고 사용하는 방법에 대한 비디오 데모는 [사용자 지정 언어 모델\(CLM\)을 사용하여 트랜스크립션 정확도 강화](#)를 참조하세요.

AWS Machine Learning 블로그로 더 자세히 알아보기

사용자 지정 어휘:

- [를 사용한 F1 레이스의 라이브 트랜스크립션 Amazon Transcribe](#)

사용자 지정 언어 모델:

- [speech-to-text트 간 성능을 극대화하기 위한 사용자 지정 언어 모델 구축 Amazon Transcribe](#)
- [Amazon Transcribe용 사용자 지정 언어 모델을 통한 강의 트랜스크립션 정확도 향상](#)

사용자 지정 어휘

사용자 지정 어휘를 사용하여 하나 이상의 특정 단어의 트랜스크립션 정확도를 향상시킵니다. 이러한 단어는 일반적으로 브랜드 이름 및 두문자어, 고유 명사 및 Amazon Transcribe가 올바르게 렌더링하지 않은 단어 등 도메인별 용어입니다.

지원되는 모든 언어에 사용자 지정 어휘를 사용할 수 있습니다. 사용자 지정 어휘에는 해당 언어의 [문자 집합](#)에 나열된 문자만 사용할 수 있다는 점에 유의하세요.

Important

Amazon Transcribe 사용 시 자체 데이터의 무결성에 대한 책임은 사용자에게 있습니다. 기밀 정보, 개인 정보(PII) 또는 보호 대상 건강 정보(PHI)를 사용자 지정 어휘에 입력하지 마세요.

사용자 지정 어휘를 생성할 때 고려할 사항:

- AWS 계정당 최대 100개의 사용자 지정 어휘 파일을 보유할 수 있습니다.
- 각 사용자 지정 어휘 파일의 크기 한도는 50Kb입니다.

- API를 사용하여 사용자 지정 어휘를 만드는 경우 어휘 파일은 텍스트(*.txt) 형식이어야 합니다. AWS Management Console을 사용하는 경우 어휘 파일은 텍스트(*.txt) 형식 또는 쉼표로 구분된 값(*.csv) 형식일 수 있습니다.
- 사용자 지정 어휘 내의 각 항목은 256자를 초과할 수 없습니다.
- 사용자 지정 어휘를 사용하려면 해당 어휘가 트랜스크립션과 동일한 AWS 리전에서 생성되어야 합니다.

Tip

AWS Management Console을 사용하여 사용자 지정 어휘를 테스트할 수 있습니다. 사용자 지정 어휘를 사용할 준비가 되면 AWS Management Console에 로그인하고 실시간 트랜스크립션을 선택한 후 사용자 지정으로 스크롤하고 사용자 지정 어휘를 컨 다음 드롭다운 목록에서 사용자 지정 어휘를 선택합니다. 그런 다음 스트리밍 시작을 선택합니다. 사용자 지정 어휘의 일부 단어를 마이크에 대고 말하여 제대로 렌더링되는지 확인합니다.

사용자 지정 어휘 테이블과 목록 비교

Important

목록 형식의 사용자 지정 어휘는 더 이상 사용되지 않습니다. 새 사용자 지정 어휘를 만들려면 [테이블 형식](#)을 사용하세요.

테이블을 사용하면 사용자 지정 어휘 내에서 단어의 입력과 출력에 대한 더 많은 옵션과 제어를 제공합니다. 테이블을 사용하면 출력을 세밀하게 조정할 수 있도록 여러 범주((Phrase and DisplayAs))를 지정해야 합니다.

목록에는 추가 옵션이 없으므로 트랜스크립트에 표시할 항목만 입력할 수 있으며 모든 스페이스는 하이픈으로 대체됩니다.

AWS Management Console, AWS CLI 및 AWS SDK는 모두 사용자 지정 어휘 테이블을 같은 방식으로 사용합니다. 목록은 각 메서드마다 다르게 사용되므로 메서드 간에 성공적으로 사용하려면 추가 서식이 필요할 수 있습니다.

자세한 정보는 [테이블을 사용하여 사용자 지정 어휘 생성](#) 및 [목록을 사용하여 사용자 지정 어휘 생성](#) 섹션을 참조하세요.

Amazon Augmented AI를 사용자 지정 어휘와 함께 사용하는 방법을 좀 더 자세히 알아보려면 [Amazon Transcribe와 함께 인적 검토 구축 시작](#)을 참조하세요.

사용자 지정 어휘와 관련된 API 작업

[CreateVocabulary](#), [DeleteVocabulary](#), [GetVocabulary](#), [ListVocabularies](#), [UpdateVocabulary](#)

테이블을 사용하여 사용자 지정 어휘 생성

사용자 지정 어휘를 만들 때는 테이블 형식을 사용하는 것이 좋습니다. 어휘 테이블은 네 개의 열 ((Phrase, SoundsLike, IPA, and DisplayAs))로 구성되어야 하며, 어떤 순서로든 포함될 수 있습니다.

구절	SoundsLike	IPA	DisplayAs
필수 사항입니다. 테이블의 모든 행에는 이 열의 항목이 포함되어야 합니다.	SoundsLike 는 더 이상 사용자 지정 어휘에 지원되지 않습니다. 열을 비워 두십시오.	IPA는 더 이상 사용자 지정 어휘에 지원되지 않습니다. 열을 비워 두십시오. 이 열의 모든 값은 무시됩니다. 이 열에 대한 지원은 향후 삭제될 예정입니다.	선택 사항입니다. 이 열의 행은 비워 둘 수 있습니다.
이 열에는 스페이스를 사용하지 마세요.	이 열의 모든 값은 무시됩니다. 이 열에 대한 지원은 향후 삭제될 예정입니다.	이 열에 대한 지원은 향후 삭제될 예정입니다.	이 열에는 스페이스를 사용할 수 있습니다.
항목에 여러 단어가 포함된 경우 각 단어를 하이픈(-)으로 구분합니다. 예: Andorra-la-Vella 또는 Los-Angeles .			트랜스크립션 출력에서 항목이 어떻게 보이길 원하는지 정의합니다. 예를 들어, Phrase 열의 Andorra-la-Vella 는 DisplayAs 열의 Andorra la Vella 입니다.
두문자어의 경우 발음되는 모든 문자를 마침표로 구분해야 합니다. 마지막 마침표도 발음해야 합니다. 두문자어가 복수형인 경우 두문자어와 's' 사이에 하			이 열의 행이 비어 있는 경우 Amazon Transcribe는 Phrase 열의 콘텐츠를 사용하여 출력을 결정합니다.

구절	SoundsLike	IPA	DisplayAs
이폰을 사용해야 합니다. 예를 들어, 'CLI'는 C.L.I. (C.L.I 아님)이고 'ABC'는 A.B.C.-s (A.B.C-s 아님)입니다.			이 열에 숫자(0-9)를 포함할 수 있습니다.
구절이 단어와 두문자어로 구성된 경우 이 두 구성 요소를 하이픈으로 구분해야 합니다. 예를 들어, 'DynamoDB'는 Dynamo-D.B. 입니다.			
이 열에 숫자를 포함하지 마세요. 숫자는 철자로 입력해야 합니다. 예를 들어, 'VX02Q'는 V.X.-zero-two-Q. 입니다.			

테이블을 만들 때 참고할 사항:

- 테이블에는 열 헤더 4개가 모두 포함되어야 합니다 (Phrase, SoundsLike, IPA, and DisplayAs). Phrase 열에는 각 행의 항목이 포함되어야 합니다. IPA 및 SoundsLike를 통해 발음 입력을 제공하는 기능은 더 이상 지원되지 않으며 열을 비워 둘 수 있습니다. 이러한 열의 모든 값은 무시됩니다.
- 각 열은 TAB 또는 쉼표(,)로 구분해야 합니다. 이는 사용자 지정 어휘 파일의 모든 행에 적용됩니다. 행에 빈 열이 있는 경우에도 각 열에 구분자(TAB 또는 쉼표)를 포함해야 합니다.
- IPA 및 DisplayAs 열 내에만 스페이스가 허용됩니다. 스페이스를 사용하여 열을 구분하지 마세요.
- IPA 및 SoundsLike는 더 이상 사용자 지정 어휘에 지원되지 않습니다. 열을 비워 두십시오. 이러한 열의 모든 값은 무시됩니다. 이 열에 대한 지원은 향후 삭제될 예정입니다.

- DisplayAs 열은 기호와 특수 문자(예: C++)를 지원합니다. 다른 모든 열은 해당 언어의 [문자 집합](#) 페이지에 나열된 문자를 지원합니다.
- Phrase 열에 숫자를 포함하려면 철자를 입력해야 합니다. 숫자(0-9)는 DisplayAs 열에서만 지원됩니다.
- 테이블을 LF 형식의 일반 텍스트(*.txt) 파일로 저장해야 합니다. 다른 형식(예: CRLF)을 사용하는 경우 사용자 지정 어휘를 처리할 수 없습니다.
- 사용자 지정 어휘 파일을 Amazon S3 버킷에 업로드하고 [CreateVocabulary](#)을 사용하여 처리해야 트랜스크립션 요청에 포함시킬 수 있습니다. 지침은 [사용자 지정 어휘 테이블 생성](#) 단원을 참조하세요.

Note

두문자어 또는 문자를 개별적으로 발음해야 하는 기타 단어는 마침표로 구분된 단일 문자로 입력합니다(A.B.C.). 두문자어의 복수 형태를 입력하려면(예: 'ABCs') 하이픈으로 두문자어에서 's'를 구분합니다(A.B.C.-s). 대문자 또는 소문자를 사용해 두문자어를 정의할 수 있습니다. 모든 언어에서 두문자어가 지원되는 것은 아닙니다. [지원되는 언어 및 언어별 기능](#)를 참조하세요.

다음은 샘플 사용자 지정 어휘 테이블입니다(여기서 **[TAB]**은 탭 문자를 나타냄).

```
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
```

시각적 명확성을 위해 동일한 테이블에 열이 정렬되어 있습니다. 사용자 지정 어휘 테이블의 열 사이에 스페이스를 추가하지 마세요. 위의 예와 같이 테이블이 잘못 정렬되어 보일 수 있습니다.

Phrase	[TAB] SoundsLike	[TAB] IPA	[TAB] DisplayAs
Los-Angeles	[TAB]	[TAB]	[TAB] Los Angeles

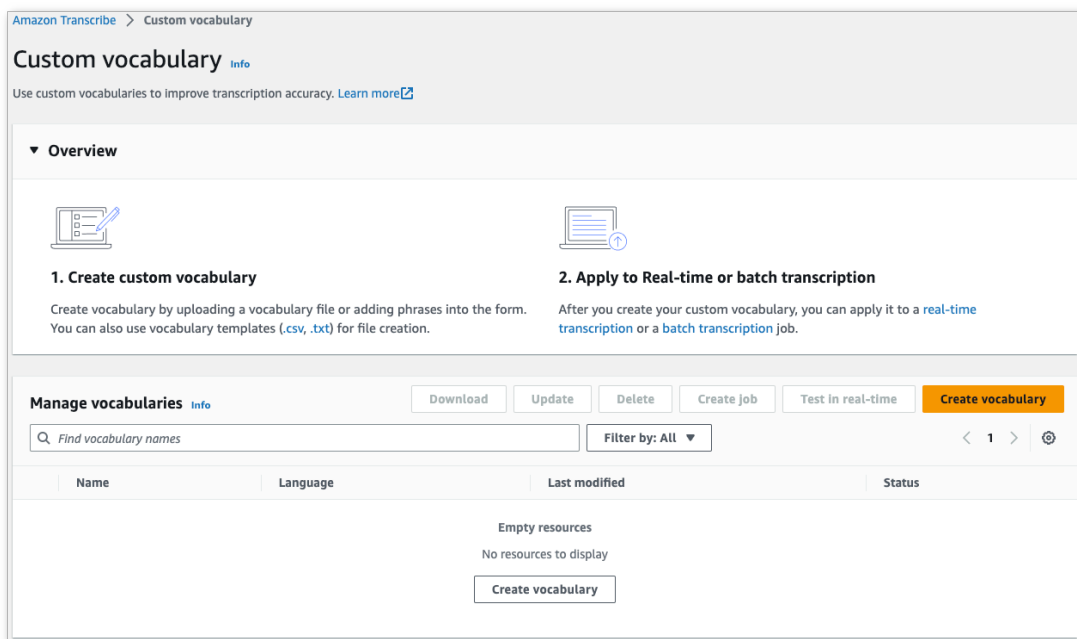
Eva-Maria	[TAB]	[TAB]	[TAB]
A.B.C.-s	[TAB]	[TAB]	[TAB]ABCs
amazon-dot-com	[TAB]	[TAB]	[TAB]amazon.com
C.L.I.	[TAB]	[TAB]	[TAB]CLI
Andorra-la-Vella	[TAB]	[TAB]	[TAB]Andorra la Vella
Dynamo-D.B.	[TAB]	[TAB]	[TAB]DynamoDB
V.X.-zero-two	[TAB]	[TAB]	[TAB]VX02
V.X.-zero-two-Q.	[TAB]	[TAB]	[TAB]VX02Q

사용자 지정 어휘 테이블 생성

Amazon Transcribe에서 사용할 사용자 지정 어휘 테이블을 처리하려면 다음 예를 참조하세요.

AWS Management Console

1. 에 로그인합니다..[AWS Management Console](#)
2. 탐색 창에서 사용자 지정 어휘를 선택합니다. 그러면 기존 어휘를 확인하거나 새 어휘를 만들 수 있는 사용자 지정 어휘 페이지가 열립니다.
3. 어휘 생성을 선택합니다.



그러면 어휘 생성 페이지로 이동합니다. 새 사용자 지정 어휘의 이름을 입력합니다.

여기에는 다음과 같은 3가지 옵션이 있습니다.

- a. 컴퓨터에서 txt 또는 csv 파일을 업로드합니다.

사용자 지정 어휘를 새로 만들거나 시작하는 데 도움이 되는 템플릿을 다운로드할 수 있습니다. 그러면 어휘 보기 및 편집 창에 어휘가 자동으로 채워집니다.

Create vocabulary [Info](#)

Vocabulary settings

Name

Vocabulary names can be up to 200 characters in length. Allowed characters: a-z, A-Z, 0-9, periods (.), dashes (-), and underscores (_).

Language

Create and import vocabulary [Info](#)

Vocabulary input source

☒ **File upload**
 Upload a vocabulary table from your computer.

☐ **S3 location**
 Import a vocabulary table from an S3 location.

☐ **Create vocabulary on console**
 Manually create a vocabulary table on the console.

Download vocabulary template – *Optional*
 Download and complete a custom vocabulary template in your preferred format.

Download template ▼

Import from file

Choose File

File format: txt, csv, maximum size 50 KB.

- b. Amazon S3 위치에서 txt 또는 csv 파일을 가져옵니다.

사용자 지정 어휘를 새로 만들거나 시작하는 데 도움이 되는 템플릿을 다운로드할 수 있습니다. 완성된 어휘 파일을 Amazon S3 버킷에 업로드하고 요청에 해당 URI를 지정합니다. 그러면 어휘 보기 및 편집 창에 어휘가 자동으로 채워집니다.

Create and import vocabulary [Info](#)

Vocabulary input source

☐ File upload
Upload a vocabulary table from your computer.

☒ S3 location
Import a vocabulary table from an S3 location.

☐ Create vocabulary on console
Manually create a vocabulary table on the console.

Download vocabulary template – Optional
Download and complete a custom vocabulary template in your preferred format.

 **Download template** ▼

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

Resource URI

[View](#) [Browse S3](#)

c. 콘솔에서 어휘를 수동으로 생성합니다.

어휘 보기 및 편집 창으로 스크롤하여 10개 행 추가를 선택합니다. 이제 용어를 수동으로 입력할 수 있습니다.

Create and import vocabulary [Info](#)

Vocabulary input source

☐ File upload
Upload a vocabulary table from your computer.

☐ S3 location
Import a vocabulary table from an S3 location.

☒ Create vocabulary on console
Manually create a vocabulary table on the console.

View and edit vocabulary (0) [Reset vocabulary](#) [Delete](#) [Download latest vocabulary](#) ▼

[Show all](#) ▼ < 1 >

☐	Phrase	SoundsLike (optional)	IPA (optional)	DisplayAs (optional)
No rows added yet				
Add 10 rows				

4. 어휘 보기 및 편집 창에서 어휘를 편집할 수 있습니다. 변경하려면 수정할 항목을 클릭합니다.

View and edit vocabulary - new (10) [Info](#) Reset vocabulary Delete Download latest ▼

Show all ▼ < 1 >

☐	Phrase ↗	SoundsLike - optional ↗	IPA - optional ↗	DisplayAs - optional ↗
☐	Amazon-E.-C.-two	-	-	Amazon EC2
☐	Amazon-S.-three	-	-	Amazon S3
☐	Amazon-elasticashe	-	-	Amazon ElastiCache
☐	Amazon-sagemaker	-	-	Amazon SageMaker
☐	A.-W.-S.-iam	-	-	AWS IAM
☐	A.-W.-S.-I.-o.-T.	-	-	AWS IoT
☐	A.-W.-S.-W.-A.-F.	-	-	AWS WAF
☐	c.-plus-plus	-	-	C++
☐	nice-d.-c.-v.	-	-	NICE DCV
☐	w.-w.-w.-dot-amazon-dot-com	-	-	www.amazon.com

Add row

오류가 발생하면 자세한 오류 메시지가 표시되므로 어휘를 처리하기 전에 문제를 수정할 수 있습니다. 단, 어휘 생성을 선택하기 전에 모든 오류를 수정하지 않으면 어휘 요청이 실패한다는 점에 유의하세요.

View and edit vocabulary - new (4) [Info](#) Reset vocabulary Delete Download latest ▼

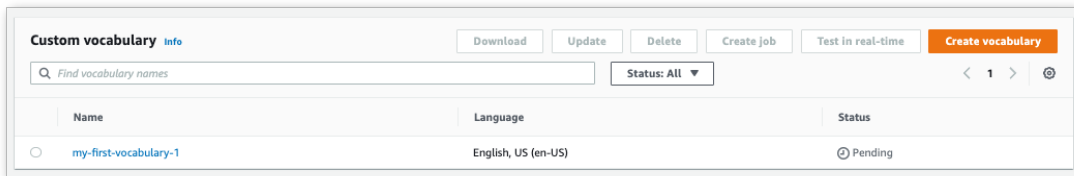
Show all ▼ < 1 >

☐	Phrase ↗	SoundsLike - optional ↗	IPA - optional ↗	DisplayAs - optional ↗
☐	Amazon-E.-C. two × ✓ ⚠ Phrase contains unsupported characters (" "). Phrase contains a formatting error.	-	-	Amazon EC2
☐	Amazon-S.-three	-	-	Amazon S3
☐	c.-plus-plus	-	-	C++
☐	w.-w.-w.-dot-amazon-dot-com	-	-	www.amazon.com

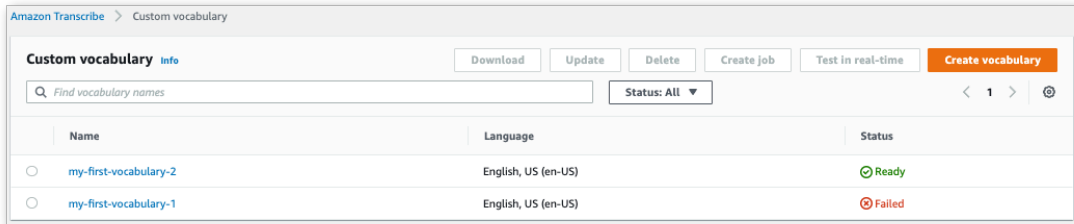
Add row

변경 내용을 저장하려면 체크 표시(✓)를 선택하고 변경 내용을 삭제하려면 'X'를 선택합니다.

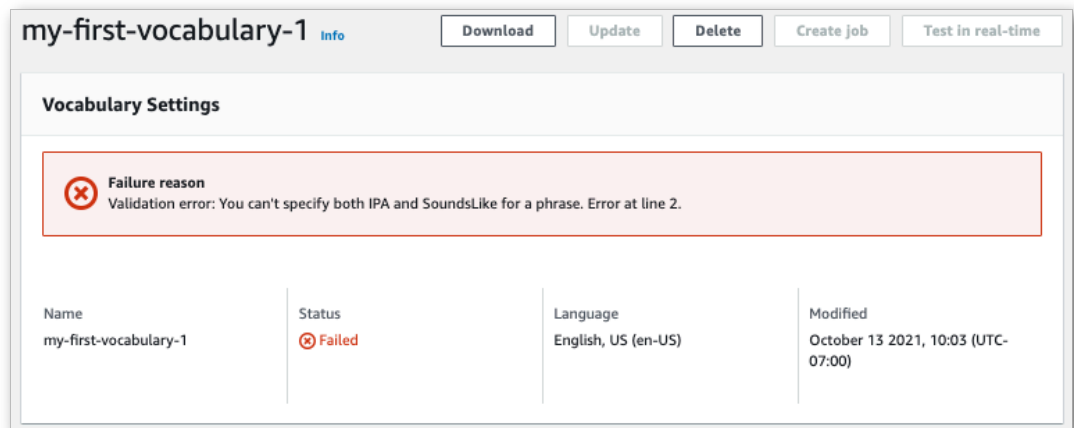
- 필요에 따라 사용자 지정 어휘에 태그를 추가합니다. 모든 필드를 작성하고 어휘에 만족하면 페이지 하단에서 어휘 생성을 선택합니다. 그러면 사용자 지정 어휘 페이지로 돌아가서 사용자 지정 어휘의 상태를 볼 수 있습니다. 상태가 '보류 중'에서 '준비'로 변경되면 사용자 지정 어휘를 트랜스크립션과 함께 사용할 수 있습니다.



6. 상태가 '실패'로 변경되면 사용자 지정 어휘의 이름을 선택하여 해당 정보 페이지로 이동합니다.



이 페이지 상단에는 사용자 지정 어휘가 실패한 이유에 대한 정보를 제공하는 실패 사유 배너가 있습니다. 텍스트 파일에서 오류를 수정하고 다시 시도하세요.



AWS CLI

이 예시에서는 테이블 형식의 어휘 파일과 함께 [create-vocabulary](#) 명령을 사용합니다. [자세한 내용은 CreateVocabulary 단원을 참조하세요.](#)

트랜스크립션 작업에서 기존 사용자 지정 어휘를 사용하려면 [StartTranscriptionJob](#) 작업을 호출할 때 [Settings](#) 필드에 VocabularyName을 설정하거나 AWS Management Console에서 드롭다운 목록에 있는 사용자 지정 어휘를 선택합니다.

```
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
```

다음은 [create-vocabulary](#) 명령을 사용하는 또 다른 예 및 사용자 지정 어휘를 생성하는 요청 본문입니다.

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

my-first-vocab-table.json 파일에는 다음과 같은 요청 본문이 포함되어 있습니다.

```
{
  "VocabularyName": "my-first-vocabulary",
  "VocabularyFileUri": "s3://amzn-s3-demo-bucket/my-vocabularies/my-vocabulary-table.txt",
  "LanguageCode": "en-US"
}
```

VocabularyState가 PENDING에서 READY로 변경되면 사용자 지정 어휘를 트랜스크립션과 함께 사용할 수 있습니다. 사용자 지정 어휘의 현재 상태를 보려면 다음을 실행합니다.

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

AWS SDK for Python (Boto3)

이 예시에서는 AWS SDK for Python (Boto3)을 사용하여 [create_vocabulary](#) 메서드를 사용하는 테이블에서 사용자 지정 어휘를 만듭니다. [자세한 내용은 CreateVocabulary 단원을 참조하세요.](#)

트랜스크립션 작업에서 기존 사용자 지정 어휘를 사용하려면 [StartTranscriptionJob](#) 작업을 호출할 때 [Settings](#) 필드에 VocabularyName을 설정하거나 AWS Management Console에서 드롭다운 목록에 있는 사용자 지정 어휘를 선택합니다.

기능별, 시나리오 및 교차 서비스 예를 포함하여 AWS SDK를 사용하는 추가 예시는 [AWS SDKs를 사용한 Amazon Transcribe의 코드 예제](#) 장을 참조하세요.

```
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

사용자 지정 어휘 파일을 위한 새 Amazon S3 버킷을 만드는 경우 [CreateVocabulary](#) 요청을 하는 IAM 역할에 이 버킷에 액세스할 수 있는 권한이 있는지 확인합니다. 역할에 올바른 권한이 없는 경우 요청이 실패합니다. 필요에 따라 `DataAccessRoleArn` 파라미터를 포함하여 요청 내에서 IAM 역할을 지정할 수 있습니다. Amazon Transcribe의 IAM 역할 및 정책에 대한 자세한 내용은 [Amazon Transcribe 자격 증명 기반 정책 예제](#)를 참조하세요.

목록을 사용하여 사용자 지정 어휘 생성

Important

목록 형식의 사용자 지정 어휘는 더 이상 사용되지 않으므로 새 사용자 지정 어휘를 생성하는 경우 [테이블 형식](#)을 사용하는 것이 좋습니다.

AWS Management Console, AWS CLI 또는 AWS SDK를 사용하여 목록에서 사용자 지정 어휘를 만들 수 있습니다.

- AWS Management Console: 사용자 지정 어휘가 포함된 텍스트 파일을 생성하고 업로드해야 합니다. 줄로 구분하거나 쉼표로 구분한 항목을 사용할 수 있습니다. 목록은 LF 형식의 텍스트(*.txt) 파일로 저장해야 한다는 점에 유의하세요. CRLF 등의 다른 형식을 사용하는 경우에는 사용자 지정 어휘는 Amazon Transcribe에서 허용되지 않습니다.

- AWS CLI 및 AWS SDK: [Phrases](#) 플래그를 사용하여 사용자 지정 어휘를 API 호출 내의 심표로 구분된 항목으로 포함해야 합니다.

항목에 여러 단어가 포함된 경우 각 단어에 하이픈을 넣어야 합니다. 예를 들어 'Los Angeles'는 **Los-Angeles**로 포함하고 'Andorra la Vella'는 **Andorra-la-Vella**로 포함합니다.

다음은 두 가지 유효한 목록 형식의 예입니다. 메서드별 예시는 [사용자 지정 어휘 목록 생성](#)를 참조하세요.

- 심표로 구분된 항목:

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

- 줄로 구분된 항목:

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

Important

해당 언어에 지원되는 문자만 사용할 수 있습니다. 자세한 내용은 해당 언어의 [문자 집합](#)을 참조하세요.

사용자 지정 어휘 목록은 [CreateMedicalVocabulary](#) 작업에서 지원되지 않습니다. 사용자 지정 의료 어휘를 생성하는 경우 테이블 형식을 사용해야 합니다. 지침은 [테이블을 사용하여 사용자 지정 어휘 생성](#)을 참조하세요.

사용자 지정 어휘 목록 생성

Amazon Transcribe에서 사용할 사용자 지정 어휘 목록을 처리하려면 다음 예를 참조하세요.

AWS CLI

이 예시에서는 목록 형식의 사용자 지정 어휘 파일과 함께 [create-vocabulary](#) 명령을 사용합니다. [자세한 내용은 CreateVocabulary 단원을 참조하세요.](#)

```
aws transcribe create-vocabulary \
--vocabulary-name my-first-vocabulary \
--language-code en-US \
--phrases {CLI,Eva-Maria,ABCs}
```

다음은 [create-vocabulary](#) 명령을 사용하는 또 다른 예 및 사용자 지정 어휘를 생성하는 요청 본문입니다.

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

my-first-vocab-list.json 파일에는 다음과 같은 요청 본문이 포함되어 있습니다.

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

VocabularyState가 PENDING에서 READY로 변경되면 사용자 지정 어휘를 트랜스크립션과 함께 사용할 수 있습니다. 사용자 지정 어휘의 현재 상태를 보려면 다음을 실행합니다.

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

AWS SDK for Python (Boto3)

이 예시에서는 AWS SDK for Python (Boto3)을 사용하여 [create_vocabulary](#) 메서드를 사용하여 목록에서 사용자 지정 어휘를 만듭니다. [자세한 내용은 CreateVocabulary 단원을 참조하세요.](#)

기능별, 시나리오 및 교차 서비스 예를 포함하여 AWS SDK를 사용하는 추가 예시는 [AWS SDKs를 사용한 Amazon Transcribe의 코드 예제](#) 장을 참조하세요.

```
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

사용자 지정 어휘 파일을 위한 새 Amazon S3 버킷을 만드는 경우 [CreateVocabulary](#) 요청을 하는 IAM 역할에 이 버킷에 액세스할 수 있는 권한이 있는지 확인합니다. 역할에 올바른 권한이 없는 경우 요청이 실패합니다. 필요에 따라 `DataAccessRoleArn` 파라미터를 포함하여 요청 내에서 IAM 역할을 지정할 수 있습니다. Amazon Transcribe의 IAM 역할 및 정책에 대한 자세한 내용은 [Amazon Transcribe 자격 증명 기반 정책 예제](#)를 참조하세요.

사용자 지정 어휘 사용

사용자 지정 어휘가 생성되면 트랜스크립션 요청에 포함시킬 수 있습니다. 예를 보려면 다음 섹션을 참조하세요.

요청에 포함하려는 사용자 지정 어휘의 언어는 미디어에 지정한 언어 코드와 일치해야 합니다. 언어가 일치하지 않으면 사용자 지정 어휘가 트랜스크립션에 적용되지 않으며 경고나 오류도 없습니다.

배치 트랜스크립션에서의 사용자 지정 어휘 사용

배치 트랜스크립션과 함께 사용자 지정 어휘를 사용하려면 다음 예를 참조하세요.

AWS Management Console

1. 에 로그인합니다.. [AWS Management Console](#)

2. 탐색 창에서 트랜스크립션 작업을 선택한 다음 작업 생성(오른쪽 상단)을 선택합니다. 그러면 작업 세부 정보 지정 페이지가 열립니다.

Specify job details
[Info](#)

Job settings

Name

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

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

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

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

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

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

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

Language
Choose the language of the input audio.

▶ **Additional settings**

작업 이름을 지정하고 입력 미디어를 지정합니다. 필요에 따라 다른 필드를 모두 포함하고 다음을 선택합니다.

3. 작업 구성 페이지 하단의 사용자 지정 패널에서 사용자 지정 어휘를 컵니다.

Configure job - optional [Info](#)

Audio settings

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

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

Content removal

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

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

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

Customization

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

Vocabulary selection
The vocabularies shown here are based on your language settings. You can choose up to one vocabulary per language. You can also [create a new vocabulary](#). [↗](#)

Choose a vocabulary ▼

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

4. 드롭다운 메뉴에서 사용자 지정 어휘를 선택합니다.

작업 생성을 선택하여 트랜스크립션 작업을 실행합니다.

AWS CLI

이 예시에서는 [start-transcription-job](#) 명령 및 Settings 파라미터를 VocabularyName 하위 파라미터와 함께 사용합니다. 자세한 내용은 [StartTranscriptionJob](#) 및 [Settings](#)을(를) 참조하세요.

```
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
```

다음은 [start-transcription-job](#) 명령을 사용하는 또 다른 예 및 해당 작업과 함께 사용자 지정 어휘를 포함하는 요청 본문입니다.

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

my-first-vocabulary-job.json 파일에는 다음과 같은 요청 본문이 포함되어 있습니다.

```
{
  "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 for Python (Boto3)

이 예시에서는 AWS SDK for Python (Boto3)을 사용하여 [start_transcription_job](#) 메서드의 `Settings` 인수를 사용하여 사용자 지정 어휘를 포함합니다. 자세한 내용은 [StartTranscriptionJob](#) 및 [Settings](#)을(를) 참조하세요.

기능별, 시나리오 및 교차 서비스 예를 포함하여 AWS SDK를 사용하는 추가 예시는 [AWS SDKs를 사용한 Amazon Transcribe의 코드 예제](#) 장을 참조하세요.

```
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'
    }
)

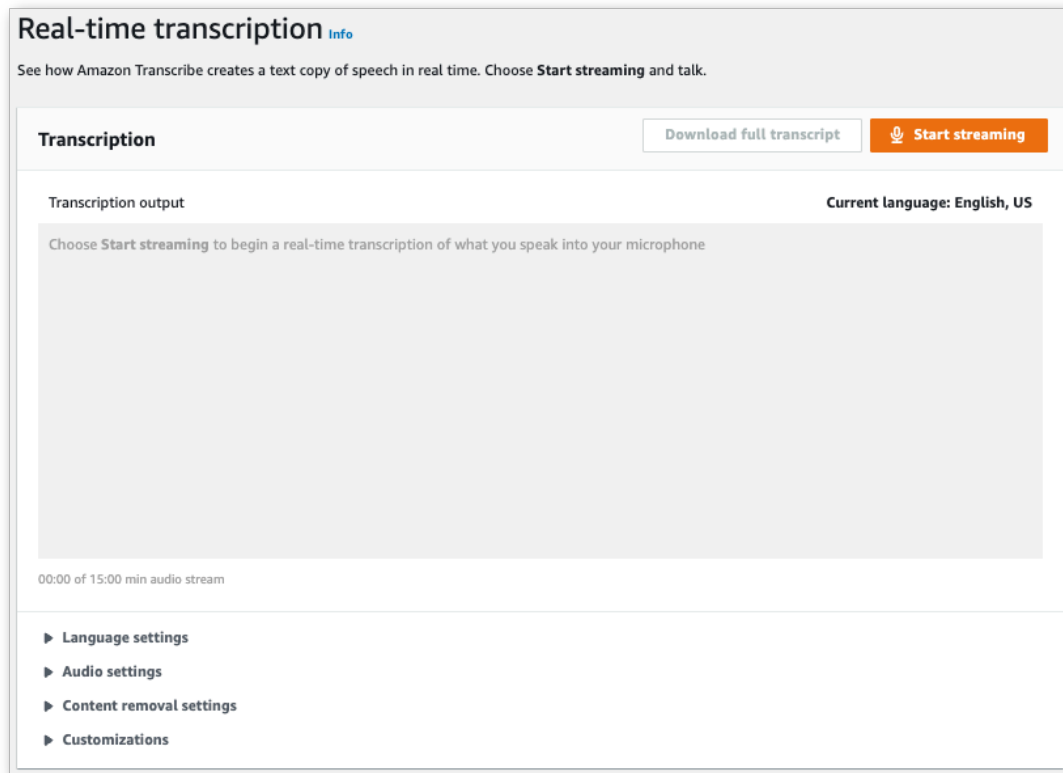
while True:
    status = transcribe.get_transcription_job(TranscriptionJobName = job_name)
    if status['TranscriptionJob']['TranscriptionJobStatus'] in ['COMPLETED', 'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
print(status)
```

스트리밍 트랜스크립션에서의 사용자 지정 어휘 사용

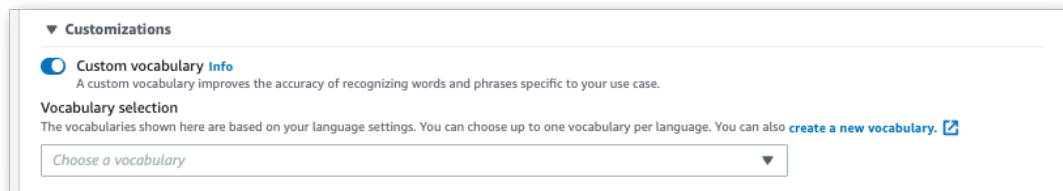
스트리밍 트랜스크립션과 함께 사용자 지정 어휘를 사용하려면 다음 예를 참조하세요.

AWS Management Console

1. 에 로그인합니다..[AWS Management Console](#)
2. 탐색 창에서 실시간 트랜스크립션을 선택합니다. 사용자 지정까지 아래로 스크롤하고 최소화된 경우 이 필드를 확장합니다.



3. 사용자 지정 어휘를 켜고 드롭다운 메뉴에서 사용자 지정 어휘를 선택합니다.



스트림에 적용하려는 기타 설정을 포함합니다.

4. 이제 스트림을 트랜스크립션할 준비가 되었습니다. 스트리밍 시작을 선택하고 말하기 시작합니다. 구술을 끝내려면 스트리밍 중지를 선택합니다.

HTTP/2 스트림

이 예시에서는 사용자 지정 어휘가 포함된 HTTP/2 요청을 생성합니다. Amazon Transcribe로 HTTP/2 스트리밍을 사용하는 방법에 대한 자세한 내용은 [HTTP/2 스트림 설정](#)을 참조하세요. Amazon Transcribe 관련 파라미터 및 헤더에 대한 자세한 내용은 [StartStreamTranscription](#)을 참조하세요.

```
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

```

파라미터 정의는 [API 참조](#)에서 찾을 수 있으며, 모든 AWS API 작업에 공통적인 파라미터는 [공통 파라미터](#) 섹션에 나열되어 있습니다.

WebSocket 스트림

이 예시에서는 사용자 지정 어휘를 WebSocket 스트림에 적용하는 미리 서명된 URL을 만듭니다. 가독성을 높이기 위해 줄바꿈이 추가되었습니다. Amazon Transcribe로 WebSocket 스트림을 사용하는 방법에 대한 자세한 내용은 [WebSocket 스트림 설정](#)을 참조하세요. 파라미터에 대한 자세한 내용은 [StartStreamTranscription](#)을 참조하세요.

```

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

```

파라미터 정의는 [API 참조](#)에서 찾을 수 있으며, 모든 AWS API 작업에 공통적인 파라미터는 [공통 파라미터](#) 섹션에 나열되어 있습니다.

사용자 지정 언어 모델

사용자 지정 언어 모델은 도메인별 음성의 트랜스크립션 정확도를 개선하도록 설계되었습니다. 여기에는 일상적인 대화에서 들을 수 있는 내용 이외의 모든 콘텐츠가 포함됩니다. 예를 들어 과학 컨퍼런스의 자료집을 트랜스크립션하는 경우 표준 트랜스크립션으로는 발표자가 사용하는 과학 용어 중 상당수를 인식하지 못할 수 있습니다. 이 경우 해당 분야에서 사용되는 특수 용어를 인식하도록 사용자 지정 언어 모델을 훈련할 수 있습니다.

힌트(예: 발음)를 제공하여 단어 인식을 높이는 사용자 지정 어휘와 달리 사용자 지정 언어 모델은 주어진 단어와 관련된 컨텍스트를 학습합니다. 여기에는 단어를 사용하는 방식과 시기, 단어와 다른 단어의 관계가 포함됩니다. 예를 들어, 기후 과학 연구 논문을 사용하여 모델을 훈련시키면 모델은 'ice flow'보다 'ice floe'가 단어 페어일 가능성이 더 높다는 것을 학습합니다.

사용자 지정 언어 모델에 지원되는 언어를 보려면 [지원되는 언어 및 언어별 기능](#)을 참조하세요. 단, 요청에 사용자 지정 언어 모델을 포함하는 경우 언어 식별을 활성화할 수 없습니다(언어 코드를 지정해야 함).

사용자 지정 언어 모델 관련 API 작업

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

데이터 소스

모델을 훈련하 데 필요한 모든 유형의 텍스트 데이터를 사용할 수 있습니다. 그러나 텍스트 콘텐츠가 오디오 콘텐츠에 가까울수록 모델의 정확도가 높아집니다. 따라서 오디오와 같은 컨텍스트에서 같은 용어를 사용하는 텍스트 데이터를 선택하는 것이 중요합니다.

모델 훈련에 가장 적합한 데이터는 정확한 트랜스크립트입니다. 이는 도메인 내 데이터로 간주됩니다. 도메인 내 텍스트 데이터는 트랜스크립션하려는 오디오와 정확히 동일한 용어, 사용법 및 컨텍스트를 가지고 있습니다.

정확한 트랜스크립트가 없는 경우 저널 기사, 기술 보고서, 백서, 회의 자료집, 지침 매뉴얼, 뉴스 기사, 웹 사이트 콘텐츠 및 오디오와 비슷한 컨텍스트에서 사용되는 원하는 용어가 포함된 기타 텍스트를 사용하세요. 이는 도메인 관련 데이터로 간주됩니다.

강력한 사용자 지정 언어 모델을 생성하려면 상당한 양의 텍스트 데이터가 필요할 수 있으며, 이 데이터에는 오디오에서 사용되는 용어가 포함되어야 합니다. 모델을 훈련하기 위해 최대 2GB의 텍스트 데

미터를 Amazon Transcribe에 제공할 수 있습니다. 이를 훈련 데이터라고 합니다. 선택적으로 도메인 내 트랜스크립트가 없거나(또는 적은 경우) 최대 200MB의 텍스트 데이터를 Amazon Transcribe에 제공하여 모델을 미세 조정할 수 있습니다. 이를 미세 조정 데이터라고 합니다.

훈련 데이터와 미세 조정 데이터 비교

훈련 데이터의 목적은 Amazon Transcribe가 새로운 용어를 인식하고 이러한 용어가 사용되는 컨텍스트를 학습하도록 가르치는 것입니다. 강력한 모델을 생성하려면 Amazon Transcribe에 많은 양의 관련 텍스트 데이터가 필요할 수 있습니다. 훈련 데이터를 최대한 많이(최대 2GB 제한) 제공하는 것이 좋습니다.

미세 조정 데이터의 목적은 훈련 데이터에서 학습한 컨텍스트 관계를 세부 조정하고 최적화하는 데 도움을 주는 것입니다. 사용자 지정 언어 모델을 생성하는 데 미세 조정 데이터가 필수는 아닙니다.

훈련 데이터 및 선택적으로 미세 조정 데이터를 선택하는 최선의 방법을 결정하는 것은 사용자의 몫입니다. 각 사례는 고유하며 보유한 데이터의 유형과 양에 따라 달라집니다. 도메인 내 훈련 데이터가 부족한 경우에는 미세 조정 데이터가 권장됩니다.

두 데이터 유형을 모두 포함하도록 선택하는 경우 훈련 데이터와 미세 조정 데이터가 겹치면 안 됩니다. 훈련 데이터와 미세 조정 데이터는 고유해야 합니다. 데이터가 겹치면 사용자 지정 언어 모델이 편향되고 왜곡되어 정확도에 영향을 미칠 수 있습니다.

일반적인 지침으로서, 가능하면 항상 정확한 도메인 내 텍스트를 훈련 데이터로 사용하는 것이 좋습니다. 다음은 몇 가지 일반적인 시나리오를 선호하는 순서대로 나열한 것입니다.

- 10,000개 단어 이상의 정확한 도메인 내 트랜스크립트 텍스트가 있는 경우 이를 훈련 데이터로 사용합니다. 이 경우 미세 조정 데이터를 포함할 필요가 없습니다. 사용자 지정 언어 모델을 훈련할 수 있는 이상적인 시나리오입니다.
- 10,000개 단어 미만의 정확한 도메인 내 트랜스크립트 텍스트가 있으며 원하는 결과를 얻지 못하는 경우 기술 보고서와 같은 도메인 관련 작성된 텍스트로 훈련 데이터를 보강하는 것을 고려해 보세요. 이 경우 도메인 내 트랜스크립트 데이터의 일부(10~25%)를 따로 남겨두어 미세 조정 데이터로 사용합니다.
- 도메인 내 트랜스크립트 텍스트가 없는 경우 모든 도메인 관련 텍스트를 훈련 데이터로 업로드합니다. 이 경우 작성된 텍스트보다 트랜스크립트 스타일의 텍스트를 사용하는 것이 좋습니다. 사용자 지정 언어 모델을 훈련할 수 있는 가장 덜 효과적인 시나리오입니다.

모델을 생성할 준비가 되었으면 [사용자 지정 언어 모델 생성](#)을 참조하세요.

사용자 지정 언어 모델 생성

사용자 지정 언어 모델을 만들려면 먼저 다음을 수행해야 합니다.

- 데이터를 준비합니다. 데이터는 일반 텍스트 형식으로 저장해야 하며 특수 문자를 포함할 수 없습니다.
- 데이터를 Amazon S3 버킷에 업로드합니다. 훈련 데이터 및 미세 조정 데이터용으로 별도의 폴더를 만드는 것이 좋습니다.
- Amazon Transcribe에 Amazon S3 버킷에 대한 액세스 권한이 있는지 확인합니다. 데이터를 사용하려면 액세스 권한이 있는 IAM 역할을 지정해야 합니다.

데이터 준비

모든 데이터를 하나의 파일로 컴파일하거나 여러 파일로 저장할 수 있습니다. 참고로 미세 조정 데이터를 포함하도록 선택한 경우, 미세 조정 데이터는 훈련 데이터와는 별도의 파일에 저장해야 합니다.

훈련 또는 미세 조정 데이터에 사용하는 텍스트 파일의 수는 중요하지 않습니다. 100,000개 단어가 포함된 파일 하나를 업로드하면 10,000개 단어가 포함된 파일 10개를 업로드하는 것과 같은 결과가 나옵니다. 가장 편리한 방법으로 텍스트 데이터를 준비합니다.

모든 데이터 파일이 다음 기준을 충족하는지 확인합니다.

- 모두 생성하려는 모델과 동일한 언어로 되어 있습니다. 예를 들어 오디오를 미국 영어(en-US)로 트랜스크립션하는 사용자 지정 언어 모델을 생성하려면 모든 텍스트 데이터가 미국 영어로 되어 있어야 합니다.
- UTF-8 인코딩을 사용하는 일반 텍스트 형식입니다.
- HTML 태그와 같은 특수 문자나 서식이 포함되어 있지 않습니다.
- 최대 총 크기는 훈련 데이터의 경우 2GB, 미세 조정 데이터의 경우 200MB입니다.

이러한 기준 중 하나라도 충족되지 않으면 모델이 작동하지 않습니다.

데이터 업로드

데이터를 업로드하기 전에 훈련 데이터를 위한 새 폴더를 만듭니다. 미세 조정 데이터를 사용하는 경우 별도의 폴더를 하나 더 생성합니다.

버킷의 URI는 다음과 같을 수 있습니다.

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

훈련 및 미세 조정 데이터를 적절한 버킷에 업로드합니다.

나중에 이러한 버킷에 더 많은 데이터를 추가할 수 있습니다. 하지만 그렇게 할 경우 새 데이터로 모델을 다시 생성해야 합니다. 기존 모델은 새 데이터로 업데이트할 수 없습니다.

데이터 액세스 허용

사용자 지정 언어 모델을 생성하려면 Amazon S3 버킷에 액세스할 권한이 있는 IAM 역할을 지정해야 합니다. 훈련 데이터를 배치한 Amazon S3 버킷에 액세스할 수 있는 역할이 아직 없는 경우에는 역할을 생성해야 합니다. 역할을 생성한 후 해당 역할에 권한을 부여하는 정책을 연결할 수 있습니다. 사용자에게 정책을 연결하지 않습니다.

예시 정책은 [Amazon Transcribe 자격 증명 기반 정책 예제](#) 섹션을 참조하세요.

새 IAM 자격 증명을 생성하는 방법을 알아보려면 [IAM 자격 증명\(사용자, 사용자 그룹 및 역할\)](#)을 참조하세요.

정책에 대한 자세한 정보는 다음을 참조하세요.

- [IAM의 정책 및 권한](#)
- [IAM 정책 생성](#)
- [AWS 리소스에 대한 액세스 관리](#)

사용자 지정 언어 모델 생성

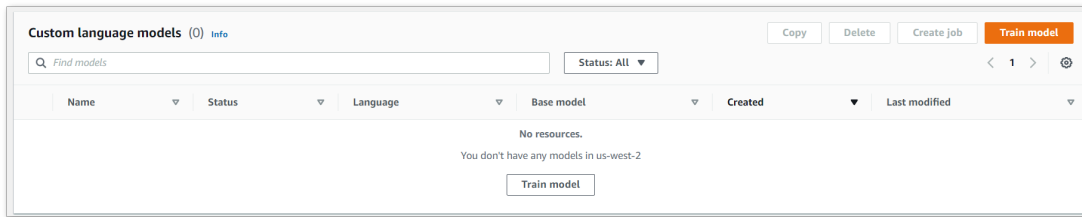
사용자 지정 언어 모델을 생성할 때는 기본 모델을 선택해야 합니다. 두 가지 기본 모델 옵션이 있습니다.

- **NarrowBand:** 샘플 속도가 16,000Hz 미만인 오디오에 이 옵션을 사용합니다. 이 모델 유형은 일반적으로 8,000Hz로 녹음된 전화 통화에 사용됩니다.
- **WideBand:** 샘플 속도가 16,000Hz 이상인 오디오에 이 옵션을 사용합니다.

AWS Management Console, AWS CLI 또는 AWS SDK를 사용하여 사용자 지정 언어 모델을 생성할 수 있습니다. 다음 예를 참조하세요.

AWS Management Console

1. 에 로그인합니다..[AWS Management Console](#)
2. 탐색 창에서 사용자 지정 언어 모델을 선택합니다. 그러면 기존 사용자 지정 언어 모델을 확인하거나 새 사용자 지정 언어 모델을 훈련할 수 있는 사용자 지정 언어 모델 페이지가 열립니다.
3. 새 모델을 훈련하려면 모델 훈련을 선택합니다.



모델 훈련 페이지로 이동됩니다. 이름을 추가하고 언어를 지정한 다음 모델에 사용할 기본 모델을 선택합니다. 그런 다음 훈련 데이터 경로를 추가하고 선택적으로 미세 조정 데이터 경로를 추가합니다. 데이터에 대한 액세스 권한이 있는 IAM 역할을 포함해야 합니다.

Train model [Info](#)

Model settings

Name

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

Language

Choose the language of your model.

Base model [Info](#)

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

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

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

Training data [Info](#)

Training data location on S3

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

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

Tuning data - optional [Info](#)

Tuning data location on S3

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

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

Access permissions

IAM role [Info](#)

☒ Use an existing IAM role

☐ Create an IAM role
By choosing Train model you are authorizing creation of this role.

Role name

A role that grants access to the S3 input locations.

4. 모든 필드를 작성했으면 페이지 하단에서 모델 훈련을 선택합니다.

AWS CLI

이 예시에서는 [create-language-model](#) 명령을 사용합니다. 자세한 내용은 [CreateLanguageModel1](#) 및 [LanguageModel](#)을(를) 참조하세요.

```
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
```

다음은 [create-language-model](#) 명령을 사용하는 또 다른 예 및 사용자 지정 언어 모델을 생성하는 요청 본문입니다.

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

my-first-language-model.json 파일에는 다음과 같은 요청 본문이 포함되어 있습니다.

```
{
  "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 for Python (Boto3)

[이 예시에서는 AWS SDK for Python \(Boto3\)을 사용하여 create_language_model 메서드를 사용하여 CLM을 생성합니다. 자세한 내용은 \[CreateLanguageModel\]\(#\) 및 \[LanguageModel\]\(#\)을\(를\) 참조하세요.](#)

기능별, 시나리오 및 교차 서비스 예를 포함하여 AWS SDK를 사용하는 추가 예시는 [AWS SDKs를 사용한 Amazon Transcribe의 코드 예제](#) 장을 참조하세요.

```
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)

```

사용자 지정 언어 모델 업데이트

Amazon Transcribe는 사용자 지정 언어 모델에 사용할 수 있는 기본 모델을 지속적으로 업데이트합니다. 이러한 업데이트를 활용하려면 6~12개월마다 새로운 사용자 지정 언어 모델을 훈련하는 것이 좋습니다.

사용자 지정 언어 모델이 최신 기본 모델을 사용하는지 확인하려면 AWS CLI 또는 AWS SDK를 사용하여 [DescribeLanguageModel](#) 요청을 실행한 다음 응답에서 UpgradeAvailability 필드를 찾습니다.

UpgradeAvailability이 true이면 모델이 기본 모델의 최신 버전을 실행하고 있지 않은 것입니다. 사용자 지정 언어 모델에서 최신 기본 모델을 사용하려면 새 사용자 지정 언어 모델을 생성해야 합니다. 사용자 지정 언어 모델은 업그레이드할 수 없습니다.

사용자 지정 언어 모델 사용

사용자 지정 어휘를 생성한 후에 트랜스크립션 요청에 포함시킬 수 있습니다. 예를 보려면 다음 섹션을 참조하세요.

요청에 포함하려는 모델의 언어는 미디어에 지정한 언어 코드와 일치해야 합니다. 언어가 일치하지 않으면 사용자 지정 언어 모델이 트랜스크립션에 적용되지 않으며 경고나 오류도 없습니다.

배치 트랜스크립션에서 사용자 지정 언어 모델 사용

배치 트랜스크립션과 함께 사용자 지정 언어 모델을 사용하려면 다음 예를 참조하세요.

AWS Management Console

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창에서 트랜스크립션 작업을 선택한 다음 작업 생성(오른쪽 상단)을 선택합니다. 그러면 작업 세부 정보 지정 페이지가 열립니다.
3. 작업 설정 패널의 모델 유형에서 사용자 지정 언어 모델 상자를 선택합니다.

The screenshot shows the 'Job settings' panel in the AWS Management Console. It includes a 'Name' field with the value 'MyTranscriptionJob' and a note about character limits. Below is the 'Model type' section with two radio buttons: 'General model' and 'Custom language model'. The 'Custom language model' option is selected. Underneath is the 'Language' dropdown menu set to 'English, US (en-US)'. At the bottom is the 'Custom model selection' dropdown menu with the text 'Choose model'. A link to 'create a new one' is also visible. A section for 'Additional settings' is partially visible at the bottom.

Job settings

Name

MyTranscriptionJob

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

Model type [Info](#)

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

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

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

Language

Choose the language of the input audio.

English, US (en-US)

Custom model selection

Choose an existing model or [create a new one](#).

Choose model

► Additional settings

또한 드롭다운 메뉴에서 입력 언어를 선택해야 합니다.

Job settings

Name

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

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

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

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

Language
 Choose the language of the input audio.

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

4. 사용자 지정 모델 선택 아래의 드롭다운 메뉴에서 기존 사용자 지정 언어 모델을 선택하거나 새 사용자 지정 언어 모델 생성을 선택합니다.

입력 데이터 패널에 입력 파일의 Amazon S3 위치를 추가합니다.

5. 추가 구성 옵션을 보려면 다음을 선택합니다.

작업 생성을 선택하여 트랜스크립션 작업을 실행합니다.

AWS CLI

이 예시에서는 [start-transcription-job](#) 명령 및 `ModelSettings` 파라미터를 `VocabularyName` 하위 파라미터와 함께 사용합니다. 자세한 내용은 [StartTranscriptionJob](#) 및 [ModelSettings](#) 섹션을 참조하세요.

```
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
```

다음은 [start-transcription-job](#) 명령을 사용하는 또 다른 예 및 해당 작업과 함께 사용자 지정 언어 모델을 포함하는 요청 본문입니다.

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

my-first-model-job.json 파일에는 다음과 같은 요청 본문이 포함되어 있습니다.

```
{
  "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 for Python (Boto3)

이 예제에서는 AWS SDK for Python (Boto3) 를 사용하여 [start_transcription_job](#) 메서드에 대한 ModelSettings 인수를 사용하여 사용자 지정 언어 모델을 포함합니다. 자세한 내용은 [StartTranscriptionJob](#) 및 [ModelSettings](#) 섹션을 참조하세요.

기능별, 시나리오 및 교차 서비스 예제를 포함하여 AWS SDKs를 사용하는 추가 예제는 [AWS SDKs를 사용한 Amazon Transcribe의 코드 예제](#)장을 참조하세요.

```
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'
}
)

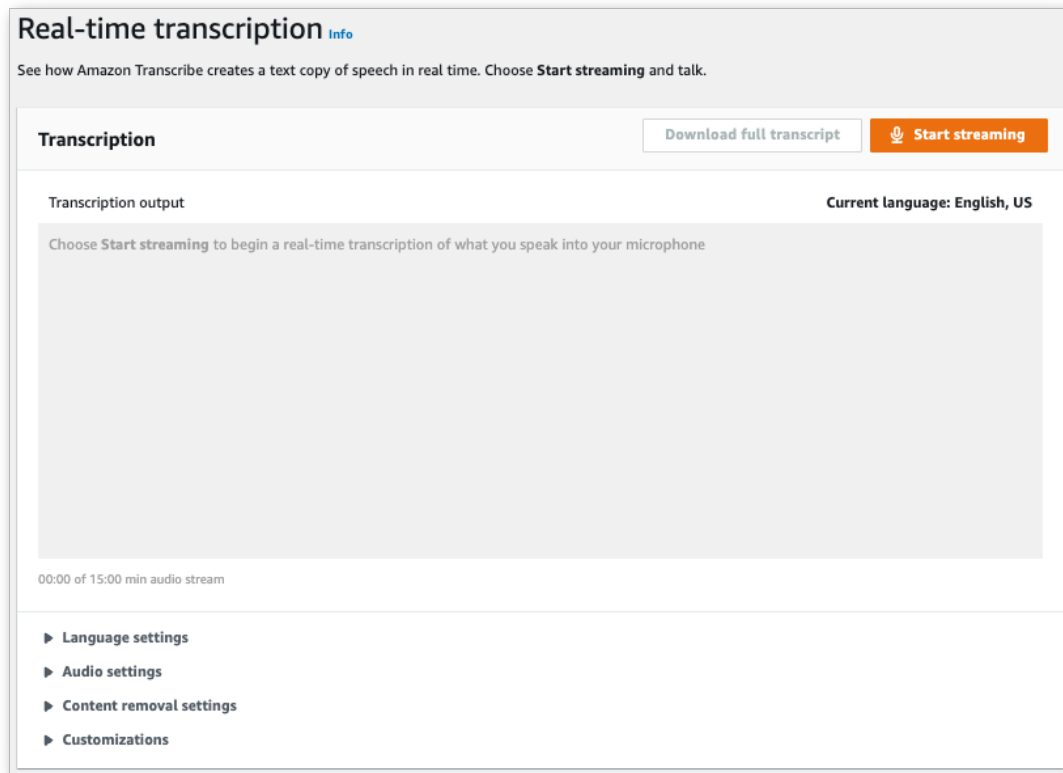
while True:
    status = transcribe.get_transcription_job(TranscriptionJobName = job_name)
    if status['TranscriptionJob']['TranscriptionJobStatus'] in ['COMPLETED', 'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
print(status)
```

스트리밍 트랜스크립션에서 사용자 지정 언어 모델 사용

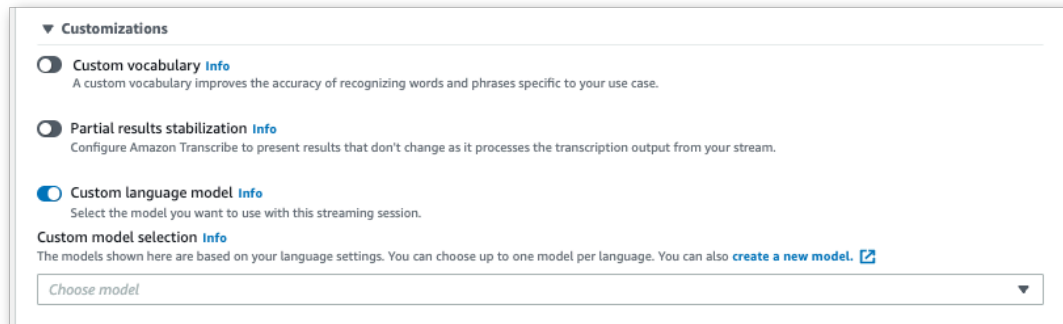
스트리밍 트랜스크립션과 함께 사용자 지정 언어 모델을 사용하려면 다음 예를 참조하세요.

AWS Management Console

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창에서 실시간 트랜스크립션을 선택합니다. 사용자 지정까지 아래로 스크롤하고 최소화된 경우 이 필드를 확장합니다.



3. 사용자 지정 언어 모델을 켜고 드롭다운 메뉴에서 모델을 선택합니다.



스트림에 적용할 기타 설정을 포함합니다.

4. 이제 스트림을 트랜스크립션할 준비가 되었습니다. 스트리밍 시작을 선택하고 말하기 시작합니다. 구술을 끝내려면 스트리밍 중지를 선택합니다.

HTTP/2 스트림

이 예시에서는 사용자 지정 언어 모델이 포함된 HTTP/2 요청을 생성합니다. 에서 HTTP/2 스트리밍을 사용하는 방법에 대한 자세한 내용은 섹션을 Amazon Transcribe참조하세요 [HTTP/2 스](#)

[트림 설정](#). 관련 파라미터 및 헤더에 대한 자세한 내용은 섹션을 Amazon Transcribe참조하세요 [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
```

파라미터 정의는 [API 참조](#)에서 찾을 수 있습니다. 모든 AWS API 작업에 공통적인 파라미터는 [공통 파라미터](#) 섹션에 나열되어 있습니다.

WebSocket 스트림

이 예시에서는 사용자 지정 언어 모델을 WebSocket 스트림에 적용하는 미리 서명된 URL을 만듭니다. 가독성을 높이기 위해 줄바꿈이 추가되었습니다. 에서 WebSocket 스트림을 사용하는 방법에 대한 자세한 내용은 섹션을 Amazon Transcribe참조하세요 [WebSocket 스트림 설정](#). 파라미터에 대한 자세한 내용은 [StartStreamTranscription](#)을 참조하세요.

```
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
```

파라미터 정의는 [API 참조](#)에서 찾을 수 있습니다. 모든 AWS API 작업에 공통적인 파라미터는 [공통 파라미터](#) 섹션에 나열되어 있습니다.

사용자 지정 어휘 필터를 사용하여 단어 삭제, 마스킹 또는 플래그 지정

사용자 지정 어휘 필터는 트랜스크립션 출력에서 수정할 개별 단어의 사용자 지정 목록이 포함된 텍스트 파일입니다.

일반적인 사용 사례는 모욕적이거나 비속한 용어를 제거하는 것이지만, 사용자 지정 어휘 필터는 전적으로 사용자가 지정하므로 원하는 단어를 선택할 수 있습니다. 예를 들어 새 제품이 곧 출시될 경우 회의 트랜스크립트에서 제품 이름을 숨길 수 있습니다. 이 경우 출시 전까지 제품 이름을 비밀로 유지하면서 이해 관계자에게 최신 정보를 알릴 수 있습니다.

어휘 필터링에는 mask, remove, tag의 세 가지 표시 방법이 있습니다. 각 예시의 작동 방식을 보려면 다음 예를 참조하세요.

- 마스킹: 지정된 단어를 별표 3개(***) 로 바꿉니다.

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

- 제거: 지정된 단어를 삭제하고 그 위치에 아무것도 남기지 않습니다.

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

- 태그: 지정된 각 단어에 태그("vocabularyFilterMatch": true)를 추가하지만 단어 자체는 변경하지 않습니다. 태그를 지정하면 트랜스크립트를 빠르게 대체하고 편집할 수 있습니다.

```
"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
```

트랜스크립션 요청을 제출할 때 사용자 지정 어휘 필터와 적용하려는 필터링 방법을 지정할 수 있습니다. 그러면 Amazon Transcribe는 지정한 필터링 방법에 따라 스크립트에 나타나는 정확한 단어 일치 항목을 수정합니다.

사용자 지정 어휘 필터는 배치 및 스트리밍 트랜스크립션 요청에 적용할 수 있습니다. 사용자 지정 어휘 필터를 생성하는 방법을 알아보려면 [어휘 필터 생성](#) 섹션을 참조하세요. 사용자 지정 어휘 필터를 적용하는 방법을 알아보려면 [사용자 지정 어휘 필터 사용](#) 섹션을 참조하세요.

Note

Amazon Transcribe는 인종적으로 민감한 용어를 자동으로 마스킹하지만 [AWS 기술 지원 부서](#)에 문의하여 이 기본 필터를 사용하지 않도록 설정할 수 있습니다.

어휘 필터링에 대한 비디오 시연은 [어휘 필터 사용](#)을 참조하세요.

어휘 필터링과 관련된 API 작업

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

어휘 필터 생성

사용자 지정 어휘 필터를 생성할 수 있는 두 가지 옵션이 있습니다.

1. 줄로 구분된 단어 목록을 UTF-8 인코딩의 일반 텍스트 파일로 저장합니다.
 - 이 접근 방식은 AWS Management Console, AWS CLI, 또는 AWS SDK에서 사용할 수 있습니다.
 - AWS Management Console을 사용하는 경우 사용자 지정 어휘 파일의 로컬 경로나 Amazon S3 URI를 제공할 수 있습니다.
 - AWS CLI 또는 AWS SDK를 사용하는 경우 사용자 지정 어휘 파일을 Amazon S3 버킷에 업로드하고 요청에 Amazon S3 URI를 포함해야 합니다.
2. API 요청에 쉼표로 구분된 단어 목록을 직접 포함합니다.
 - 이 접근 방식은 [Words](#) 파라미터를 통해 AWS CLI 또는 AWS SDK에서 사용할 수 있습니다.

각 방법의 예는 [사용자 지정 어휘 필터 생성](#) 섹션을 참조하세요.

사용자 지정 어휘 필터를 생성할 때 참고할 사항:

- 단어는 대/소문자를 구분하지 않습니다 예를 들어, "curse"와 "CURSE"는 동일한 단어로 취급합니다.
- 정확히 일치하는 단어만 필터링합니다. 예를 들어 필터에 "swear"가 포함되어 있지만 미디어에 "swears" 또는 "swearing"이라는 단어가 포함된 경우 이러한 단어는 필터링하지 않습니다. "swear" 인스턴스만 필터링합니다. 따라서 필터링하려는 단어의 모든 변형을 포함해야 합니다.
- 다른 단어에 포함된 단어에는 필터가 적용되지 않습니다. 예를 들어, 사용자 지정 어휘 필터에 "marine"이 포함되어지만 "submarine"은 포함되지 않은 경우 트랜스크립션에서 "submarine"이 변경되지 않습니다.
- 각 항목은 공백 없이 한 단어만 포함할 수 있습니다.
- 사용자 지정 어휘 필터를 텍스트 파일로 저장하는 경우 UTF-8 인코딩의 일반 텍스트 형식이어야 합니다.
- 사용자 지정 어휘 필터는 AWS 계정당 최대 100개까지 사용할 수 있으며 각 필터의 최대 용량은 50Kb일 수 있습니다.
- 해당 언어에 지원되는 문자만 사용할 수 있습니다. 자세한 내용은 해당 언어의 [문자 집합](#)을 참조하세요.

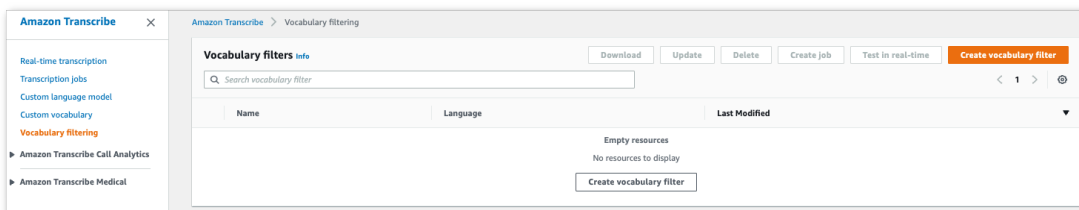
사용자 지정 어휘 필터 생성

Amazon Transcribe에서 사용할 사용자 지정 어휘 필터를 처리하려면 다음 예를 참조하세요.

AWS Management Console

계속하기 전에 사용자 지정 어휘 필터를 텍스트(*.txt) 파일로 저장하세요. 필요에 따라 파일을 Amazon S3 버킷에 업로드할 수 있습니다.

1. 에 로그인합니다..[AWS Management Console](#)
2. 탐색 창에서 어휘 필터링을 선택합니다. 그러면 기존 사용자 지정 어휘 필터를 확인하거나 새 필터를 생성할 수 있는 어휘 필터 페이지가 열립니다.
3. 어휘 필터 생성을 선택합니다.



그러면 어휘 필터 생성 페이지로 이동합니다. 새 사용자 지정 어휘 필터의 이름을 입력합니다.

어휘 입력 소스에서 파일 업로드 또는 S3 위치 옵션을 선택합니다. 그런 다음 사용자 지정 어휘 파일의 위치를 지정합니다.

4. 필요에 따라 사용자 지정 어휘 필터에 태그를 추가합니다. 모든 필드를 작성하면 페이지 하단에서 어휘 필터 생성을 선택합니다. 파일을 처리하는 중에 오류가 없는 경우 어휘 필터 페이지로 돌아갑니다.

이제 사용자 지정 어휘 필터를 사용할 준비가 되었습니다.

AWS CLI

이 예시에서는 [create-vocabulary-filter](#) 명령을 사용하여 단어 목록을 사용 가능한 사용자 지정 어휘 필터로 처리합니다. [자세한 내용은 CreateVocabularyFilter 단원을 참조하세요.](#)

옵션 1: words 파라미터를 사용하여 요청에 단어 목록을 포함할 수 있습니다.

```
aws transcribe create-vocabulary-filter \
```



```
--vocabulary-filter-name my-first-vocabulary-filter \
--language-code en-US \
--words profane,offensive,Amazon,Transcribe
```

옵션 2: 단어 목록을 텍스트 파일로 저장하고 Amazon S3 버킷에 업로드한 다음 `vocabulary-filter-file-uri` 파라미터를 사용하여 요청에 파일의 URI를 포함시킬 수 있습니다.

```
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
```

다음은 [create-vocabulary-filter](#) 명령을 사용하는 또 다른 예 및 사용자 지정 어휘 필터를 생성하는 요청 본문입니다.

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

`my-first-vocab-filter.json` 파일에는 다음과 같은 요청 본문이 포함되어 있습니다.

옵션 1: Words 파라미터를 사용하여 요청에 단어 목록을 포함할 수 있습니다.

```
{
  "VocabularyFilterName": "my-first-vocabulary-filter",
  "LanguageCode": "en-US",
  "Words": [
    "profane", "offensive", "Amazon", "Transcribe"
  ]
}
```

옵션 2: 단어 목록을 텍스트 파일로 저장하고 Amazon S3 버킷에 업로드한 다음 `VocabularyFilterFileUri` 파라미터를 사용하여 요청에 파일의 URI를 포함시킬 수 있습니다.

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

Note

요청에 VocabularyFilterFileUri를 포함하는 경우 Words를 사용할 수 없으며 둘 중 하나를 선택해야 합니다.

AWS SDK for Python (Boto3)

이 예시에서는 AWS SDK for Python (Boto3)를 통해 [create_vocabulary_filter](#) 메서드를 사용하여 사용자 지정 어휘 필터를 생성합니다. [자세한 내용은 CreateVocabularyFilter 단원을 참조하세요.](#)

기능별, 시나리오 및 교차 서비스 예를 포함하여 AWS SDK를 사용하는 추가 예시는 [AWS SDKs를 사용한 Amazon Transcribe의 코드 예제](#) 장을 참조하세요.

옵션 1: Words 파라미터를 사용하여 요청에 단어 목록을 포함할 수 있습니다.

```
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'
    ]
)
```

옵션 2: 단어 목록을 텍스트 파일로 저장하고 Amazon S3 버킷에 업로드한 다음 VocabularyFilterFileUri 파라미터를 사용하여 요청에 파일의 URI를 포함시킬 수 있습니다.

```
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

요청에 VocabularyFilterFileUri를 포함하는 경우 Words를 사용할 수 없으며 둘 중 하나를 선택해야 합니다.

Note

사용자 지정 어휘 필터 파일을 위한 새 Amazon S3 버킷을 만드는 경우 [CreateVocabularyFilter](#) 요청을 하는 IAM 역할에 이 버킷에 액세스할 수 있는 권한이 있는지 확인합니다. 역할에 올바른 권한이 없는 경우 요청이 실패합니다. 필요에 따라 DataAccessRoleArn 파라미터를 포함하여 요청 내에서 IAM 역할을 지정할 수 있습니다. Amazon Transcribe의 IAM 역할 및 정책에 대한 자세한 내용은 [Amazon Transcribe 자격 증명 기반 정책 예제](#)를 참조하세요.

사용자 지정 어휘 필터 사용

사용자 지정 어휘 필터가 생성되면 트랜스크립션 요청에 포함시킬 수 있습니다. 예를 보려면 다음 섹션을 참조하세요.

요청에 포함하려는 사용자 지정 어휘 필터의 언어는 미디어에 지정한 언어 코드와 일치해야 합니다. 언어 식별을 사용하고 여러 언어 옵션을 지정하는 경우 지정된 언어당 사용자 지정 어휘 필터 하나를 포함할 수 있습니다. 사용자 지정 어휘 필터의 언어가 오디오에서 식별된 언어와 일치하지 않으면 필터가 트랜스크립션에 적용되지 않으며 경고나 오류도 없습니다.

배치 트랜스크립션 작업에서의 어휘 필터 사용

배치 트랜스크립션과 함께 사용자 지정 어휘 필터를 사용하려면 다음 예를 참조하세요.

AWS Management Console

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창에서 트랜스크립션 작업을 선택한 다음 작업 생성(오른쪽 상단)을 선택합니다. 그러면 작업 세부 정보 지정 페이지가 열립니다.

Specify job details [Info](#)

Job settings

Name

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

Model type [Info](#)

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

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

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

Language settings

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

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

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

Language

Choose the language of the input audio.

► **Additional settings**

작업 이름을 지정하고 입력 미디어를 지정합니다. 필요에 따라 다른 필드를 모두 포함하고 다음을 선택합니다.

- 작업 구성 페이지의 콘텐츠 제거 패널에서 어휘 필터링을 설정합니다.

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

Audio settings

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

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

Content removal

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

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

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

Filter selection
The vocabulary filters shown here are based on your language settings. You can choose up to one vocabulary filter per language. You can also [create a new vocabulary filter](#). [↗](#)

Choose a vocabulary filter ▼

Customization

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

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

4. 드롭다운 메뉴에서 사용자 지정 어휘 필터를 선택하고 필터링 방법을 지정합니다.

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

Filter selection
The vocabulary filters shown here are based on your language settings. You can choose up to one vocabulary filter per language. You can also [create a new vocabulary filter](#).

my-vocab-filter ▼

Vocabulary filtering method
Use a vocabulary filter to filter vocabularies from your transcript. For example, in the sentence "The quick brown fox jumps over the lazy dog," you remove the word "lazy" using the following options.

- ☒ **Mask vocabulary**
Example: The quick brown fox jumps over the *** dog.
- ☐ **Remove vocabulary**
Example: The quick brown fox jumps over the dog.
- ☐ **Tag vocabulary**
Example: The quick brown fox jumps over the lazy dog.

5. 작업 생성을 선택하여 트랜스크립션 작업을 실행합니다.

AWS CLI

이 예시에서는 [start-transcription-job](#) 명령 및 Settings 파라미터를 VocabularyFilterName 및 VocabularyFilterMethod 하위 파라미터와 함께 사용합니다. 자세한 내용은 [StartTranscriptionJob](#) 및 [Settings](#)을(를) 참조하세요.

```
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
```

다음은 [start-transcription-job](#) 명령을 사용하는 또 다른 예 및 해당 작업과 함께 사용자 지정 어휘 필터를 포함하는 요청 본문입니다.

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

my-first-vocabulary-filter-job.json 파일에는 다음과 같은 요청 본문이 포함되어 있습니다.

```
{
  "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 for Python (Boto3)

이 예에서는 AWS SDK for Python (Boto3)를 통해 [start_transcription_job](#) 메서드의 `Settings` 인수를 사용하여 사용자 지정 어휘 필터를 포함합니다. 자세한 내용은 [StartTranscriptionJob](#) 및 [Settings](#)을(를) 참조하세요.

기능별, 시나리오 및 교차 서비스 예를 포함하여 AWS SDK를 사용하는 추가 예시는 [AWS SDKs를 사용한 Amazon Transcribe의 코드 예제](#) 장을 참조하세요.

```
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'
    }
)
```

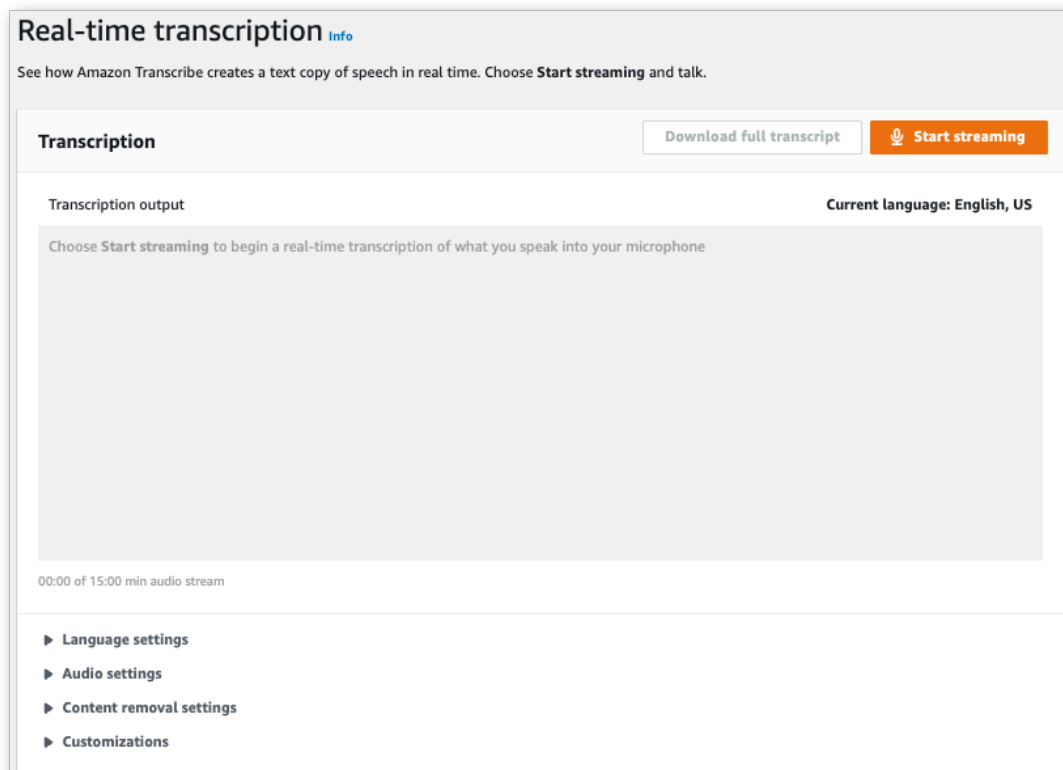
```
while True:
    status = transcribe.get_transcription_job(TranscriptionJobName = job_name)
    if status['TranscriptionJob']['TranscriptionJobStatus'] in ['COMPLETED', 'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
print(status)
```

스트리밍 트랜스크립션에서의 사용자 지정 어휘 필터 사용

스트리밍 트랜스크립션과 함께 사용자 지정 어휘 필터를 사용하려면 다음 예를 참조하세요.

AWS Management Console

1. 에 로그인합니다.. [AWS Management Console](#)
2. 탐색 창에서 실시간 트랜스크립션을 선택합니다. 콘텐츠 제거 설정까지 아래로 스크롤하고 최소화된 경우 이 필드를 확장합니다.



3. 어휘 필터링을 엽니다. 드롭다운 메뉴에서 사용자 지정 어휘 필터를 선택하고 필터링 방법을 지정합니다.

Content removal settings

Vocabulary filtering [Info](#)
Vocabulary filtering removes, masks, or tags words that you specify in your vocabulary filter. Choose a vocabulary filter to see an example.

Filter selection
The vocabulary filters shown here are based on your language settings. You can choose up to one vocabulary filter per language. You can also [create a new vocabulary filter](#).
[+](#)

my-vocab-filter ▼

Vocabulary filtering method [Info](#)
Use a vocabulary filter to filter vocabularies from your transcript. For example, in the sentence "The quick brown fox jumps over the lazy dog," you remove the word "lazy" using the following options.

- ☒ **Mask vocabulary**
Example: The quick brown fox jumps over the *** dog.
- ☐ **Remove vocabulary**
Example: The quick brown fox jumps over the dog.
- ☐ **Tag vocabulary**
Example: The quick brown fox jumps over the lazy dog.

스트림에 적용할 기타 설정을 포함합니다.

4. 이제 스트림을 트랜스크립션할 준비가 되었습니다. 스트리밍 시작을 선택하고 말하기 시작합니다. 구술을 끝내려면 스트리밍 중지를 선택합니다.

HTTP/2 스트림

이 예에서는 사용자 지정 어휘 필터와 필터 방법이 포함된 HTTP/2 요청을 생성합니다.

Amazon Transcribe로 HTTP/2 스트리밍을 사용하는 방법에 대한 자세한 내용은 [HTTP/2 스트림 설정](#)을 참조하세요. Amazon Transcribe 관련 파라미터 및 헤더에 대한 자세한 내용은 [StartStreamTranscription](#)을 참조하세요.

```
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
```

파라미터 정의는 [API 참조](#)에서 찾을 수 있으며, 모든 AWS API 작업에 공통적인 파라미터는 [공통 파라미터](#) 섹션에 나열되어 있습니다.

WebSocket 스트림

이 예에서는 사용자 지정 어휘 필터를 WebSocket 스트림에 적용하는 미리 서명된 URL을 만듭니다. 가독성을 높이기 위해 줄바꿈이 추가되었습니다. Amazon Transcribe로 WebSocket 스트림을 사용하는 방법에 대한 자세한 내용은 [WebSocket 스트림 설정](#)을 참조하세요. 파라미터에 대한 자세한 내용은 [StartStreamTranscription](#)을 참조하세요.

```
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
```

파라미터 정의는 [API 참조](#)에서 찾을 수 있으며, 모든 AWS API 작업에 공통적인 파라미터는 [공통 파라미터](#) 섹션에 나열되어 있습니다.

유해 언어 감지

유해 언어 감지는 P2P 대화가 필요한 소셜 미디어 플랫폼(예: 온라인 게임 플랫폼, 소셜 채팅 플랫폼)을 조정하는 데 도움이 되도록 고안되었습니다. 유해 언어의 사용은 개인, 동료 집단 및 커뮤니티에 심각한 해를 끼칠 수 있습니다. 유해한 언어에 플래그를 지정하면 조직에서 교양 있게 대화하고 사용자가 자유롭게 만들고 공유하고 참여할 수 있는 안전하고 포용적인 온라인 환경을 유지할 수 있습니다.

Amazon Transcribe 유해성 감지는 오디오 및 텍스트 기반 신호를 모두 활용하여 성희롱, 혐오 표현, 위협, 학대, 욕설, 모욕, 노골적인 표현 등 7가지 범주에 걸쳐 음성 기반 유해 콘텐츠를 식별하고 분류합니다. Amazon Transcribe 유해성 감지는 텍스트 외에도 어조 및 음높이와 같은 음성 신호를 사용하여 말의 유해한 의도를 찾아냅니다. 이는 의도를 고려하지 않고 특정 용어에만 집중하도록 설계된 표준 콘텐츠 조정 시스템을 개선한 것입니다.

Amazon Transcribe는 유해 언어에 플래그를 지정하고 분류하여 수동으로 처리해야 하는 데이터의 양을 최소화합니다. 이를 통해 콘텐츠 중재자는 플랫폼에서 담화를 빠르고 효율적으로 관리할 수 있습니다.

유해 언어 범주에는 다음이 포함됩니다.

- 욕설: 무례하거나 저속하거나 불쾌감을 주는 단어, 문구 또는 두문자어가 포함된 말.
- 혐오 표현: 정체성(예: 인종, 민족, 성별, 종교, 성적 취향, 능력, 출신 국가)을 근거로 개인이나 집단을 비판, 모욕, 비인간화하는 말.
- 성적 표현: 신체 부위, 신체적 특징 또는 성별을 직간접적으로 언급하여 성적 관심, 활동 또는 흥분을 나타내는 말.
- 모욕: 비하하거나, 굴욕하거나, 조롱하거나, 모욕하거나, 알보는 표현이 포함된 말. 이러한 유형의 표현은 괴롭힘(bullying)으로도 분류됩니다.
- 폭력 또는 위협: 개인이나 집단에 대해 고통, 부상 또는 적대감을 주려는 위협이 포함된 말.
- 노골적인 표현: 시각적으로 묘사적이고 불쾌할 정도로 생생한 이미지를 사용하는 말. 이러한 유형의 표현은 상대방의 불편함을 증폭시키기 위해 일부러 장황하게 말하는 경우가 많습니다.
- 괴롭힘(harassment) 또는 학대: 상대방을 비하하고 객관화하는 표현을 포함하여 상대방의 심리적 안녕에 영향을 미치려는 의도를 가진 말. 이러한 유형의 표현은 괴롭힘(harassment)으로도 분류됩니다.

유해성 감지는 음성 세그먼트(자연스러운 끊어읽기 사이의 음성)를 분석하고 이러한 세그먼트에 신뢰도 점수를 할당합니다. 신뢰도 점수는 0에서 1 사이의 값입니다. 신뢰도 점수가 높을수록 해당 콘텐츠

가 관련 범주에서 유해 언어일 가능성이 커집니다. 이러한 신뢰도 점수를 사용하여 사용 사례에 적합한 유해성 감지 임계값을 설정할 수 있습니다.

 Note

유해성 감지는 미국 영어(en-US)로 된 배치 트랜스크립션에서만 사용할 수 있습니다.

JSON 형식의 [예시 출력](#)을 볼 수 있습니다.

유해 언어 감지 사용

배치 트랜스크립션에 유해 언어 감지 기능 사용

배치 트랜스크립션에서 유해 언어 감지를 사용하려면 다음 예를 참조하십시오.

AWS Management Console

1. 에 로그인합니다..[AWS Management Console](#)
2. 탐색 창에서 트랜스크립션 작업을 선택한 다음 작업 생성(오른쪽 상단)을 선택합니다. 그러면 작업 세부 정보 지정 페이지가 열립니다.

Specify job details [Info](#)

Job settings

Name

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

Model type [Info](#)

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

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

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

Language settings

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

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

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

Language

Choose the language of the input audio.

► **Additional settings**

- 작업 세부 정보 지정 페이지에서 원하는 경우 PII 삭제를 활성화할 수도 있습니다. 나열된 다른 옵션은 유해성 감지에서 지원되지 않습니다. 다음을 선택합니다. 그러면 작업 구성 - 선택 사항 페이지로 이동합니다. 오디오 설정 패널에서 유해성 감지를 선택합니다.

Audio settings

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

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

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

Content removal

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

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

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

Customization

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

Cancel Previous **Create job**

- 작업 생성을 선택하여 트랜스크립션 작업을 실행합니다.
- 트랜스크립션 작업이 완료되면 트랜스크립션 작업 세부 정보 페이지의 다운로드 드롭다운 메뉴에서 트랜스크립션을 다운로드할 수 있습니다.

AWS CLI

이 예시에서는 [start-transcription-job](#) 명령 및 `ToxicityDetection` 파라미터를 사용합니다. 자세한 내용은 [StartTranscriptionJob](#) 및 [ToxicityDetection](#)을(를) 참조하세요.

```
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
```

다음은 [start-transcription-job](#) 명령을 사용하는 또 다른 예 및 유해성 감지를 포함하는 요청 본문입니다.

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

my-first-toxicity-job.json 파일에는 다음과 같은 요청 본문이 포함되어 있습니다.

```
{
  "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 for Python (Boto3)

이 예시에서는 AWS SDK for Python (Boto3)를 사용하여 [start_transcription_job](#) 메서드의 ToxicityDetection를 활성화합니다. 자세한 내용은 [StartTranscriptionJob](#) 및 [ToxicityDetection](#)을(를) 참조하세요.

기능별, 시나리오 및 교차 서비스 예를 포함하여 AWS SDK를 사용하는 추가 예시는 [AWS SDKs를 사용한 Amazon Transcribe의 코드 예제](#) 장을 참조하세요.

```

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)

```

출력 예시

트랜스크립션 출력에서 유해 언어를 태깅하고 분류합니다. 유해 언어의 각 사례를 분류하고 신뢰도 점수(0에서 1 사이의 값)를 할당합니다. 신뢰도 값이 높을수록 콘텐츠가 지정된 범주 내에서 유해 언어일 가능성이 커집니다.

출력 예(JSON)

다음은 분류된 유해 언어와 관련 신뢰도 점수를 보여주는 JSON 형식의 출력 예시입니다.

```
{
```



```

"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"  
}
```

개인 식별 정보 편집 또는 식별

수정은 개인 식별 정보(PII) 형식의 민감한 콘텐츠를 트랜스크립트에서 숨기거나 제거하는 데 사용됩니다. Amazon Transcribe가 편집할 수 있는 PII 유형은 배치 트랜스크립션과 스트리밍 트랜스크립션 간에 다릅니다. 각 트랜스크립션 메서드에 대한 PII 목록을 보려면 [배치 작업의 PII 편집](#) 및 [실시간 스트림에서 PII 편집 또는 식별](#)을 참조하세요. 스트리밍 트랜스크립션의 경우 PII를 편집하지 않고 플래그를 지정하는 옵션도 있습니다. 출력 예시를 보려면 [PII 식별 출력 예시](#)를 참조하세요.

수정이 활성화된 경우 편집된 트랜스크립트만 생성하거나 수정된 트랜스크립트와 수정되지 않은 트랜스크립트를 모두 생성할 수 있습니다. 편집된 트랜스크립트만 생성하도록 선택하는 경우 전체 대화가 저장되는 유일한 장소는 미디어뿐이라는 점에 유의하세요. 원본 미디어를 삭제하면 편집되지 않은 PII는 기록되지 않습니다. 따라서 편집된 트랜스크립트와 함께 편집되지 않은 트랜스크립트도 함께 생성하는 것이 현명할 수 있습니다.

배치 트랜스크립션을 통한 PII 수정에 대한 자세한 내용은 [배치 작업의 PII 편집](#)를 참조하세요.

스트리밍 트랜스크립션을 통한 PII 수정 또는 식별에 대한 자세한 내용은 [실시간 스트림에서 PII 편집 또는 식별](#)을 참조하세요.

Important

수정 기능은 민감한 데이터를 식별하고 제거하도록 설계되었습니다. 그러나 기계 학습의 예측 특성으로 인해 Amazon Transcribe는 트랜스크립트에 있는 민감한 데이터의 모든 인스턴스를 식별하여 제거하지 못할 수 있습니다. 편집된 출력을 검토하여 요구 사항에 맞는지 확인하는 것이 좋습니다.

수정 기능은 1996년 미국 건강 보험 양도 및 책임에 관한 법(HIPAA)과 같은 의료 개인 정보 보호법에 따른 비식별화 요구 사항을 충족하지 않습니다.

Amazon Transcribe의 수정 기능에 대한 비디오 시연은 [콘텐츠 수정을 사용하여 PII 식별 및 수정](#)을 참조하세요.

배치 작업의 PII 편집

배치 트랜스크립션 작업 중에 트랜스크립트에서 개인 식별 정보(PII)를 수정할 때는 식별된 각 PII 인스턴스를 트랜스크립트의 본문[PII]에 있는 로 바 Amazon Transcribe 끝니다. 트랜스크립션 출력의 단어별 부분에서 편집된 PII의 유형도 확인할 수 있습니다. 출력 샘플은 [편집된 출력\(배치\) 예시](#)를 참조하세요.

배치 트랜스크립션을 통한 수정은 미국 영어(en-US)와 미국 스페인어(es-US)로 제공됩니다. 수정은 [언어 식별](#)과 호환되지 않습니다.

수정된 트랜스크립트와 수정되지 않은 트랜스크립트는 모두 동일한 출력 Amazon S3 버킷에 저장됩니다.는 지정한 버킷 또는 서비스에서 관리하는 기본 Amazon S3 버킷에 트랜스크립트를 Amazon Transcribe 저장합니다.

배치 트랜스크립션에 대해 인식 Amazon Transcribe 할 수 있는 PII 유형

PII 유형	설명
ADDRESS	실제 주소(예: 100 Main Street, Anytown, USA 또는 Suite #12, Building 123). 주소에는 거리, 건물, 위치, 도시, 주, 국가, 카운티, 우편번호, 구역, 지역 등의 정보가 포함될 수 있습니다.
ALL	이 테이블에 나열된 모든 PII 유형을 편집하거나 식별합니다.
BANK_ACCOUNT_NUMBER	미국 은행 계좌 번호. 일반적으로 10~12자리 길이지만 Amazon Transcribe 는 마지막 4자리만 있으면 은행 계좌 번호도 인식합니다.
BANK_ROUTING	미국 은행 계좌 라우팅 번호. 일반적으로 9자리 길이지만 Amazon Transcribe 는 마지막 4자리만 있으면 라우팅 번호도 인식합니다.
CREDIT_DEBIT_CVV	비자, 마스터카드, 디스커버 신용카드 및 직불카드에 있는 3자리 카드 인증 코드(CVV). 아메리칸 익스프레스 신용카드나 직불카드의 경우 4자리 숫자 코드입니다.
CREDIT_DEBIT_EXPIRY	신용카드 또는 직불카드 만료 날짜. 이 숫자는 보통 4자리 숫자이며, 월/년 또는 MM/YY 형식으로 지정됩니다. 예를 들어 Amazon Transcribe 는 01/21, 01/2021, 2021년 1월과 같은 만료 날짜를 인식할 수 있습니다.
CREDIT_DEBIT_NUMBER	신용카드 또는 직불카드 번호. 이러한 번호는 13~16자리까지 다양할 수 있지만 마지막 4자

PII 유형	설명
	리만 있는 경우 신용 카드 또는 직불 카드 번호 Amazon Transcribe 도 인식합니다.
EMAIL	이메일 주소(예: efua.owusu@email.com).
NAME	개인의 이름. 이 엔터티 유형에는 Mr., Mrs., Miss 또는 Dr. Amazon Transcribe 와 같은 제목 이 포함되지 않습니다. 조직 또는 주소의 일부 인 이름에는이 엔터티 유형을 적용하지 않습니다. 예를 들어는 John Doe Organization을 조직 으로 Amazon Transcribe 인식하고 Jane Doe Street를 주소로 인식합니다.
PHONE	전화번호. 이 엔터티 유형에는 팩스 및 호출기 번호도 포함됩니다.
PIN	다른 사람이 자신의 은행 계좌 정보에 액세스할 수 있도록 하는 4자리 개인 식별 번호(PIN).
SSN	사회보장번호(SSN)는 미국 시민, 영주권자 및 임시 근로 거주자에게 발급되는 9자리 숫자입니다. Amazon Transcribe 는 마지막 4자리만 있는 경우에도 사회보장번호를 인식합니다.

AWS Management Console AWS CLI또는 AWS SDK를 사용하여 배치 트랜스크립션 작업을 시작할 수 있습니다.

AWS Management Console

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창에서 트랜스크립션 작업을 선택한 다음 작업 생성(오른쪽 상단)을 선택합니다. 그러면 작업 세부 정보 지정 페이지가 열립니다.
3. 작업 세부 정보 지정 페이지에서 원하는 필드를 채운 후 다음을 선택하여 작업 구성 - 선택 사항 페이지로 이동합니다. 여기에서 PII 수정 토글이 있는 콘텐츠 제거 패널을 찾을 수 있습니다.

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

Audio settings

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

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

Content removal

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

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

4. PII 수정을 선택하면 편집하려는 모든 PII 유형을 선택할 수 있습니다. 또한 작업 출력에 편집되지 않은 트랜스크립트 포함을 선택하면 편집되지 않은 트랜스크립트가 포함되도록 선택할 수 있습니다.

Content removal

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

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

☐ **Include unredacted transcript in job output**
Returns unredacted version of the transcript in addition to the redacted version.

Select PII entity types (11 of 11 selected)

☒ **Select All**

☒ **Financial (6 of 6 selected)**

☒ BANK_ACCOUNT_NUMBER	☒ BANK_ROUTING	☒ CREDIT_DEBIT_NUMBER
☒ CREDIT_DEBIT_CVV	☒ CREDIT_DEBIT_EXPIRY	☒ PIN

☒ **Personal (5 of 5 selected)**

☒ NAME	☒ ADDRESS	☒ PHONE
☒ EMAIL	☒ SSN	

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

5. 작업 생성을 선택하여 트랜스크립션 작업을 실행합니다.

AWS CLI

이 예시에서는 [start-transcription-job](#) 명령 및 content-redaction 파라미터를 사용합니다. 자세한 내용은 [StartTranscriptionJob](#) 및 [ContentRedaction](#) 섹션을 참조하세요.

```
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
```

다음은 [start-transcription-job](#) 메서드를 사용하는 또 다른 예 및 해당 작업에 대한 PII를 편집하는 요청 본문입니다.

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

my-first-redaction-job.json 파일에는 다음과 같은 요청 본문이 포함되어 있습니다.

```
{
  "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 for Python (Boto3)

이 예제에서는 AWS SDK for Python (Boto3) 를 사용하여 [start_transcription_job](#) 메서드에 대한 ContentRedaction 인수를 사용하여 콘텐츠를 수정합니다. 자세한 내용은 [StartTranscriptionJob](#) 및 [ContentRedaction](#) 섹션을 참조하세요.

기능별, 시나리오 및 교차 서비스 예제 AWS SDKs를 사용하는 추가 예제는 [AWS SDKs를 사용한 Amazon Transcribe의 코드 예제](#)장을 참조하세요.

```
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

배치 작업에 대한 PII 수정은 AWS 리전아시아 태평양(홍콩), 아시아 태평양(뭄바이), 아시아 태평양(서울), 아시아 태평양(싱가포르), 아시아 태평양(시드니), 아시아 태평양(도쿄), GovCloud(미국 서부), 캐나다(중부), EU(프랑크푸르트), EU(아일랜드), EU(런던), EU(파리), 중동(바레인), 남아메리카(상파울루), 미국 동부(버지니아 북부), 미국 동부(오하이오), 미국 서부(오레곤) 및 미국 서부(캘리포니아 북부)에서만 지원됩니다.

실시간 스트림에서 PII 편집 또는 식별

스트리밍 트랜스크립션에서 개인 식별 정보(PII)를 편집하는 경우 Amazon Transcribe 는 식별된 각 PII 인스턴스를 트랜스크립트의 [PII]로 대체합니다.

스트리밍 트랜스크립션에 사용할 수 있는 추가 옵션은 PII 식별입니다. PII 식별을 활성화하면 Entities 객체 아래에 있는 트랜스크립션 결과의 PII에 Amazon Transcribe 레이블을 지정합니다. 출력 샘플은 [편집된 스트리밍 출력 예시](#) 및 [PII 식별 출력 예시](#)를 참조하세요.

스트리밍 트랜스크립션을 사용하는 PII의 수정 및 식별은 지역 별로 영어 방언(스코틀랜드(en-AB), 호주(en-AU), 캐나다(en-CA), 영국(en-GB), 아일랜드(en-IE), 인도(en-IN), 뉴질랜드(en-NZ), 미국(en-US), 웨일스(en-WL) 및 남아프리카(en-ZA)), 스페인어 방언(미국(es-US), 스페인(es-ES), 프랑스어 방언: 프랑스(fr-FR), 캐나다(fr-CA)), 포르투갈어 방언(포르투갈(pt-PT), 브라질(pt-BR)), 이탈리아어 방언(이탈리아(it-IT)), 독일어 방언(독일(de-DE), 스위스(de-CH))로 제공됩니다.

스트리밍 작업에 대한 PII 식별 및 수정은 오디오 세그먼트의 완전한 트랜스크립션 시에만 수행됩니다.

스트리밍 트랜스크립션에 대해 인식 Amazon Transcribe 할 수 있는 PII 유형

PII 유형	설명
ADDRESS	실제 주소(예: 100 Main Street, Anytown, USA 또는 Suite #12, Building 123). 주소에는 거리, 건물, 위치, 도시, 주, 국가, 카운티, 우편번호, 구역, 지역 등의 정보가 포함될 수 있습니다.
ALL	이 테이블에 나열된 모든 PII 유형을 편집하거나 식별합니다.

PII 유형	설명
BANK_ACCOUNT_NUMBER	미국 은행 계좌 번호. 일반적으로 10~12자리 길이이지만 Amazon Transcribe 는 마지막 4자리만 있으면 은행 계좌 번호도 인식합니다.
BANK_ROUTING	미국 은행 계좌 라우팅 번호. 일반적으로 9자리 길이이지만 Amazon Transcribe 는 마지막 4자리만 있으면 라우팅 번호도 인식합니다.
CREDIT_DEBIT_CVV	비자, 마스터카드, 디스커버 신용카드 및 직불카드에 있는 3자리 카드 인증 코드(CVV). 아메리칸 익스프레스 신용카드나 직불카드의 경우 4자리 숫자 코드입니다.
CREDIT_DEBIT_EXPIRY	신용카드 또는 직불카드 만료 날짜. 이 숫자는 보통 4자리 숫자이며, 월/년 또는 MM/YY 형식으로 지정됩니다. 예를 들어 Amazon Transcribe 는 01/21, 01/2021, 2021년 1월과 같은 만료 날짜를 인식할 수 있습니다.
CREDIT_DEBIT_NUMBER	신용카드 또는 직불카드 번호. 이러한 번호는 13~16자리까지 다양할 수 있지만 마지막 4자리만 있는 경우 신용 카드 또는 직불 카드 번호 Amazon Transcribe 도 인식합니다.
EMAIL	이메일 주소(예: efua.owusu@email.com).
NAME	개인의 이름. 이 엔터티 유형에는 Mr., Mrs., Miss 또는 Dr. Amazon Transcribe 와 같은 제목이 포함되지 않습니다. 조직 또는 주소의 일부인 이름에는이 엔터티 유형을 적용하지 않습니다. 예를 들어 John Doe Organization을 조직으로 Amazon Transcribe 인식하고 Jane Doe Street를 주소로 인식합니다.
PHONE	전화번호. 이 엔터티 유형에는 팩스 및 호출기 번호도 포함됩니다.

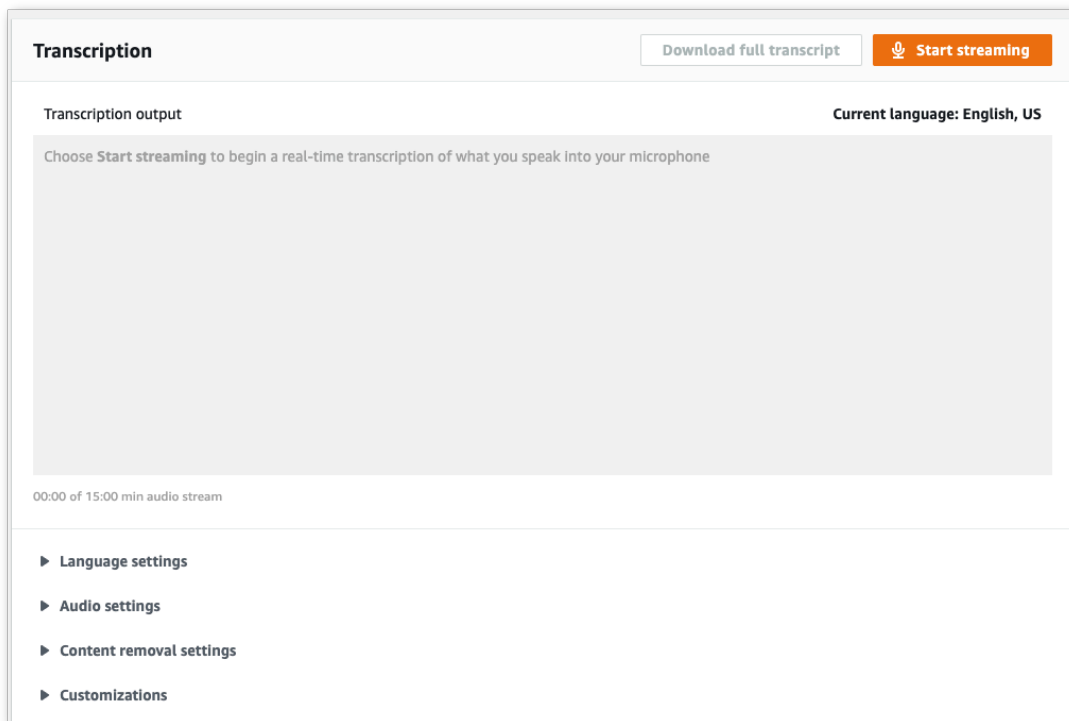
PII 유형	설명
PIN	다른 사람이 자신의 은행 계좌 정보에 액세스할 수 있도록 하는 4자리 개인 식별 번호(PIN).
SSN	사회보장번호(SSN)는 미국 시민, 영주권자 및 임시 근로 거주자에게 발급되는 9자리 숫자입니다. Amazon Transcribe 는 마지막 4자리만 있는 경우에도 사회보장번호를 인식합니다.
AGE	개인의 연령(수량 및 시간 단위 포함). 예를 들어 "40세입니다"라는 문구에서는 "40세"를 나이로 Amazon Transcribe 인식합니다.
DATE_TIME	날짜에는 년, 월, 일, 요일 또는 시각이 포함될 수 있습니다. 예를 들어는 "2020년 1월 19일" 또는 "오전 11시"를 날짜로 Amazon Transcribe 인식합니다. Amazon Transcribe 는 부분 날짜, 날짜 범위 및 날짜 간격을 인식합니다. 또한 "1990년대"와 같은 십년 단위를 인식할 것입니다.
LICENSE_PLATE	차량 번호판은 차량이 등록된 주 또는 국가에서 발급합니다. 승용차의 형식은 일반적으로 대문자와 숫자로 구성된 5~8자리 숫자입니다. 형식은 발급한 주 또는 국가의 위치에 따라 다릅니다.
PASSPORT_NUMBER	개인의 여권에 할당된 고유 식별자입니다. 형식에는 일반적으로 문자와 숫자의 조합이 포함되며 국가별로 다릅니다.
PASSWORD	비밀번호로 사용되는 영숫자 문자열(예: "*very20special#pass*")입니다.
USERNAME	계정을 식별하는 사용자 이름(예: 로그인 이름, 화면 이름, 닉네임 또는 핸들).

PII 유형	설명
VEHICLE_IDENTIFICATION_NUMBER	차량 식별 번호(VIN)는 차량을 고유하게 식별합니다. VIN 콘텐츠와 형식은 ISO 3779 사양에 정의되어 있습니다. 각 국가별로 VIN에 대한 특정 코드와 형식을 가지고 있습니다.

AWS Management Console, WebSocket 또는 HTTP/2를 사용하여 스트리밍 트랜스크립션을 시작할 수 있습니다.

AWS Management Console

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창에서 실시간 트랜스크립션을 선택합니다. 콘텐츠 제거 설정까지 아래로 스크롤하고 최소화된 경우 이 필드를 확장합니다.



3. PII 식별 및 수정을 봅니다.

▼ Content removal settings

☐ Vocabulary filtering [Info](#)
Vocabulary filtering removes, masks, or tags words that you specify in your vocabulary filter. Choose a vocabulary filter to see an example.

☐ PII Identification & redaction [Info](#)
Identify or redact one or more types of personally identifiable information (PII) in your transcript

4. 식별만 또는 식별 및 수정을 선택한 다음, 트랜스크립트에서 식별하거나 수정하려는 PII 엔터티 유형을 선택합니다.

▼ Content removal settings

☐ Vocabulary filtering [Info](#)
Vocabulary filtering removes, masks, or tags words that you specify in your vocabulary filter. Choose a vocabulary filter to see an example.

☒ PII Identification & redaction [Info](#)
Identify or redact one or more types of personally identifiable information (PII) in your transcript

Select PII detection type

☒ Identification only
Label the type of PII identified but not redact it in the transcription output

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

Select PII entity types (22 of 22 selected)

☒ Select All

☒ Financial (6 of 6 selected)

☒ BANK_ACCOUNT_NUMBER	☒ BANK_ROUTING	☒ CREDIT_DEBIT_NUMBER
☒ CREDIT_DEBIT_CVV	☒ CREDIT_DEBIT_EXPIRY	☒ PIN

☒ Personal (8 of 8 selected)

☒ NAME	☒ ADDRESS	☒ PHONE
☒ EMAIL	☒ SSN	☒ PASSPORT_NUMBER
☒ DRIVER_ID	☒ AGE	

☒ Digital footprint (7 of 7 selected)

☒ URL	☒ USERNAME	☒ PASSWORD
☒ AWS_ACCESS_KEY	☒ AWS_SECRET_KEY	☒ IP_ADDRESS
☒ MAC_ADDRESS		

☒ Other (1 of 1 selected)

☒ DATE_TIME

► Customizations

5. 이제 스트림을 트랜스크립션할 준비가 되었습니다. 스트리밍 시작을 선택하고 말하기 시작합니다. 구술을 끝내려면 스트리밍 중지를 선택합니다.

WebSocket 스트림

이 예시에서는 WebSocket 스트림에서 PII 수정(또는 PII 식별)을 사용하는 미리 서명된 URL을 만듭니다. 가독성을 높이기 위해 줄바꿈이 추가되었습니다. 에서 WebSocket 스트림을 사용하는 방법에 대한 자세한 내용은 섹션을 Amazon Transcribe참조하세요 [WebSocket 스트림 설정](#). 파라미터에 대한 자세한 내용은 [StartStreamTranscription](#)을 참조하세요.

```
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)
```

동일한 요청에서 content-identification-type 및 content-redaction-type 둘 다 사용할 수는 없습니다.

파라미터 정의는 [API 참조](#)에서 찾을 수 있습니다. 모든 AWS API 작업에 공통적인 파라미터는 [공통 파라미터](#) 섹션에 나열되어 있습니다.

HTTP/2 스트림

이 예시에서는 PII 식별 또는 PII 수정이 활성화된 HTTP/2 요청을 생성합니다. 에서 HTTP/2 스트리밍을 사용하는 방법에 대한 자세한 내용은 섹션을 [Amazon Transcribe참조하세요HTTP/2 스트림 설정](#). 관련 파라미터 및 헤더에 대한 자세한 내용은 섹션을 [Amazon Transcribe참조하세요StartStreamTranscription](#).

```
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
```

동일한 요청에서 content-identification-type 및 content-redaction-type 둘 다 사용할 수는 없습니다.

파라미터 정의는 [API 참조](#)에서 찾을 수 있습니다. 모든 AWS API 작업에 공통적인 파라미터는 [공통 파라미터](#) 섹션에 나열되어 있습니다.

Note

스트리밍을 위한 PII 수정은 AWS 리전아시아 태평양(서울), 아시아 태평양(시드니), 아시아 태평양(도쿄), 캐나다(중부), EU(프랑크푸르트), EU(아일랜드), EU(런던), 미국 동부(버지니아 북부), 미국 동부(오하이오), 미국 서부(오레곤)에서만 지원됩니다.

PII 수정 및 식별 출력 예시

다음 예시는 배치 작업 및 스트리밍 작업의 편집된 출력 및 스트리밍 작업의 PII 식별을 보여줍니다.

콘텐츠 수정을 사용하는 트랜스크립션 작업은 두 가지 유형의 confidence 값을 생성합니다. 자동 음성 인식(ASR) 신뢰도는 type 또는 pronunciation punctuation을 갖는 항목이 특정 발화임을 나타냅니다. 다음 트랜스크립트 출력에서 단어 Good은 confidence가 1.0입니다. 이 신뢰도 값은 Amazon Transcribe가 이 트랜스크립트에서 발화된 단어가 'Good'임을 100% 신뢰한다는 것을 나타냅니다. [PII] 태그의 confidence 값은 수정하도록 플래그가 지정된 음성이 실제로 PII라는 것의 신뢰도입니다. 다음 트랜스크립트 출력에서 confidence 0.9999는 Amazon Transcribe가 트랜스크립트에서 편집한 엔터티가 PII임을 99.99% 신뢰한다는 것을 나타냅니다.

편집된 출력(배치) 예시

```
{
  "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"

```



```
}

```

비교를 위해 편집되지 않은 스크립트는 다음과 같습니다.

```
{
  "jobName": "job id",
  "accountId": "111122223333",
  "isRedacted": false,
  "results": {
    "transcripts": [
      {
        "transcript": "Good morning, everybody. My name is Mike, and today I
feel like
my Social
Security number 000000000. My credit card number is 5555555555555555
and my C V V code is 000. I hope that Amazon Transcribe is doing a good
job
at redacting that personal information away. Let's check."
      }
    ],
    "items": [
      {
        "id": 0,
        "start_time": "2.86",
        "end_time": "3.35",
        "alternatives": [
          {
            "confidence": "1.0",
            "content": "Good"
          }
        ],
        "type": "pronunciation"
      },
      Items removed for brevity
      {
        "id": 8,
        "start_time": "5.56",
        "end_time": "6.25",
        "alternatives": [
          {
            "confidence": "0.9999",
            "content": "Mike",

```

```

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

```

편집된 스트리밍 출력 예시

```

{
  "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
}
]
}
}
}
}
}

```

PII 식별 출력 예시

PII 식별은 스트리밍 트랜스크립션 작업에 사용할 수 있는 추가 기능입니다. 식별된 PII는 각 세그먼트의 Entities 섹션에 나열되어 있습니다.

```

{
  "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
}
]
}

```

```
}  
  }  
}
```

비디오 자막 생성

Amazon Transcribe는 WebVTT(*.vtt) 및 SubRip(*.srt) 출력을 지원하여 비디오 자막으로 사용할 수 있습니다. 배치 비디오 트랜스크립션 작업을 설정할 때 파일 유형을 하나 또는 두 개 모두 선택할 수 있습니다. 자막 기능을 사용하면 선택한 자막 파일과 일반 트랜스크립트 파일(추가 정보 포함)이 생성됩니다. 자막 파일과 트랜스크립션 파일은 동일한 대상으로 출력됩니다.

텍스트가 발화되는 것과 동시에 자막이 표시되며 말이 자연스럽게 끊어읽거나 화자가 말을 마칠 때까지 계속 자막이 표시됩니다. 트랜스크립션 요청에서 자막을 활성화했는데 오디오에 음성이 없는 경우 자막 파일은 생성되지 않습니다.

Important

Amazon Transcribe는 자막 출력의 기본 시작 색인인 0을 사용하는데, 이는 널리 사용되는 1 값과는 다릅니다. 1의 시작 인덱스가 필요한 경우 [OutputStartIndex](#) 파라미터를 사용하여 AWS Management Console 또는 API 요청에서 시작 인덱스를 지정할 수 있습니다.

잘못된 시작 색인을 사용하면 다른 서비스와의 호환성 오류가 발생할 수 있으므로 자막을 만들기 전에 필요한 시작 색인을 확인해야 합니다. 어떤 값을 사용해야 할지 확실하지 않은 경우 1을 선택하는 것이 좋습니다. 자세한 내용은 [Subtitles](#) 섹션을 참조하세요.

자막과 함께 지원되는 기능:

- 콘텐츠 수정 — 수정된 모든 콘텐츠는 자막과 일반 트랜스크립트 출력 파일 모두에 'PII'로 반영됩니다. 오디오는 변경되지 않습니다.
- 어휘 필터 — 자막 파일은 트랜스크립션 파일에서 생성되므로 표준 트랜스크립션 출력에서 필터링한 모든 단어는 자막에서도 필터링됩니다. 필터링된 콘텐츠는 공백으로 표시되거나 트랜스크립트와 자막 파일에 ***로 표시됩니다. 오디오는 변경되지 않습니다.
- 화자 분할 — 특정 자막 세그먼트에 화자가 여러 명 있는 경우 대시를 사용하여 각 화자를 구분합니다. 이는 WebVTT 및 SubRIP 형식 모두에 적용됩니다. 예를 들면 다음과 같습니다.
 - -- 사람 1이 말한 텍스트
 - -- 사람 2가 말한 텍스트

자막 파일은 트랜스크립션 출력과 동일한 Amazon S3 위치에 저장됩니다.

자막 제작에 대한 비디오 시연은 [Amazon Transcribe Video Snacks: 코드를 작성하지 않고 비디오 자막 제작](#)을 참조하세요.

자막 파일 생성

AWS Management Console, AWS CLI 또는 AWS SDK를 사용하여 자막 파일을 만들 수 있습니다. 다음 예를 참조하세요.

AWS Management Console

1. 에 로그인합니다..[AWS Management Console](#)
2. 탐색 창에서 트랜스크립션 작업을 선택한 다음 작업 생성(오른쪽 상단)을 선택합니다. 그러면 작업 세부 정보 지정 페이지가 열립니다. 자막 옵션은 출력 데이터 패널에 있습니다.
3. 원하는 자막 파일 형식을 선택한 다음 시작 색인 값을 선택합니다. Amazon Transcribe 기본값은 0이지만 1이 더 널리 사용됩니다. 어떤 값을 사용해야 할지 확실하지 않은 경우 다른 서비스와의 호환성이 향상될 수 있으므로 1을 선택하는 것이 좋습니다.

Output data

Output data location type info [Info](#)

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

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

Subtitle file format [Info](#)

☒ SRT (SubRip)

☐ VTT (WebVTT)

Specify the start index

0 ▼

4. 작업 세부 정보 지정 페이지에 포함하려는 다른 필드를 모두 채운 후 다음을 선택합니다. 그러면 작업 구성 - 선택 사항 페이지로 이동합니다.
5. 작업 생성을 선택하여 트랜스크립션 작업을 실행합니다.

AWS CLI

이 예시에서는 [start-transcription-job](#) 명령 및 Subtitles 파라미터를 사용합니다. 자세한 내용은 [StartTranscriptionJob](#) 및 [Subtitles](#)을(를) 참조하세요.

```
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
```

다음은 [start-transcription-job](#) 명령을 사용하는 또 다른 예 및 해당 작업에 자막을 추가하는 요청 본문입니다.

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

my-first-subtitle-job.json 파일에는 다음과 같은 요청 본문이 포함되어 있습니다.

```
{
  "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 for Python (Boto3)

이 예에서는 AWS SDK for Python (Boto3)를 통해 [start_transcription_job](#) 메서드의 Subtitles 인수를 사용하여 자막을 추가합니다. 자세한 내용은 [StartTranscriptionJob](#) 및 [Subtitles](#)을(를) 참조하세요.

기능별, 시나리오 및 교차 서비스 예를 포함하여 AWS SDK를 사용하는 추가 예시는 [AWS SDKs를 사용한 Amazon Transcribe의 코드 예제](#) 장을 참조하세요.


```
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)
```

통화 분석을 사용한 콜센터 오디오 분석

Amazon Transcribe Call Analytics를 사용하여 고객-에이전트 상호 작용에 대한 인사이트를 얻습니다. 통화 분석은 콜 센터 오디오용으로 특별히 설계되었으며 각 통화 및 각 참가자와 관련된 중요한 데이터를 자동으로 제공합니다. 또한 통화 중 특정 지점의 데이터에 집중할 수 있습니다. 예를 들어, 통화의 처음 몇 초와 마지막 분기까지의 고객 감정을 비교하여 에이전트가 긍정적인 경험을 제공했는지 확인할 수 있습니다. 다른 사용 사례는 [다음 섹션](#)에 나열되어 있습니다.

통화 분석은 통화 후 트랜스크립션과 실시간 트랜스크립션에 사용할 수 있습니다. Amazon S3 버킷에 있는 파일을 트랜스크립션하는 경우 통화 후 트랜스크립션을 수행하는 것입니다. 오디오 스트림을 트랜스크립션하는 경우 실시간 트랜스크립션을 수행하는 것입니다. 이 두 가지 트랜스크립션 방법은 서로 다른 통화 분석 인사이트와 기능을 제공합니다. 각 방법에 대한 자세한 내용은 [통화 후 분석](#) 및 [실시간 통화 분석](#) 섹션을 참조하세요.

실시간 통화 분석 트랜스크립션을 사용하면 요청에 [통화 후 분석](#)을 포함할 수도 있습니다. 통화 후 분석 기록은 요청에 지정한 Amazon S3 버킷에 저장됩니다. 자세한 내용은 [실시간 트랜스크립션을 통한 통화 후 분석](#) 섹션을 참조하세요.

통화 분석과 관련된 API 작업

통화 후: [CreateCallAnalyticsCategory](#), [DeleteCallAnalyticsCategory](#), [DeleteCallAnalyticsJob](#), [GetCallAnalyticsCategory](#), [GetCallAnalyticsJob](#), [ListCallAnalyticsCategories](#), [ListCallAnalyticsJobs](#), [StartCallAnalyticsJob](#), [UpdateCallAnalyticsCategory](#)
 실시간: [StartCallAnalyticsStreamTranscription](#),
[StartCallAnalyticsStreamTranscriptionWebSocket](#)

일반 사용 사례

통화 후 트랜스크립션:

- 시간 경과에 따른 문제 빈도 모니터링: [통화 분류](#)를 사용하여 대화 내용 내에서 반복되는 키워드를 식별합니다.
- 고객 서비스 경험에 대한 인사이트 확보: [통화 특성](#)(침묵 시간, 발언 시간, 중단, 음성 크기, 발언 속도)과 감정 분석을 사용하여 통화 중에 고객 문제가 적절하게 해결되고 있는지 확인합니다.

- 규정 준수 또는 회사 정책 준수 확인: 회사별 인사말 또는 고지 사항의 [키워드와 문구를](#) 설정하여 에이전트가 규제 요구 사항을 충족하는지 확인합니다.
- 고객 개인 데이터 처리 개선: 트랜스크립션 출력 또는 오디오 파일에 [PII 수정](#)을 사용하여 고객 개인 정보를 보호합니다.
- 직원 교육 개선: 기준(감정, 침묵 시간, 중단, 발언 속도)을 사용하여 긍정적이거나 부정적인 고객 상호 작용의 예로 사용할 수 있는 트랜스크립트에 플래그를 지정합니다.
- 긍정적인 고객 경험을 만들기 위한 직원 효율성 측정: [감성 분석](#)을 사용하여 에이전트가 통화가 진행됨에 따라 부정적인 고객 감정을 긍정적인 감정으로 바꿀 수 있는지 측정합니다.
- 데이터 구성 개선: [사용자 지정 범주](#)(키워드와 구절, 감정, 발언 시간, 중단 등)를 기반으로 통화에 레이블을 지정하고 분류합니다.
- 생성형 AI를 사용하여 통화의 중요한 측면 요약: [생성형 통화 요약](#)을 사용하여 통화에서 논의한 문제, 조치 항목, 결과 등의 주요 구성 요소를 포함하는 트랜스크립트의 간결한 요약을 얻습니다.

실시간 트랜스크립션:

- 실시간 에스컬레이션 완화: "매니저와 대화"라는 표현을 사용하는 고객의 경우와 같이, 주요 구절에 [실시간 알림](#)을 설정하여 통화가 에스컬레이션되기 시작할 때 플래그를 지정합니다. 실시간 범주 일치 사용하여 실시간 알림을 생성할 수 있습니다.
- 고객 데이터 처리 개선: 트랜스크립션 출력에 [PII 식별](#) 또는 [PII 수정](#)을 사용하여 고객 개인 정보를 보호합니다.
- 사용자 지정 키워드 및 구절 식별: [사용자 지정 범주](#)를 사용하여 통화 시 특정 키워드에 플래그를 지정합니다.
- 문제 자동 식별: 자동 [문제 감지](#) 기능을 사용하여 통화에서 확인된 모든 문제를 간결하게 요약할 수 있습니다.
- 긍정적인 고객 경험을 만들기 위한 직원 효율성 측정: [감성 분석](#)을 사용하여 에이전트가 통화가 진행됨에 따라 부정적인 고객 감정을 긍정적인 감정으로 바꿀 수 있는지 측정합니다.
- 에이전트 지원 설정: 원하는 인사이트를 사용하여 에이전트가 고객 전화를 해결하는 데 능동적으로 도움을 줄 수 있습니다. 자세한 내용은 [Amazon 언어 AI 서비스를 사용하는 고객 센터의 Live 통화 분석 및 에이전트 지원](#)을 참조하세요.

Call Analytics에서 사용할 수 있는 기능을 Amazon Transcribe 및 Amazon Transcribe Medical용 기능과 비교하려면 [기능 표](#)를 참조하세요.

시작하려면 [통화 후 분석 트랜스크립션 시작](#) 및 [실시간 통화 분석 트랜스크립션 시작](#) 섹션을 참조하세요. 통화 분석 출력은 표준 트랜스크립션 작업의 출력과 비슷하지만 추가 분석 데이터를 포함합니다. 샘플 출력을 보려면 [통화 후 분석 출력](#) 및 [실시간 통화 분석 출력](#)을 참조하세요.

고려 사항 및 추가 정보

통화 분석을 사용하기 전에 다음 사항을 참고하세요.

- 통화 분석은 2채널 오디오만 지원합니다. 한 채널에는 에이전트가 있고 두 번째 채널에는 고객이 있습니다.
- [작업 대기열](#)은 통화 후 분석 작업에 항상 사용할 수 있으므로 동시 통화 분석 작업은 100개로 제한됩니다. 할당량 증가를 요청하려면 [AWS 서비스 할당량](#) 섹션을 참조하세요.
- 통화 후 분석 작업의 입력 파일은 500MB를 초과할 수 없으며 4시간 미만이어야 합니다. 주파수가 8kHz보다 큰 오디오 파일의 경우 오디오 지속 시간은 2시간 이하여야 합니다. 특정 압축된 비WAV 오디오 파일 형식의 경우 파일 크기 제한이 더 작을 수 있습니다.
- 범주를 사용하는 경우 통화 분석 트랜스크립션을 시작하기 전에 원하는 범주를 모두 만들어야 합니다. 기존 트랜스크립션에는 새 범주를 적용할 수 없습니다. 새 범주를 만드는 방법을 알아보려면 [통화 후 자막의 범주 생성](#) 및 [실시간 트랜스크립션의 범주 생성](#) 섹션을 참조하세요.
- 일부 Call Analytics 할당량은 Amazon Transcribe 및 Amazon Transcribe Medical과 다릅니다. 자세한 내용은 [AWS 일반 참조](#)를 참조하세요.

AWS Machine Learning 블로그로 더 자세히 알아보기

통화 분석 옵션에 대한 자세한 내용은 다음을 참조하세요.

- [Amazon 언어 AI 서비스를 사용하는 고객 센터의 사후 통화 분석](#)
- [Amazon 언어 AI 서비스를 사용하는 고객 센터의 실시간 통화 분석 및 에이전트 지원](#)

샘플 통화 분석 출력 및 기능을 보려면 [GitHub](#) 데모를 참조하세요. 또한 트랜스크립트를 읽기 쉬운 형식으로 변환할 수 있는 [JSON-Word 문서](#) 애플리케이션도 제공합니다.

리전 가용성 및 할당량

Call Analytics는 AWS 리전다음에서 지원됩니다.

리전	트랜스크립션 유형
ap-northeast-1(도쿄)	post-call, real-time
ap-northeast-2(서울)	post-call, real-time
ap-south-1(뭄바이)	post-call
ap-southeast-1(싱가포르)	post-call
ap-southeast-2(시드니)	post-call, real-time
ca-central-1(캐나다, 중부)	post-call, real-time
eu-central-1(프랑크푸르트)	post-call, real-time
eu-west-2(런던)	post-call, real-time
us-east-1(버지니아 북부)	post-call, real-time
us-west-2(오레곤)	post-call, real-time

[Amazon Transcribe](#), [Amazon Transcribe Medical](#) 및 통화 분석에 대한 리전 지원은 다릅니다.

지원되는 각 리전에 대한 엔드포인트를 얻으려면 AWS 일반 참조의 [서비스 엔드포인트](#)를 참조하세요.

트랜스크립션과 관련된 할당량 목록은 AWS 일반 참조의 [서비스 할당량](#)을 참조하세요. 일부 할당량은 요청 시 변경될 수 있습니다. 조정 가능 열에 '예'가 포함된 경우 증가를 요청할 수 있습니다. 이렇게 하려면 제공된 링크를 선택합니다.

통화 후 분석

통화 분석은 고객 서비스 동향을 모니터링하는 데 유용한 통화 후 분석을 제공합니다.

통화 후 트랜스크립션은 다음과 같은 인사이트를 제공합니다.

- 발원 시간, 침묵 시간, 화자 음량, 중단, 발원 속도, 문제, 결과, 조치 항목 등의 [통화 특성](#)
- 전체 통화의 간결한 요약 생성하는 [생성형 통화 요약](#)
- 특정 키워드와 기준을 정확히 파악하는 데 사용할 수 있는 규칙을 포함한 [사용자 지정 분류](#)
- 텍스트 트랜스크립트 및 오디오 파일의 [PII 수정](#)

- 통화의 다양한 시점에서 각 발신자의 [화자 감정](#)

통화 후 인사이트

이 섹션에서는 통화 후 분석 트랜스크립션에 사용할 수 있는 인사이트를 자세히 설명합니다.

통화 특성

통화 특성 기능은 다음 기준을 사용하여 에이전트-고객 상호 작용의 품질을 측정합니다.

- 중단: 문장 중간에 한 참가자가 다른 참가자의 말을 중간에 끊는지 여부와 그 시점을 측정합니다. 잦은 중단은 무례함이나 분노와 관련이 있을 수 있으며, 참가자 중 한 명 또는 두 명 모두에게 부정적인 감정을 불러올 수 있습니다.
- 음량: 각 참가자가 말하는 음량을 측정합니다. 이 지표를 사용하여 발신자 또는 에이전트가 큰 소리로 말하거나 소리를 지르는지 확인합니다. 이는 보통 화가 났음을 나타냅니다. 이 지표는 0에서 100까지의 척도로 정규화된 값(특정 세그먼트의 초당 음성 수준)으로 표시되며, 값이 높을수록 목소리가 커집니다.
- 침묵 시간: 음성이 포함되지 않은 시간을 측정합니다. 이 지표를 사용하면 에이전트가 고객을 과도한 시간 동안 대기 상태로 두는 등 오랜 시간 동안 침묵이 유지되는지 확인할 수 있습니다.
- 발연 속도: 두 참가자가 말하는 속도를 측정합니다. 한 참가자가 너무 빨리 말을 하면 이해력에 영향을 미칠 수 있습니다. 이 지표는 분당 단어 수로 측정합니다.
- 발연 시간: 통화 중에 각 참가자가 발연한 시간(밀리초)을 측정합니다. 이 지표를 사용하면 한 참가자가 통화를 주도하고 있는지 또는 대화가 균형을 이루고 있는지 파악하는 데 도움이 됩니다.
- 문제, 결과 및 조치 항목: 통화 트랜스크립트에서 문제, 결과 및 조치 항목을 식별합니다.

다음은 [출력 예](#)입니다.

생성형 통화 요약

생성형 통화 요약은 통화 이유, 문제 해결을 위해 취한 조치, 후속 조치와 같은 주요 구성 요소를 추출하여 전체 통화의 간결한 요약을 생성합니다.

생성형 통화 요약을 사용하면 다음과 같은 이점이 있습니다.

- 통화 도중과 통화 후에 수동으로 메모를 할 필요가 줄어듭니다.
- 통화 후 작업을 수행하는 대신 대기열에서 대기 중인 발신자와 대화하는 데 더 많은 시간을 할애할 수 있으므로 에이전트의 효율성이 개선됩니다.

- 통화 요약은 전체 트랜스크립트보다 훨씬 빠르게 검토할 수 있으므로 감독자의 검토 시간이 단축됩니다.

통화 후 분석 작업에 생성형 통화 요약을 사용하려면 [생성형 통화 요약 활성화](#)를 참조하세요. 출력 예시는 [생성형 통화 요약 출력 예시](#)를 참조하세요. 생성형 통화 요약은 별도로 요금이 책정됩니다([요금 페이지](#) 참조).

Note

생성형 통화 요약은 현재 us-east-1 및 us-west-2에서 사용할 수 있습니다. 이 기능은 호주(en-AU), 영국(en-GB), 인도(en-IN), 아일랜드(en-IE), 스코틀랜드(en-AB), 미국(en-US) 및 웨일스(en-WL) 영어로 지원됩니다.

사용자 지정 분류

통화 분류를 사용하여 통화 내의 키워드, 구절, 감정 또는 행동에 플래그를 지정할 수 있습니다. 분류 옵션을 사용하면 중단 횟수가 많은 부정적인 감정의 통화 등의 에스컬레이션을 분류하거나 통화를 회사 부서 등의 특정 범주로 분류할 수 있습니다.

범주에 추가할 수 있는 기준은 다음과 같습니다.

- 침묵 시간: 고객이나 에이전트가 발언하지 않은 시간.
- 중단: 고객이나 에이전트가 상대방의 말을 끊는 경우.
- 고객 또는 에이전트의 감정: 특정 기간 동안 고객이나 에이전트가 느끼는 감정. 지정된 기간 동안 대화의 50% 이상이 (두 화자 back-and-forth 간의 전후) 지정된 감정과 일치하는 경우는 해당 감정을 Amazon Transcribe 일치시킵니다.
- 키워드 또는 구절: 정확한 구절을 기반으로 트랜스크립션의 일부를 일치시킵니다. 예를 들어 '매니저와 대화하고 싶습니다'라는 구절에 필터를 설정하면 Amazon Transcribe 는 정확히 일치하는 구절을 필터링합니다.

이전 기준과 반대(발언 시간, 중단 없음, 감정 없음, 특정 구절 없음)에 플래그를 지정할 수도 있습니다.

다음은 [출력 예](#)입니다.

범주에 대한 자세한 내용이나 새 범주를 만드는 방법에 대한 자세한 내용은 [통화 후 자막의 범주 생성](#) 섹션을 참조하세요.

민감한 데이터 수정

민감한 데이터 수정은 텍스트 트랜스크립트와 오디오 파일의 개인 식별 정보(PII)를 대체합니다. 수정된 트랜스크립트는 원본 텍스트를 [PII]로 대체하고, 수정된 오디오 파일은 발음한 개인 정보를 무음으로 대체합니다. 이 파라미터는 고객 정보를 보호하는 데 유용합니다.

Note

통화 후 PII 수정에는 미국 영어(en-US)와 미국 스페인어(es-US)가 지원됩니다.

이 기능을 사용하여 수정된 PII 목록을 보거나 Amazon Transcribe를 사용한 수정에 대해 자세히 알아보려면 [개인 식별 정보 편집 또는 식별](#) 섹션을 참조하세요.

다음은 [출력 예](#)입니다.

감정 분석

감정 분석은 통화 내내 고객과 에이전트가 느끼는 감정을 추정합니다. 이 지표는 양적 값(범위: 5~-5)과 질적 값(positive, neutral, mixed 또는 negative)으로 표시됩니다. 질적 값은 분기별 및 통화별로 제공되고, 양적 값은 터별로 제공됩니다.

이 지표는 통화가 끝날 때까지 에이전트가 화가 난 고객을 기쁘게 할 수 있는지 파악하는 데 도움이 될 수 있습니다.

감정 분석은 기본적으로 작동하므로 모델 교육 또는 사용자 지정 범주와 같은 사용자 지정을 지원하지 않습니다.

다음은 [출력 예](#)입니다.

통화 후 자막의 범주 생성

통화 후 분석은 사용자 지정 범주 생성을 지원하므로 특정 비즈니스 요구 사항에 가장 적합하도록 트랜스크립트 분석을 조정할 수 있습니다.

다양한 시나리오를 다루기 위해 원하는 만큼 범주를 생성할 수 있습니다. 범주를 하나 생성할 때마다 1~20개의 규칙을 생성해야 합니다. 각 규칙은 중단, 키워드, 침묵 시간, 감정 등 4가지 기준 중 하나를 기반으로 합니다. [CreateCallAnalyticsCategory](#) 작업에 이러한 기준을 사용하는 방법에 대한 자세한 내용은 [통화 후 분석 범주의 규칙 기준](#) 섹션을 참조하세요.

미디어의 콘텐츠가 특정 범주에서 지정한 모든 규칙과 일치하는 경우 Amazon Transcribe에서는 해당 범주로 출력에 레이블을 지정합니다. JSON 출력의 범주 일치 예는 [통화 분류 출력](#)을 참조하세요.

사용자 지정 범주로 수행할 수 있는 작업의 몇 가지 예는 다음과 같습니다.

- 부정적인 고객 감정으로 끝나는 통화와 같이 특정 특성을 가진 통화를 분리합니다.
- 특정 키워드 세트에 플래그를 지정하고 추적하여 고객 문제의 추세를 파악합니다.
- 에이전트가 통화 후 처음 몇 초 동안 특정 구절을 말하거나 생략하는 등, 규정 준수를 모니터링합니다.
- 에이전트의 중단이 많고 부정적인 고객 감정이 있는 통화에 플래그를 지정하여 고객 경험에 대한 인사이트를 확보합니다.
- 환영 문구를 사용하는 에이전트가 긍정적인 고객 감정과 상관관계가 있는지 분석하는 등 여러 범주를 비교하여 상관관계를 측정합니다.

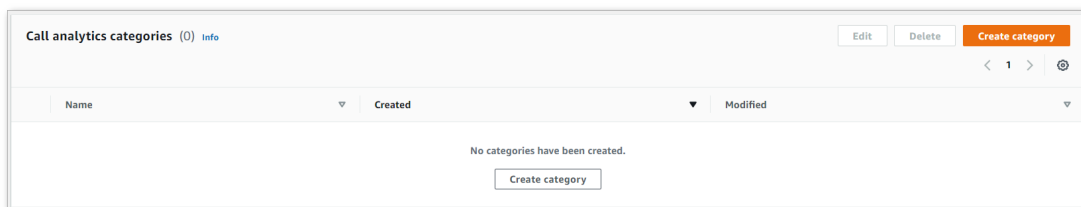
통화 후 범주와 실시간 범주의 비교

새 범주를 만들 때 해당 범주를 통화 후 분석 범주(POST_CALL)로 생성할지 아니면 실시간 통화 분석 범주-REAL_TIME)로 생성할지 지정할 수 있습니다. 옵션을 지정하지 않으면 기본적으로 범주가 통화 후 범주로 생성됩니다. 통화 후 분석 트랜스크립션이 완료되면 출력에서 통화 후 분석 범주 일치를 확인할 수 있습니다.

통화 후 분석을 위한 새 범주를 생성하려면 AWS Management Console, AWS CLI 또는 AWS SDK를 사용할 수 있습니다. 예를 보려면 다음을 참조하세요.

AWS Management Console

1. 탐색 창의 아래에서 Amazon Transcribe Call Analytics를 Amazon Transcribe 선택합니다.
2. 통화 분석 범주를 선택하면 통화 분석 범주 페이지로 이동합니다. 범주 생성을 선택합니다.



3. 이제 범주 생성 페이지로 이동했습니다. 범주 이름을 입력한 다음 범주 유형 드롭다운 메뉴에서 '배치 통화 분석'을 선택합니다.

Category settings

Category name

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

Category type [Info](#)

Batch call analytics

Real time call analytics

☒ Use a template (recommended)
 Use a template to edit predefined rules.

☐ Create from scratch
 If you know the rules that you want to define, choose this option.

Template type [Info](#)
 Choose the template for the category that most closely matches the one you want to create.

4. 템플릿을 선택하여 범주를 생성하거나 처음부터 새로 생성할 수 있습니다.

템플릿을 사용하는 경우: 템플릿 사용(권장)을 선택하고 원하는 템플릿과 범주 생성을 차례로 선택합니다.

Category settings

Category name

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

Category type [Info](#)

Category creation method [Info](#)

☒ Use a template (recommended)
 Use a template to edit predefined rules.

☐ Create from scratch
 If you know the rules that you want to define, choose this option.

Template type [Info](#)
 Choose the template for the category that most closely matches the one you want to create.

Non-talk time exceeds 5 minutes for the whole call
 Customer sentiment is negative for the last 5 minutes of the call
 Agent spoke over the customer more than 15 seconds for the entire call

5. 사용자 지정 범주를 생성하는 경우: 새로 생성을 선택합니다.

Create category [Info](#)

Category settings

Category name

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

Category creation method [Info](#)

☐ Use a template (recommended)
Use a template to edit predefined rules.

☒ Create from scratch
If you know the rules that you want to define, choose this option.

Rules

All the rule conditions must be met for a transcription job to be classified in this category.

▼ Rule 1 Delete rule

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

Add rule

You can add up to 19 more rules.

6. 드롭다운 메뉴를 사용하여 범주에 규칙을 추가합니다. 범주당 최대 20개의 규칙을 추가할 수 있습니다.

Rules

All the rule conditions must be met for a transcription job to be classified in this category.

▼ Rule 1 Delete rule

When no word has been spoken for more than 5 minute(s) during the entire call.

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

Non-talk time ▲

Non-talk time
Trigger the rule when nothing has been said for more than the time that you specify.

Interruption time
Trigger the rule when the speaker has been interrupted for more than the time that you specify.

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

Transcript sentiment match
Trigger the rule when the sentiment of the speaker matches the sentiment that you specify.

Add rule

You can add up to 19 more rules.

7. 다음은 두 가지 규칙이 있는 범주의 예입니다. 하나는 통화 중에 15초 이상 고객의 말을 끊는 에이전트이고, 하나는 통화 마지막 2분 동안 고객 또는 에이전트가 느낀 부정적인 감정입니다.

Rules

All the rule conditions must be met for a transcription job to be classified in this category.

▼ Rule 1

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

Rule type

Info

Choose the rule that you want to define.

Interruption time ▼

Logic

Info

Define the conditions that must be met.

When the duration of the interruption was more than

15

second(s) ▼

during the

entire call ▼

when the speaker was

agent ▼

AND

▼ Rule 2

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

Rule type

Info

Choose the rule that you want to define.

Transcript sentiment match ▼

Logic

Info

Define the conditions that must be met.

When the sentiment is

negative ▼

during the

last ▼

2

minute(s) ▼

when the speaker was

either ▼

Add rule

You can add up to 18 more rules.

8. 범주에 규칙을 모두 추가했으면 범주 생성을 선택합니다.

AWS CLI

이 예에서는 [create-call-analytics-category](#) 명령을 사용합니다. 자세한 내용은 [CreateCallAnalyticsCategory](#), [CategoryProperties](#), [Rule](#) 섹션을 참조하세요.

다음 예에서는 규칙을 사용하여 범주를 생성합니다.

- 고객은 처음 60,000밀리초 내에 방해를 받았습니다. 이러한 중단은 지속 시간은 최소 10,000밀리초였습니다.
- 통화 시작 후 10%와 80% 사이에 최소 20,000밀리초 동안 침묵이 지속되었습니다.
- 에이전트는 통화 중 어느 시점에서 부정적인 감정을 느꼈습니다.
- 통화의 첫 10,000밀리초 동안 "환영합니다" 또는 "안녕하세요"라는 단어를 사용하지 않았습니다.

범주 생성

334

이 예에서는 [create-call-analytics-category](#) 명령과 범주에 여러 규칙을 추가하는 요청 본문을 사용합니다.

```
aws transcribe create-call-analytics-category \  
--cli-input-json file://filepath/my-first-analytics-category.json
```

my-first-analytics-category.json 파일에는 다음과 같은 요청 본문이 포함되어 있습니다.

```
{  
  "CategoryName": "my-new-category",  
  "InputType": "POST_CALL",  
  "Rules": [  
    {  
      "InterruptionFilter": {  
        "AbsoluteTimeRange": {  
          "First": 60000  
        },  
        "Negate": false,  
        "ParticipantRole": "CUSTOMER",  
        "Threshold": 10000  
      }  
    },  
    {  
      "NonTalkTimeFilter": {  
        "Negate": false,  
        "RelativeTimeRange": {  
          "EndPercentage": 80,  
          "StartPercentage": 10  
        },  
        "Threshold": 20000  
      }  
    },  
    {  
      "SentimentFilter": {  
        "ParticipantRole": "AGENT",  
        "Sentiments": [  
          "NEGATIVE"  
        ]  
      }  
    },  
    {  
      "TranscriptFilter": {  
        "Negate": true,  

```

```

        "AbsoluteTimeRange": {
            "First": 10000
        },
        "Targets": [
            "welcome",
            "hello"
        ],
        "TranscriptFilterType": "EXACT"
    }
}
]
}

```

AWS SDK for Python (Boto3)

이 예제에서는 AWS SDK for Python (Boto3) 를 사용하여 [create_call_analytics_category](#) 메서드에 대한 `CategoryName` 및 `Rules` 인수를 사용하여 범주를 생성합니다. 자세한 내용은 [CreateCallAnalyticsCategory](#), [CategoryProperties](#), [Rule](#) 섹션을 참조하세요.

기능별, 시나리오 및 교차 서비스 예제를 포함하여 AWS SDKs를 사용하는 추가 예제는 [AWS SDKs를 사용한 Amazon Transcribe의 코드 예제](#)를 참조하세요.

다음 예에서는 규칙을 사용하여 범주를 생성합니다.

- 고객은 처음 60,000밀리초 내에 방해를 받았습니다. 이러한 중단은 지속 시간은 최소 10,000밀리초였습니다.
- 통화 시작 후 10%와 80% 사이에 최소 20,000밀리초 동안 침묵이 지속되었습니다.
- 에이전트는 통화 중 어느 시점에서 부정적인 감정을 느꼈습니다.
- 통화의 첫 10,000밀리초 동안 "환영합니다" 또는 "안녕하세요"라는 단어를 사용하지 않았습니다.

```

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)
```

통화 후 분석 범주의 규칙 기준

이 섹션에서는 [CreateCallAnalyticsCategory](#) API 작업을 사용하여 생성할 수 있는 사용자 지정 POST_CALL 규칙의 유형을 간략하게 설명합니다.

중단 일치

중단([InterruptionFilter](#) 데이터 유형)을 사용하는 규칙은 다음과 일치하도록 고안되었습니다.

- 에이전트가 고객의 말을 끊는 경우
- 고객이 에이전트의 말을 끊는 경우
- 상대방의 말을 끊는 모든 참가자
- 중단 없음

[InterruptionFilter](#)로 사용할 수 있는 파라미터의 예는 다음과 같습니다.

```
"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
},
```

이러한 파라미터 및 각 파라미터와 관련된 유효한 값에 대한 자세한 내용은 [CreateCallAnalyticsCategory](#) 및 [InterruptionFilter](#)를 참조하세요.

키워드 일치

키워드([TranscriptFilter](#) 데이터 유형)를 사용하는 규칙은 다음과 일치하도록 고안되었습니다.

- 에이전트, 고객 또는 둘 다 사용하는 사용자 지정 단어나 구절, 또는 둘 다
- 에이전트, 고객 또는 둘 다 사용하지 않는 사용자 지정 단어나 구절, 또는 둘 다

- 특정 시간대에 나오는 사용자 지정 단어 또는 구절

[TranscriptFilter](#)로 사용할 수 있는 파라미터의 예는 다음과 같습니다.

```
"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
}
```

이러한 파라미터 및 각 파라미터와 관련된 유효한 값에 대한 자세한 내용은 [CreateCallAnalyticsCategory](#) 및 [TranscriptFilter](#)를 참조하세요.

침묵 시간 일치

침묵 시간([NonTalkTimeFilter](#) 데이터 유형)을 사용하는 규칙은 다음과 일치하도록 고안되었습니다.

- 통화 중 지정된 시간 동안 무음이 유지되는 경우
- 통화 중 지정된 시간에 음성이 나오는 경우

[NonTalkTimeFilter](#)로 사용할 수 있는 파라미터의 예는 다음과 같습니다.

```
"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
},
```

이러한 파라미터 및 각 파라미터와 관련된 유효한 값에 대한 자세한 내용은 [CreateCallAnalyticsCategory](#) 및 [NonTalkTimeFilter](#)를 참조하세요.

감정 일치

감정 일치([SentimentFilter](#) 데이터 유형)를 사용하는 규칙은 다음과 일치하도록 고안되었습니다.

- 통화의 특정 시점에서 고객, 에이전트 또는 두 사람 모두가 표현한 긍정적인 감정의 유무
- 통화의 특정 시점에서 고객, 에이전트 또는 두 사람 모두가 표현한 부정적인 감정의 유무
- 통화의 특정 시점에서 고객, 에이전트 또는 두 사람 모두가 표현한 중립적인 감정의 유무
- 통화의 특정 시점에서 고객, 에이전트 또는 두 사람 모두가 표현한 복합적인 감정의 유무

[SentimentFilter](#)로 사용할 수 있는 파라미터의 예는 다음과 같습니다.

```
"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 ]
},
```

이러한 파라미터 및 각 파라미터와 관련된 유효한 값에 대한 자세한 내용은 [CreateCallAnalyticsCategory](#) 및 [SentimentFilter](#)를 참조하세요.

통화 후 분석 트랜스크립션 시작

통화 후 분석 트랜스크립션을 시작하기 전에 오디오에서 일치시킬 모든 [범주](#) Amazon Transcribe 를 생성해야 합니다.

 Note

통화 분석 트랜스크립트를 새 범주에 소급 적용할 수는 없습니다. 통화 분석 트랜스크립션을 시작하기 전에 생성한 범주만 트랜스크립션 출력에 적용할 수 있습니다.

범주를 하나 이상 생성했고 오디오가 범주 중 하나 이상의 모든 규칙과 일치하는 경우 Amazon Transcribe에서는 일치하는 범주가 있는 출력에 플래그를 지정합니다. 범주를 사용하지 않기로 선택하거나 오디오가 범주에 지정된 규칙과 일치하지 않는 경우 트랜스크립트에 플래그가 지정되지 않습니다.

통화 후 분석 트랜스크립션을 시작하려면 AWS Management Console, AWS CLI 또는 AWS SDK를 사용할 수 있습니다. 예를 보려면 다음을 참조하세요.

AWS Management Console

다음 절차에 따라 통화 후 분석 작업을 시작합니다. 범주별로 정의된 모든 특성과 일치하는 통화에는 해당 범주로 레이블이 지정됩니다.

1. 탐색 창의 Amazon Transcribe Call Analytics에서 통화 분석 작업을 선택합니다.
2. 작업 생성을 선택합니다.

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

Content removal

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

☒ PII redaction [Info](#)

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

☐ Vocabulary filtering [Info](#)

Vocabulary filtering can remove, mask or tag specified words in the final transcript.

Customization

☐ Custom vocabulary [Info](#)

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

Summarization

☐ Generative call summarization [Info](#)

Generative call summarization provides a summary of the transcript, including important components of the conversation.

Categories

Create categories to classify calls. For example, you can create a category for all cancellation requests. When you run an analytics job, Amazon Transcribe applies that category to all calls that request cancellation.

Call analytics categories (1) [Info](#)

< 1 > 

	Name ▾	Type ▾	Created ▾	Modified ▾
☐	CatchNegativeSentiment	POST_CALL	February 17 2023, 10:43 (UTC-08:00)	February 17 2023, 10:43 (UTC-08:00)

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

Cancel

Previous

Create job

- 작업 세부 정보 지정 페이지에서는 입력 데이터의 위치를 포함하여 통화 분석 작업에 대한 정보를 제공합니다.

Specify job details [Info](#)

Job settings

Name

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

Model type [Info](#)

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

☒ General model

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

☐ Custom language model

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

Language settings

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

☒ Specific language [Info](#)

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

☐ Automatic language identification [Info](#)

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

Language

Choose the language of the input audio.

Input data [Info](#)

Input file location on S3

Choose an input audio or video file in Amazon S3.

[Browse S3](#)

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

Agent audio channel identification [Info](#)

Choose the channel that has the speech from the agent. The other channel is used for the customer's speech.

출력 데이터의 원하는 Amazon S3 위치와 사용할 IAM 역할을 지정합니다.

Output data

Output data location type info [Info](#)

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

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

Access permissions

IAM role [Info](#)

☐ Use an existing IAM role

☒ Create an IAM role
By choosing **Create job** you are authorizing creation of this role.

Permissions to access
Your role has access to these resources. The KMS key permission is used only if your input bucket is encrypted

☐ Input S3 bucket and KMS decrypt permission to input bucket

☒ Any S3 bucket and any KMS keys

Role name
Roles are prefixed with "AmazonTranscribeServiceRoleFullAccess-". Your newly created role has full access to the S3 bucket and KMS key for your account.

The name can be up to 64 characters long

▼ Role permissions details

Your new role has these permissions to give Amazon Transcribe access to the resources that you've specified.

Service	Access level	Resource
S3	List, Read, Write	All resources
Key Management Service	GenerateDataKey, Decrypt	All resources

Cancel
Next

- 다음을 선택합니다.
- 작업 구성에서 통화 분석 작업에 포함하려는 모든 옵션 기능을 복사합니다. 이전에 생성한 범주는 범주 패널에 표시되며 통화 분석 작업에 자동으로 적용됩니다.

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

Content removal

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

☐ PII redaction [Info](#)

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

☐ Vocabulary filtering [Info](#)

Vocabulary filtering can remove, mask or tag specified words in the final transcript.

Customization

☐ Custom vocabulary [Info](#)

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

Summarization

☐ Generative call summarization [Info](#)

Generative call summarization provides a summary of the transcript, including important components of the conversation.

Categories

Create categories to classify calls. For example, you can create a category for all cancellation requests. When you run an analytics job, Amazon Transcribe applies that category to all calls that request cancellation.

Call analytics categories (1) [Info](#)

< 1 > 

	Name ▾	Type ▾	Created ▾	Modified ▾
☐	CatchNegativeSentiment	POST_CALL	February 17 2023, 10:43 (UTC-08:00)	February 17 2023, 10:43 (UTC-08:00)

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

Cancel

Previous

Create job

6. 작업 생성을 선택합니다.

AWS CLI

이 예시에서는 [start-call-analytics-job](#) 명령과 channel-definitions 파라미터를 사용합니다. 자세한 내용은 [StartCallAnalyticsJob](#) 및 [ChannelDefinition](#)을 참조하세요.

```
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
```

다음은 [start-call-analytics-job](#) 명령을 사용하는 또 다른 예 및 해당 작업의 통화 분석을 활성화하는 요청 본문입니다.

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

my-call-analytics-job.json 파일에는 다음과 같은 요청 본문이 포함되어 있습니다.

```
{
  "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 for Python (Boto3)

이 예제에서는 AWS SDK for Python (Boto3) 를 사용하여 [start_call_analytics_job](#) 메서드를 사용하여 Call Analytics 작업을 시작합니다. 자세한 내용은 [StartCallAnalyticsJob](#) 및 [ChannelDefinition](#) 섹션을 참조하세요.

기능별, 시나리오 및 교차 서비스 예제를 포함하여 AWS SDKs를 사용하는 추가 예제는 [AWS SDKs를 사용한 Amazon Transcribe의 코드 예제](#)를 참조하세요.

```
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)
```

통화 후 분석 출력

통화 후 분석 트랜스크립트는 세그먼트별 턴바이턴 형식으로 표시됩니다. 여기에는 통화 분류, 통화 특성(음량 점수, 중단, 침묵 시간, 발연 속도), 통화 요약(문제, 결과 및 조치 항목), 수정 및 감정이 포함됩니다. 또한 대화 특성 요약 정보는 트랜스크립트 끝부분에 있습니다.

정확도를 높이고 산업별 용어 포함 등 사용 사례에 맞게 트랜스크립트를 추가로 사용자 지정하려면 통화 분석 요청에 [사용자 지정 어휘](#) 또는 [사용자 지정 언어 모델](#)을 추가하세요. 트랜스크립션 결과에서 원하지 않는 욕설 등의 단어를 마스킹하거나 삭제하거나 태그를 지정하려면 [어휘 필터링](#)을 추가하세요. 미디어 파일에 전달할 언어 코드를 잘 모르는 경우 [배치 언어 식별](#)을 활성화하여 미디어 파일의 언어를 자동으로 식별할 수 있습니다.

다음 섹션에서는 인사이트 수준에서 JSON 출력의 예를 보여줍니다. 컴파일된 출력은 [컴파일된 통화 후 분석 출력](#) 섹션을 참조하세요.

통화 분류

트랜스크립션 출력에서 범주 일치에 어떻게 보이는지는 다음과 같습니다. 이 예시는 40,040밀리초 타임스탬프에서 42,460밀리초 타임스탬프까지의 오디오가 '포지티브 해상도' 범주와 일치한다는 것을 보여줍니다. 이 경우 사용자 지정 '포지티브 해상도' 범주에서는 마지막 몇 초 동안 긍정적인 감정이 필요했습니다.

```
"Categories": {
  "MatchedDetails": {
    "positive-resolution": {
      "PointsOfInterest": [
        {
          "BeginOffsetMillis": 40040,
          "EndOffsetMillis": 42460
        }
      ]
    }
  },
  "MatchedCategories": [
    "positive-resolution"
  ]
},
```

통화 특성

트랜스크립션 출력의 통화 특성은 다음과 같습니다. 음량 점수는 각 대화 턴에 대해 제공되지만 다른 모든 특성은 자막 끝에 제공됩니다.

```
"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
    }
  },

```

문제, 조치 항목 및 다음 단계

- 다음 예시에서 문제는 문자 7에서 시작하여 문자 51에서 끝나는 것으로 식별되며, 이는 "레시피 구독을 취소하고 싶습니다"라는 텍스트의 이 부분을 나타냅니다.

```

"Content": "Well, I would like to cancel my recipe subscription.",

"IssuesDetected": [
  {
    "CharacterOffsets": {
      "Begin": 7,
      "End": 51
    }
  }
],

```

- 다음 예시에서 결과는 문자 12에서 시작하여 문자 78에서 끝나는 것으로 식별되며, 이는 텍스트에서 "계정을 모두 변경했으며 이제 할인이 적용됩니다" 부분을 나타냅니다.

```
"Content": "Wonderful. I made all changes to your account and now this discount is applied, please check.",
"OutcomesDetected": [
  {
    "CharacterOffsets": {
      "Begin": 12,
      "End": 78
    }
  }
],
```

- 다음 예시에서 작업 항목은 문자 0에서 시작하여 문자 103에서 끝나는 것으로 식별되며, 이는 텍스트에서 "오늘 모든 세부 정보가 포함된 이메일을 보내고 다음 주에 다시 전화하여 후속 조치를 취하겠습니다" 부분을 나타냅니다.

```
"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
    }
  }
],
```

생성형 통화 요약

트랜스크립션 출력에서 생성형 통화 요약은 다음과 같이 표시됩니다.

```
"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."
```

```

    }
  }
}

```

다음과 같은 경우 요약 생성 없이 분석 작업이 완료됩니다.

- 대화 내용 부족: 대화에는 에이전트와 고객 모두의 턴이 하나 이상 포함되어야 합니다. 대화 콘텐츠가 충분하지 않으면 서비스가 오류 코드 `INSUFFICIENT_CONVERSATION_CONTENT`를 반환합니다.
- 안전 가이드라인: 적절한 요약이 생성되도록 대화가 안전 가이드라인을 충족해야 합니다. 이러한 가이드라인이 충족되지 않으면 서비스는 오류 코드 `FAILED_SAFETY_GUIDELINES`를 반환합니다.

오류 코드는 출력의 `AnalyticsJobDetails` 내 `Skipped` 섹션에서 찾을 수 있습니다.

[GetCallAnalyticsJob](#) API 응답의 `CallAnalyticsJobDetails`에서 오류 이유를 찾을 수도 있습니다.

샘플 오류 출력

```

{
  "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",
  ...
}

```

감정 분석

트랜스크립션 출력의 감정 분석은 다음과 같습니다.

- 질적 턴바이턴 감정 값:

```
"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"
```

- 전체 통화의 질적 감정 값:

```
"Sentiment": {
  "OverallSentiment": {
    "AGENT": 2.5,
    "CUSTOMER": 2.1
  },
}
```

- 참가자별 및 통화 분기별 질적 감정 값:

```
"SentimentByPeriod": {
  "QUARTER": {
    "AGENT": [
      {
        "Score": 0.0,
        "BeginOffsetMillis": 0,
        "EndOffsetMillis": 9862
      },
      {
        "Score": -5.0,
        "BeginOffsetMillis": 9862,
```

```

        "EndOffsetMillis": 19725
    },
    {
        "Score": 5.0,
        "BeginOffsetMillis": 19725,
        "EndOffsetMillis": 29587
    },
    {
        "Score": 5.0,
        "BeginOffsetMillis": 29587,
        "EndOffsetMillis": 39450
    }
],
"CUSTOMER": [
    {
        "Score": -2.5,
        "BeginOffsetMillis": 0,
        "EndOffsetMillis": 10615
    },
    {
        "Score": 5.0,
        "BeginOffsetMillis": 10615,
        "EndOffsetMillis": 21230
    },
    {
        "Score": 2.5,
        "BeginOffsetMillis": 21230,
        "EndOffsetMillis": 31845
    },
    {
        "Score": 5.0,
        "BeginOffsetMillis": 31845,
        "EndOffsetMillis": 42460
    }
]
}

```

PII 수정

트랜스크립션 출력의 PII 수정은 다음과 같습니다.


```
"Content": "[PII], my name is [PII], how can I help?",
"Redaction": [{
  "Confidence": "0.9998",
  "Type": "NAME",
  "Category": "PII"
}]
```

자세한 내용은 [배치 작업의 PII 편집](#)을 참조하세요.

언어 식별

이 기능이 활성화된 경우 트랜스크립션 출력의 언어 식별은 다음과 같습니다.

```
"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"
}]
```

위 출력 예시에서 언어 식별은 언어 코드에 신뢰도 점수를 부여합니다. 점수가 가장 높은 결과가 트랜스크립션 언어 코드로 선택됩니다. 모드 세부 정보는 미디어에서 [미디어의 지배적 언어 식별](#)을 참조하세요.

컴파일된 통화 후 분석 출력

다음 트랜스크립션 출력에서는 일부 내용을 간결하게 줄임표로 대체했습니다.

이 샘플에는 선택적 기능 - 생성형 통화 요약이 포함되어 있습니다.

```
{
  "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": []
  },
  ...
}

```

생성형 통화 요약 활성화

Note

Amazon Bedrock:implements 자동 침해 탐지로 구동 AWS 됩니다. <https://docs.aws.amazon.com/bedrock/latest/userguide/abuse-detection.html> 생성형 AI 기반의 고객 응대 후 요약은 Amazon Bedrock을 기반으로 구축되었으므로 사용자는 Amazon Bedrock에 구현된 제어 기능을 최대한 활용하여 안전, 보안 및 인공 지능(AI)의 책임 있는 사용을 강화할 수 있습니다.

통화 후 분석 작업에서 생성형 통화 요약을 사용하려면 다음 예시를 참조하세요.

AWS Management Console

요약 패널에서 생성형 통화 요약을 활성화하면 출력에서 요약을 받을 수 있습니다.

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

Content removal

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

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

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

Customization

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

Summarization

☐ **Generative call summarization** [Info](#)
Generative call summarization provides a summary of the transcript, including important components of the conversation.

Categories

Create categories to classify calls. For example, you can create a category for all cancellation requests. When you run an analytics job, Amazon Transcribe applies that category to all calls that request cancellation.

Call analytics categories (1) [Info](#)

< 1 > 

	Name ▾	Type ▾	Created ▾	Modified ▾
☐	CatchNegativeSentiment	POST_CALL	February 17 2023, 10:43 (UTC-08:00)	February 17 2023, 10:43 (UTC-08:00)

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

Cancel

Previous

Create job

AWS CLI

이 예시에서는 [start-call-analytics-job](#) 명령과 Settings 파라미터 및 Summarization 하위 파라미터를 사용합니다. 자세한 내용은 [StartCallAnalyticsJob](#) 단원을 참조하십시오.

```
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}}'
```

다음은 [start-call-analytics-job](#) 명령을 사용하는 또 다른 예 및 해당 작업의 요약을 활성화하는 요청 본문입니다.

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

my-call-analytics-job.json 파일에는 다음과 같은 요청 본문이 포함되어 있습니다.

```
{
  "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 for Python (Boto3)

이 예제에서는 AWS SDK for Python (Boto3) 를 사용하여 [start_call_analytics_job](#) 메서드를 사용하여 요약이 활성화된 Call Analytics를 시작합니다. 자세한 내용은 [StartCallAnalyticsJob](#) 단원을 참조하십시오.

기능별, 시나리오 및 교차 서비스 예제를 포함하여 AWS SDKs를 사용하는 추가 예제는 [AWS SDKs를 사용한 Amazon Transcribe의 코드 예제](#)장을 참조하세요.

```

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)

```

실시간 통화 분석

실시간 통화 분석은 문제를 해결하고 에스컬레이션 발생 시 이를 완화하는 데 사용할 수 있는 실시간 인사이트를 제공합니다.

실시간 통화 분석을 통해 다음과 같은 인사이트를 얻을 수 있습니다.

- 규칙을 사용하여 특정 키워드와 구절에 플래그를 지정하는 [범주 이벤트](#). 범주 이벤트는 [실시간 알림](#)을 생성하는 데 사용할 수 있습니다.
- [문제 감지](#)는 각 오디오 세그먼트 내에서 제기된 문제를 식별합니다.
- 텍스트 트랜스크립트의 [PII\(민감한 데이터\) 식별](#)
- 텍스트 트랜스크립트의 [PII\(민감한 데이터\) 수정](#)
- 각 음성 세그먼트에 대한 [감정 분석](#)
- [언어 식별](#)은 각 오디오 채널에서 사용되는 기본 언어를 감지합니다.

실시간 Call Analytics 외에도 Amazon Transcribe 는 미디어 스트림에서 [통화 후 분석](#)을 수행할 수 있습니다. [PostCallAnalyticsSettings](#) 파라미터를 사용하여 실시간 통화 분석 요청에 통화 후 분석을 포함할 수 있습니다.

실시간 인사이트

이 섹션에서는 실시간 통화 분석 트랜스크립션에 사용할 수 있는 인사이트를 자세히 설명합니다.

범주 이벤트

범주 이벤트를 사용하면 정확한 키워드나 구절을 기반으로 트랜스크립션을 일치시킬 수 있습니다. 예를 들어 "I want to speak to the manager"라는 문구에 대해 필터를 설정하면 해당 정확한 문구에 대해 Amazon Transcribe 필터링합니다.

다음은 [출력 예](#)입니다.

실시간 통화 분석 범주를 만드는 방법에 대한 자세한 내용은 [실시간 트랜스크립션의 범주 생성](#) 섹션을 참조하세요.

Tip

범주 이벤트를 통해 실시간 알림을 설정할 수 있습니다. 자세한 내용은 [범주 일치에 대한 실시간 알림 생성](#) 섹션을 참조하세요.

문제 감지

문제 감지는 각 오디오 세그먼트 내에서 감지된 문제에 대한 간결한 요약を提供합니다. 문제 감지 기능을 사용하면 다음과 같은 이점이 있습니다.

- 통화 도중과 통화 후에 수동으로 메모를 할 필요가 줄어듭니다.
- 에이전트의 효율성이 개선되므로 에이전트가 고객에게 더 빠르게 응답할 수 있습니다.

Note

문제 감지에는 호주(en-AU), 영국(en-GB), 미국(en-US) 등의 영어 방언이 지원됩니다.

문제 감지 기능은 모든 산업 및 비즈니스 부문에서 작동하며 컨텍스트에 따라 달라집니다. 문제 감지는 기본적으로 작동하므로 모델 교육 또는 사용자 지정 범주와 같은 사용자 지정을 지원하지 않습니다.

실시간 통화 분석을 통한 문제 감지는 각각의 오디오 세그먼트 전체에서 이루어집니다.

다음은 [출력 예](#)입니다.

PII(민감한 데이터) 식별

민감한 데이터 식별은 텍스트 트랜스크립트의 개인 식별 정보(PII)에 레이블을 지정합니다. 이 파라미터는 고객 정보를 보호하는 데 유용합니다.

Note

실시간 PII 식별에는 호주(en-AU), 영국(en-GB), 미국(en-US) 등의 영어 방언과 스페인어 방언(es-US)이 지원됩니다.

실시간 통화 분석을 통한 PII 식별은 각각의 오디오 세그먼트 전체에서 이루어집니다.

이 기능을 사용하여 식별되는 PII 목록을 보거나 사용한 PII 식별에 대해 자세히 알아보려면 섹션을 [Amazon Transcribe참조하세요](#) [개인 식별 정보 편집 또는 식별](#).

다음은 [출력 예](#)입니다.

PII(민감한 데이터) 수정

민감한 데이터 수정은 텍스트 트랜스크립트의 개인 식별 정보(PII)를 PII 유형(예: [NAME])으로 대체합니다. 이 파라미터는 고객 정보를 보호하는 데 유용합니다.

Note

실시간 PII 수정에는 호주(en-AU), 영국(en-GB), 미국(en-US) 등의 영어 방언과 스페인어 방언(es-US)이 지원됩니다.

실시간 통화 분석을 통한 PII 수정은 각각의 오디오 세그먼트 전체에서 이루어집니다.

이 기능을 사용하여 수정된 PII 목록을 보거나 Amazon Transcribe를 사용한 수정에 대해 자세히 알아보려면 [개인 식별 정보 편집 또는 식별](#) 섹션을 참조하세요.

다음은 [출력 예](#)입니다.

감정 분석

감정 분석은 통화 내내 고객과 에이전트가 느끼는 감정을 추정합니다. 이 지표는 모든 음성 세그먼트에 대해 제공되며 질적 값(positive, neutral, mixed 또는negative)으로 표시됩니다.

이 파라미터를 사용하면 각 통화 참가자의 전반적인 감정과 각 음성 세그먼트 내 각 참가자의 감정을 질적으로 평가할 수 있습니다. 이 지표는 통화가 끝날 때까지 에이전트가 화가 난 고객을 기쁘게 할 수 있는지 파악하는 데 도움이 될 수 있습니다.

실시간 통화 분석을 통한 감정 분석은 각각의 오디오 세그먼트 전체에서 이루어집니다.

감정 분석은 기본적으로 작동하므로 모델 교육 또는 사용자 지정 범주와 같은 사용자 지정을 지원하지 않습니다.

다음은 [출력 예](#)입니다.

언어 식별

언어 식별은 실시간 통화 분석 중에 오디오 스트림의 각 채널 내에서 사용되는 기본 언어를 자동으로 인식하고 결정합니다. 식별되면 통화 분석은 감지된 언어를 기반으로 가장 적합한 트랜스크립션을 처리하고 반환하여, 이 정보를 스트림을 통해 실시간으로 다시 전송합니다.

이 기능을 사용하면 오디오 스트림의 각 채널에서 사용되는 지배적 언어를 자동으로 인식하고 식별할 수 있습니다. 언어가 감지되면 통화 분석은 식별된 언어에 대한 적절한 트랜스크립션을 실시간으로 처리하고 전달합니다.

자동 언어 식별은 현재 스트리밍 트랜스크립션에 대해 추가 비용 없이 지원되는 모든 [통화 분석 스트리밍 지원 언어](#)에 대해 지원되며 통화 분석 스트리밍 지원 [AWS 리전](#)에서 사용할 수 있습니다.

Important

통화 분석은 오디오 채널에서 사용되는 지배적 언어를 식별하는 단일 언어 식별만 지원합니다. 다국어 식별은 지원되지 않습니다. 즉, 각 채널은 한 언어로만 트랜스크립션할 수 있습니다. 스트리밍 언어 식별을 사용하려면 언어 코드를 두 개 이상 최대 다섯 개까지 제공해야 하며 한 스트림당 언어당 [지원 통화 분석 스트리밍 언어](#) 중 하나의 언어 방언만 선택할 수 있습니다. 즉, 동일한 트랜스크립션에 대해 en-US 및 en-AU를 언어 옵션으로 선택할 수 없습니다. LanguageCode와 IdentifyLanguage는 상호 배타적인 옵션이므로, 이 기능을 사용할 때 LanguageCode 파라미터는 요청에서 null로 유지되어야 합니다.

Warning

지정된 언어 코드가 실제 음성 언어와 일치하지 않는 경우 시스템에서 옵션에서 가장 유사한 언어를 선택하여 트랜스크립션이 부정확해질 수 있습니다.

언어 식별 기능을 사용하여 다음을 수행할 수 있습니다.

- 실시간으로 지배적 언어 자동 감지
- 별도의 채널에서 다양한 언어 처리
- 언어 감지에 대한 신뢰도 점수 받기
- 언어별 사용자 지정 어휘 적용

언어 식별을 사용하려면 다음 파라미터를 구성해야 합니다.

필수 파라미터:

- `identifyLanguage` - 언어 식별을 활성화하려면 `true`로 설정합니다.
- `languageOptions` - `identifyLanguage`가 `true`로 설정된 경우 사용할 수 있는 언어 코드 목록입니다. 단일 언어 선택은 지원되지 않으므로 최소 두 개의 언어 선택을 제공해야 합니다.

선택적 파라미터:

- `preferredLanguage` - 제공된 `languageOptions`의 예상 기본 언어입니다. 기본 언어를 추가하면 통화 분석이 언어를 더 빠르게 식별하는 데 도움이 될 수 있습니다.
- `vocabularyNames` - 정확도 향상을 위한 사용자 지정 어휘 이름입니다. 어휘 이름은 대소문자를 구분하며, 사용자 지정 어휘의 언어가 식별된 미디어 언어와 일치하지 않는 경우 트랜스크립션에 적용되지 않습니다.
- `vocabularyFilterNames` - 트랜스크립트 출력을 사용자 지정하는 어휘 필터 이름입니다.

다음은 [출력 예](#)입니다.

실시간 트랜스크립션의 범주 생성

실시간 통화 분석은 사용자 지정 범주 생성을 지원하므로 특정 비즈니스 요구 사항에 가장 적합하도록 트랜스크립트 분석을 조정할 수 있습니다.

다양한 시나리오를 다루기 위해 원하는 만큼 범주를 생성할 수 있습니다. 범주를 하나 생성할 때마다 1~20개의 규칙을 생성해야 합니다. 실시간 통화 분석 트랜스크립션은 [TranscriptFilter](#)(키워드 일치)를 사용하는 규칙만 지원합니다. [CreateCallAnalyticsCategory](#) 작업에 규칙을 사용하는 방법에 대한 자세한 내용은 [실시간 통화 분석 범주의 규칙 기준](#) 섹션을 참조하세요.

미디어의 콘텐츠가 특정 범주에서 지정한 모든 규칙과 일치하는 경우 Amazon Transcribe에서는 해당 범주로 출력에 레이블을 지정합니다. JSON 출력 형식의 범주 일치 예는 [범주 이벤트 출력](#)을 참조하세요.

사용자 지정 범주로 수행할 수 있는 작업의 몇 가지 예는 다음과 같습니다.

- 특정 키워드 세트에 플래그를 지정하고 추적하여 즉각적인 주의가 필요한 문제를 식별합니다.
- 에이전트가 특정 구절을 말하거나 생략하는 등, 규정 준수를 모니터링합니다.
- 특정 단어와 구절에 실시간으로 플래그를 지정한 다음 범주 일치를 설정하여 즉시 알림을 설정할 수 있습니다. 예를 들어 "매니저와 대화"라는 구절을 사용하는 고객의 경우에 대한 실시간 통화 분석 범주를 생성한 경우 이 실시간 범주 일치에 대한 [이벤트 알림](#)을 설정하여 듀티 매니저에게 알릴 수 있습니다.

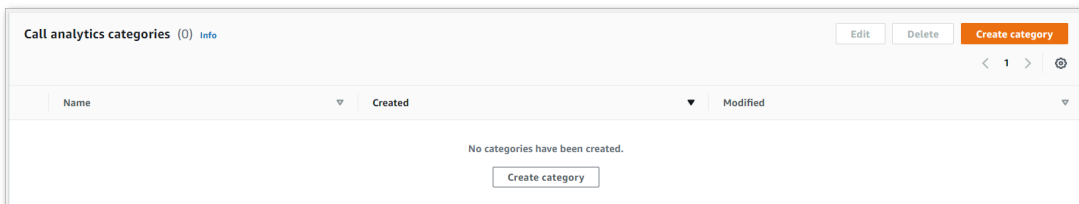
통화 후 범주와 실시간 범주의 비교

새 범주를 만들 때 해당 범주를 통화 후 범주(POST_CALL)로 생성할지 아니면 실시간 범주 (REAL_TIME)로 생성할지 지정할 수 있습니다. 옵션을 지정하지 않으면 기본적으로 범주가 통화 후 범주로 생성됩니다. 실시간 범주 일치를 사용하여 실시간 알림을 생성할 수 있습니다. 자세한 내용은 [범주 일치에 대한 실시간 알림 생성](#) 단원을 참조하십시오.

실시간 통화 분석을 위한 새 범주를 생성하려면 AWS Management Console, AWS CLI 또는 AWS SDK를 사용할 수 있습니다. 예를 보려면 다음을 참조하세요.

AWS Management Console

- 탐색 창의 아래에서 Amazon Transcribe Call Analytics를 Amazon Transcribe 선택합니다.
- 통화 분석 범주를 선택하면 통화 분석 범주 페이지로 이동합니다. 범주 생성 버튼을 선택합니다.



- 이제 범주 생성 페이지로 이동했습니다. 범주 이름을 입력한 다음 범주 유형 드롭다운 메뉴에서 '실시간 통화 분석'을 선택합니다.

Category settings

Category name

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

Category type [Info](#)

Choose category type ▲

Batch call analytics

Real time call analytics

☒ Use a template (recommended)
Use a template to edit predefined rules.

☐ Create from scratch
If you know the rules that you want to define, choose this option.

Template type [Info](#)

Choose the template for the category that most closely matches the one you want to create.

Choose a template ▼

4. 템플릿을 선택하여 범주를 생성하거나 처음부터 새로 생성할 수 있습니다.

템플릿을 사용하는 경우: 템플릿 사용(권장)을 선택하고 원하는 템플릿과 범주 생성을 차례로 선택합니다.

Category settings

Category name

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

Category type [Info](#)

Real time call analytics ▼

Category creation method [Info](#)

☒ Use a template (recommended)
Use a template to edit predefined rules.

☐ Create from scratch
If you know the rules that you want to define, choose this option.

Template type [Info](#)

Choose the template for the category that most closely matches the one you want to create.

Choose a template ▲

Customer content is negative and mentioned manager

5. 사용자 지정 범주를 생성하는 경우: 새로 생성을 선택합니다.

Create category [Info](#)

Category settings

Category name

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

Category creation method [Info](#)

☐ Use a template (recommended)
Use a template to edit predefined rules.

☒ Create from scratch
If you know the rules that you want to define, choose this option.

Rules

All the rule conditions must be met for a transcription job to be classified in this category.

▼ Rule 1 Delete rule

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

Add rule

You can add up to 19 more rules.

- 드롭다운 메뉴를 사용하여 범주에 규칙을 추가합니다. 범주당 최대 20개의 규칙을 추가할 수 있습니다. 실시간 통화 분석 트랜스크립션에서는 트랜스크립트 콘텐츠 일치에 수반되는 규칙만 포함할 수 있습니다. 모든 일치에는 실시간으로 플래그가 지정됩니다.

Rules

All the rule conditions must be met for a transcription job to be classified in this category.

▼ Rule 1 Delete rule

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

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

Add rule

You can add up to 19 more rules.

- 다음은 한 가지 규칙이 있는 범주의 예입니다. 바로, 통화 중 언제라도 "매니저와 대화"하는 구절을 사용하는 고객의 경우입니다.

Rules
All the rule conditions must be met for a transcription job to be classified in this category.

▼ **Rule 1** Delete rule

When any of the words were mentioned during the entire call when the speaker was customer.

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

Transcript content match ▼

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

When any of the words were mentioned ▼

during the entire call ▼

when the speaker was customer ▼

Words or phrases [Info](#)
Enter the words or phrases that you want to look for in the transcript. You can enter up to 100 words or phrases.

speak to a manager Add a new word or phrase

The word or phrase can be up to 2,000 characters.

Add rule

You can add up to 19 more rules.

8. 범주에 규칙을 모두 추가했으면 범주 생성을 선택합니다.

AWS CLI

이 예에서는 [create-call-analytics-category](#) 명령을 사용합니다. 자세한 내용은 [CreateCallAnalyticsCategory](#), [CategoryProperties](#), [Rule](#) 섹션을 참조하세요.

다음 예에서는 규칙을 사용하여 범주를 생성합니다.

- 고객은 통화 중 어느 시점에서 "매니저와 대화"라는 구절을 사용했습니다.

이 예에서는 [create-call-analytics-category](#) 명령과 해당 범주에 규칙을 추가하는 요청 본문을 사용합니다.

```
aws transcribe create-call-analytics-category \
--cli-input-json file:///filepath/my-first-analytics-category.json
```

my-first-analytics-category.json 파일에는 다음과 같은 요청 본문이 포함되어 있습니다.

```
{
  "CategoryName": "my-new-real-time-category",
```

```

"InputType": "REAL_TIME",
"Rules": [
    {
        "TranscriptFilter": {
            "Negate": false,
            "Targets": [
                "speak to the manager"
            ],
            "TranscriptFilterType": "EXACT"
        }
    }
]
}

```

AWS SDK for Python (Boto3)

이 예제에서는 AWS SDK for Python (Boto3) 를 사용하여 [create_call_analytics_category](#) 메서드에 대한 CategoryName 및 Rules 인수를 사용하여 범주를 생성합니다. 자세한 내용은 [CreateCallAnalyticsCategory](#), [CategoryProperties](#), [Rule](#) 섹션을 참조하세요.

기능별, 시나리오 및 교차 서비스 예제 AWS SDKs를 사용하는 추가 예제는 [AWS SDKs를 사용한 Amazon Transcribe의 코드 예제](#)장을 참조하세요.

다음 예에서는 규칙을 사용하여 범주를 생성합니다.

- 고객은 통화 중 어느 시점에서 "매니저와 대화"라는 구절을 사용했습니다.

```

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)

```

실시간 통화 분석 범주의 규칙 기준

이 섹션에서는 [CreateCallAnalyticsCategory](#) API 작업을 사용하여 생성할 수 있는 사용자 지정 REAL_TIME 규칙의 유형을 간략하게 설명합니다.

문제 감지는 자동으로 수행되므로 문제에 플래그를 지정하기 위해 규칙이나 범주를 생성할 필요가 없습니다.

실시간 통화 분석 트랜스크립션에는 키워드 일치만 지원됩니다. 중단, 침묵 또는 감정이 포함된 범주를 생성하려면 [통화 후 분석 범주의 규칙 기준](#) 섹션을 참조하세요.

키워드 일치

키워드([TranscriptFilter](#) 데이터 유형)를 사용하는 규칙은 다음과 일치하도록 고안되었습니다.

- 에이전트, 고객 또는 둘 다 사용하는 사용자 지정 단어나 구절, 또는 둘 다
- 에이전트, 고객 또는 둘 다 사용하지 않는 사용자 지정 단어나 구절, 또는 둘 다
- 특정 시간대에 나오는 사용자 지정 단어 또는 구절

[TranscriptFilter](#)로 사용할 수 있는 파라미터의 예는 다음과 같습니다.

```

"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
}
```

이러한 파라미터 및 각 파라미터와 관련된 유효한 값에 대한 자세한 내용은 [CreateCallAnalyticsCategory](#) 및 [TranscriptFilter](#)를 참조하세요.

실시간 트랜스크립션을 통한 통화 후 분석

통화 후 분석은 실시간 통화 분석 트랜스크립션과 함께 사용할 수 있는 선택적 기능입니다. 표준 [실시간 분석 인사이트](#) 외에도 통화 후 분석은 다음을 제공합니다.

- 작업 항목: 통화에서 확인된 모든 작업 항목을 나열합니다.
- 중단: 문장 중간에 한 참가자가 다른 참가자의 말을 중간에 끊는지 여부와 그 시점을 측정합니다.
- 문제: 통화에서 확인된 문제를 알려줍니다.
- 음량: 각 참가자가 말하는 음량을 측정합니다.
- 침묵 시간: 음성이 포함되지 않은 시간을 측정합니다.
- 결과: 통화에서 확인된 결과 또는 해결 방법을 제시합니다.
- 발연 속도: 두 참가자가 말하는 속도를 측정합니다.
- 발연 시간: 통화 중 각 참가자가 발연한 시간(밀리초)을 측정합니다.

활성화되면 오디오 스트림의 통화 후 분석은 [오디오 파일의 통화 후 분석](#)과 유사한 트랜스크립트를 생성하여에 지정된 Amazon S3 버킷에 저장합니다OutputLocation. 또한 통화 후 분석은 오디오 스트림을 기록하고 동일한 Amazon S3 버킷에 오디오 파일(WAV 형식)로 저장합니다. 수정을 활성화하면 수정된 트랜스크립트와 수정된 오디오 파일도 지정된 Amazon S3 버킷에 저장됩니다. 오디오 스트림으로 통화 후 분석을 활성화하면 다음과 같이 2~4개의 파일이 생성됩니다.

- 수정을 활성화하지 않은 경우 출력 파일은 다음과 같습니다.
 1. 수정되지 않은 트랜스크립트
 2. 수정되지 않은 오디오 파일
- 수정되지 않은 옵션(redacted) 없이 수정을 활성화한 경우 출력 파일은 다음과 같습니다.
 1. 수정된 트랜스크립트
 2. 수정된 오디오 파일
- 수정되지 않은 옵션(redacted_and_unredacted)을 사용하여 수정을 활성화한 경우 출력 파일은 다음과 같습니다.

1. 수정된 트랜스크립트
2. 수정된 오디오 파일
3. 수정되지 않은 트랜스크립트
4. 수정되지 않은 오디오 파일

참고: 요청과 함께 통화 후 분석([PostCallAnalyticsSettings](#))을 활성화하고 FLAC 또는 OPUS-0GG 미디어를 사용하는 경우 트랜스크립트에 loudnessScore가 포함되지 않으며 스트림의 오디오 녹음도 생성되지 않습니다. Transcribe는 90분 이상 지속되는 장기 실행 오디오 스트림에 대한 통화 후 분석을 제공하지 못할 수도 있습니다.

오디오 스트림의 통화 후 분석에서 사용할 수 있는 인사이트에 대한 자세한 내용은 [통화 후 분석 인사이트](#) 섹션을 참조하세요.

Tip

실시간 통화 분석 요청으로 통화 후 분석을 활성화하면 모든 POST_CALL 및 REAL-TIME 범주가 통화 후 분석 트랜스크립터에 적용됩니다.

통화 후 분석 활성화

통화 후 분석을 활성화하려면 실시간 통화 분석 요청에 [PostCallAnalyticsSettings](#) 파라미터를 포함해야 합니다. PostCallAnalyticsSettings를 활성화할 때는 다음 파라미터를 포함해야 합니다.

- OutputLocation: 통화 후 트랜스크립트를 저장할 Amazon S3 버킷입니다.
- DataAccessRoleArn: 지정된 Amazon S3 버킷에 액세스할 권한이 있는 Amazon S3 역할의 Amazon 리소스 이름(ARN). [실시간 분석에 대한 신뢰 정책](#)도 사용해야 합니다.

수정된 버전의 트랜스크립트가 필요한 경우 요청에 ContentRedactionOutput 또는 ContentRedactionType를 포함시킬 수 있습니다. 이러한 파라미터에 대한 자세한 내용은 API 참조의 [StartCallAnalyticsStreamTranscription](#)을 참조하세요.

통화 후 분석을 활성화한 상태에서 실시간 통화 분석 트랜스크립션을 시작하려면 AWS Management Console(데모만 해당), HTTP/2 또는 WebSocket을 사용할 수 있습니다. 예시는 [실시간 통화 분석 트랜스크립션 시작](#) 섹션을 참조하세요.

⚠ Important

현재는 오디오 예제가 사전 로드된 실시간 Call Analytics 데모 AWS Management Console 만 제공합니다. 자체 오디오를 사용하려면 API(HTTP/2, WebSocket 또는 SDK)를 사용해야 합니다.

통화 후 분석 출력 예시

통화 후 트랜스크립트는 세그먼트별 턴바이턴 형식으로 표시됩니다. 여기에는 통화 특성, 감정, 통화 요약, 문제 감지 및 PII 수정(선택 사항)이 포함됩니다. 통화 후 범주가 오디오 콘텐츠와 일치하는 경우 해당 범주도 출력에 표시됩니다.

정확도를 높이고 산업별 용어 포함 등 사용 사례에 맞게 트랜스크립트를 추가로 사용자 지정하려면 통화 분석 요청에 [사용자 지정 어휘](#) 또는 [사용자 지정 언어 모델](#)을 추가하세요. 트랜스크립션 결과에서 원하지 않는 욕설 등의 단어를 마스킹하거나 삭제하거나 태그를 지정하려면 [어휘 필터링](#)을 추가하세요.

다음은 컴파일된 통화 후 분석 출력 예입니다.

```
{
  "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
  }
},
...
}

```

실시간 통화 분석 트랜스크립션 시작

실시간 Call Analytics 트랜스크립션을 시작하기 전에 통화에서 일치시킬 모든 [범주](#) Amazon Transcribe 를 생성해야 합니다.

Note

통화 분석 트랜스크립트를 새 범주에 소급 적용할 수는 없습니다. 통화 분석 트랜스크립션을 시작하기 전에 생성한 범주만 트랜스크립션 출력에 적용할 수 있습니다.

범주를 하나 이상 생성했고 오디오가 범주 중 하나 이상의 모든 규칙과 일치하는 경우 Amazon Transcribe 에서는 일치하는 범주가 있는 출력에 플래그를 지정합니다. 범주를 사용하지 않기로 선택하거나 오디오가 범주에 지정된 규칙과 일치하지 않는 경우 트랜스크립트에 플래그가 지정되지 않습니다.

실시간 통화 분석 트랜스크립션에 통화 후 분석을 포함하려면 `OutputLocation` 파라미터를 사용하여 요청에 Amazon S3 버킷을 제공해야 합니다. 또한 지정된 버킷에 대한 쓰기 권한이 있는 `DataAccessRoleArn`도 포함해야 합니다. 실시간 통화 분석 스트리밍 세션이 완료되면 별도의 트랜스크립트가 생성되어 지정된 버킷에 저장됩니다.

실시간 통화 분석을 사용하면 실시간 범주 알림을 생성할 수도 있습니다. 지침은 [범주 일치에 대한 실시간 알림 생성](#) 섹션을 참조하세요.

실시간 통화 분석 트랜스크립션을 시작하려면 AWS Management Console, HTTP/2 또는 WebSocket을 사용할 수 있습니다. 예를 보려면 다음을 참조하세요.

⚠ Important

현재는 오디오 예제가 사전 로드된 실시간 Call Analytics 데모 AWS Management Console 만 제공합니다. 자체 오디오를 사용하려면 API(HTTP/2, WebSocket 또는 SDK)를 사용해야 합니다.

AWS Management Console

다음 절차에 따라 통화 분석 요청을 시작합니다. 범주별로 정의된 모든 특성과 일치하는 통화에는 해당 범주로 레이블이 지정됩니다.

i Note

AWS Management Console에서는 데모만 사용할 수 있습니다. 사용자 지정 실시간 분석 트랜스크립션을 시작하려면 [API](#)를 사용해야 합니다.

1. 탐색 창의 Amazon Transcribe Call Analytics에서 실시간 통화 분석을 선택합니다.

Amazon Transcribe > Real-time Analytics

Real-time Analytics info

Transcribe Real-time Call Analytics combines powerful speech-to-text and natural language processing (NLP) models that are trained specifically to understand customer service and sales calls. With Transcribe Call Analytics, developers can get a redacted and unredacted transcript, and insights such as customer and agent sentiment, detected issues, and supervisor alerts during the live call.

How it works
This demo experience has been configured to use preloaded audio examples of customer-agent interactions. Before starting the demo, you can optionally create categories in the Category Management page and update content redaction settings under the advance settings



Step 1: Specify input audio

Input audio file

Insurance complaints (en-US)

00:00/00:00



Step 2: Review call categories - optional

Categorize your calls based on custom keywords or phrases.

View categories



Step 3: Configure output - optional

Apply content redaction settings to your calls.

Configure advanced settings

Post-call Analytics
Post-call analytics enabled with real-time analytics provides consolidated transcript and audio backup, with the associated analytics, along with further insights such as call summaries and conversation characteristics like non-talk time, interruptions, loudness, and talk speed, after the end of the call in the provided Amazon S3 bucket.

☐ Post-call Analytics

Start streaming

2. 1단계: 입력 오디오 지정에서는 드롭다운 메뉴에서 데모 테스트 파일을 선택합니다.



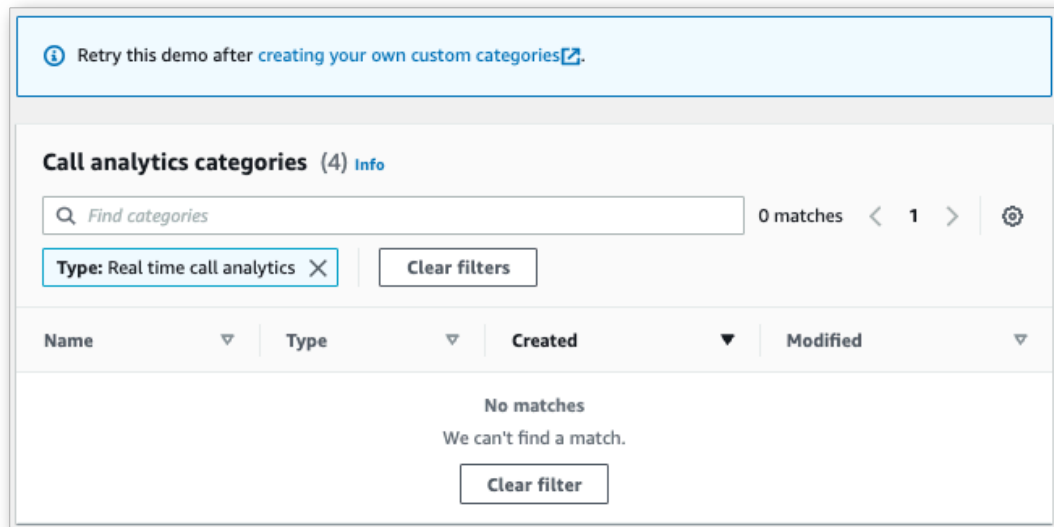
Step 1: Specify input audio

Input audio file

Insurance complaints (en-US)	▲
Insurance complaints (en-US)	✓
Hospitality complaints (en-US)	

- 2단계: 통화 범주 검토에서는 이전에 만든 실시간 통화 분석 범주를 검토할 수 있습니다. 모든 실시간 통화 분석 범주가 트랜스크립션에 적용됩니다.

범주 보기를 선택하면 새 창이 열리면서 기존 실시간 통화 분석 범주를 표시하고 새 범주를 생성할 수 있는 링크를 제공합니다.



- 3단계: 입력 및 출력 구성에서는 추가 설정을 적용할 수 있습니다.

고급 설정 구성을 선택하면 콘텐츠 수정 설정을 지정할 수 있는 새 창이 열립니다.

Use the following options to identify or redact content from your transcript. Other settings such as Custom Vocabulary, Custom Language Models, Partial results stabilization, Vocabulary Filtering are available through the API, SDK, CLI

▼ Content removal

☒ PII Identification & redaction [Info](#)

Identify or redact one or more types of personally identifiable information (PII) in your transcript

Select PII detection type

☒ Identification only

Label the type of PII identified but not redact it in the transcription output

☐ Identification & redaction

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

Select PII entity types (11 of 11 selected)

☒ Select All

☒ Financial (6 of 6 selected)

☒ BANK_ACCOUNT_NUMBER

☒ CREDIT_DEBIT_CVV

☒ BANK_ROUTING

☒ CREDIT_DEBIT_EXPIRY

☒ CREDIT_DEBIT_NUMBER

☒ PIN

☒ Personal (5 of 5 selected)

☒ NAME

☒ EMAIL

☒ ADDRESS

☒ SSN

☒ PHONE

The updates that you make here will only be applied when you start stream again.

Cancel

Save

선택을 모두 마쳤으면 저장을 선택하여 메인 페이지로 돌아갑니다.

- 추가 분석을 적용하려면 통화 후 분석을 켜면 됩니다. 이를 통해 중단, 음량, 침묵 시간, 발언 속도, 발언 시간, 문제, 조치 항목 및 결과를 포함하여 통화 후 분석 트랜스크립션과 동일한 분석을 얻을 수 있습니다. 통화 후 분석 출력은 실시간 통화 분석 트랜스크립션과 별도의 파일에 저장됩니다.

Post-call Analytics

Post-call analytics enabled with real-time analytics provides consolidated transcript and audio backup, with the associated analytics, along with further insights such as call summaries and conversation characteristics like non-talk time, interruptions, loudness, and talk speed, after the end of the call in the provided Amazon S3 bucket.

☐ Post-call Analytics

통화 후 분석을 적용하는 경우 Amazon S3 출력 파일 대상과 IAM 역할을 지정해야 합니다. 출력을 암호화하도록 선택할 수 있습니다.

Post-call Analytics
Post-call analytics enabled with real-time analytics provides consolidated transcript and audio backup, with the associated analytics, along with further insights such as call summaries and conversation characteristics like non-talk time, interruptions, loudness, and talk speed, after the end of the call in the provided Amazon S3 bucket.

☒ Post-call Analytics

Output file destination on S3 [info](#)
Choose the location to store the output of the post-call analytics. If you input a location in an Amazon S3 bucket that doesn't yet exist, it will be created for you.

Resource URI
 [View](#) [Browse S3](#)

Format: s3://bucket, s3://bucket/prefix/, or s3://bucket/prefix/object.

☐ Encryption [info](#)

IAM role [info](#)
[Create an IAM role](#) that grants access to the output bucket and KMS key (if specified) with the trust policy shown below
 ▶ Trust Policy

[Choose a role](#)

[Start streaming](#)

6. 스트리밍 시작을 선택합니다.

HTTP/2 스트림

이 예시에서는 통화 분석이 활성화된 상태에서 HTTP/2 요청을 생성합니다. 에서 HTTP/2 스트리밍을 사용하는 방법에 대한 자세한 내용은 섹션을 Amazon Transcribe참조하세요[HTTP/2 스트림 설정](#). 관련 파라미터 및 헤더에 대한 자세한 내용은 섹션을 Amazon Transcribe참조하세요[StartCallAnalyticsStreamTranscription](#).

이 예에는 [통화 후 분석](#)이 포함됩니다. 통화 후 분석을 원하지 않는 경우 요청에서 해당 PostCallAnalyticsSettings 섹션을 삭제하세요.

다음 예시에 표시된 구성 이벤트는 스트림의 첫 번째 이벤트로 전달해야 합니다.

```
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"
      }
    }
  }
}

```

파라미터 정의는 [API 참조](#)에서 찾을 수 있습니다. 모든 AWS API 작업에 공통적인 파라미터는 [공통 파라미터](#) 섹션에 나열되어 있습니다.

WebSocket 스트림

이 예에서는 WebSocket 스트림에서 통화 분석을 사용하는 미리 서명된 URL을 만듭니다. 가독성을 높이기 위해 줄바꿈이 추가되었습니다. Amazon Transcribe로 WebSocket 스트림을 사용하는 방법에 대한 자세한 내용은 [WebSocket 스트림 설정](#)을 참조하세요. 파라미터에 대한 자세한 내용은 [StartCallAnalyticsStreamTranscription](#)을 참조하세요.

이 예에는 [통화 후 분석](#)이 포함됩니다. 통화 후 분석을 원하지 않는 경우 요청에서 해당 `PostCallAnalyticsSettings` 섹션을 삭제하세요.

다음 예시에 표시된 구성 이벤트는 스트림의 첫 번째 이벤트로 전달해야 합니다.

```

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"
      }
    }
  }
}

```

파라미터 정의는 [API 참조](#)에서 찾을 수 있습니다. 모든 AWS API 작업에 공통적인 파라미터는 [공통 파라미터](#) 섹션에 나열되어 있습니다.

Tip

위의 HTTP/2 및 WebSocket 예시에는 통화 후 분석이 포함됩니다. 통화 후 분석을 원하지 않는 경우 요청에서 해당 PostCallAnalyticsSettings 섹션을 삭제하세요. PostCallAnalyticsSettings를 활성화한 경우 구성 이벤트를 첫 번째 이벤트로 전송해야 합니다. 구성 이벤트에는 이전 예와 같이 ChannelDefinitions 및 PostStreamAnalyticsSettings에 대한 설정이 포함됩니다.

이진 데이터는 content-type application/octet-stream과 함께 이진 메시지로 전달되고 구성 이벤트는 content-type application/json과 함께 텍스트 메시지로 전달됩니다.

자세한 내용은 [스트리밍 트랜스크립션 설정](#) 단원을 참조하십시오.

범주 일치에 대한 실시간 알림 생성

실시간 알림을 설정하려면 먼저 REAL_TIME 플래그가 있는 [TranscriptFilterType](#) 범주를 생성해야 합니다. 이 플래그를 사용하면 범주를 실시간 통화 분석 트랜스크립션에 적용할 수 있습니다.

새 범주 생성에 대한 지침은 [실시간 트랜스크립션의 범주 생성](#) 섹션을 참조하세요.

실시간 통화 분석 트랜스크립션을 시작하면 REAL_TIME 플래그가 있는 모든 범주가 세그먼트 수준에서 트랜스크립션 출력에 자동으로 적용됩니다. TranscriptFilterType 일치가 발생하면 트랜스크립트의 CategoryEvent 섹션 아래에 표시됩니다. 그런 다음 이 파라미터와 해당 하위 파라미터 (MatchedCategories, MatchedDetails)를 사용하고 사용자 지정 실시간 알림을 설정할 수 있습니다.

다음은 CategoryEvent 일치에 대한 실시간 통화 분석 트랜스크립션 출력의 예시입니다.

```
"CategoryEvent": {
  "MatchedCategories": [ "shipping-complaint" ],
  "MatchedDetails": {
    "my package never arrived" : {
      "TimestampRanges": [
        {
          "BeginOffsetMillis": 19010,
          "EndOffsetMillis": 22690
        }
      ]
    }
  }
},
```

위의 예시는 '배송-불만' 범주의 규칙을 나타내는 "택배가 도착하지 않았습니다"라는 말과 정확히 일치하는 텍스트를 나타냅니다.

나열된 파라미터를 원하는 대로 조합하여 포함하도록 실시간 알림을 설정할 수 있습니다. 예를 들어 일치하는 구절(MatchedDetails)만 포함하거나 범주 이름(MatchedCategories)만 포함하도록 알림을 설정할 수 있습니다. 또는 모든 파라미터를 포함하도록 알림을 설정할 수도 있습니다.

실시간 알림을 설정하는 방법은 조직의 인터페이스와 원하는 알림 유형에 따라 달라집니다. 예를 들어 팝업 알림, 이메일, 문자 메시지 또는 시스템에서 수락할 수 있는 기타 알림을 보내도록 CategoryEvent 일치를 설정할 수 있습니다.

실시간 통화 분석 출력

실시간 통화 분석 트랜스크립트는 세그먼트별 턴바이턴 형식으로 표시됩니다. 여기에는 범주 이벤트, 문제 감지, 감정, PII 식별 및 수정이 포함됩니다. 범주 이벤트를 통해 실시간 알림을 설정할 수 있습니다. 자세한 내용은 [범주 일치에 대한 실시간 알림 생성](#) 섹션을 참조하세요.

정확도를 높이고 산업별 용어 포함 등 사용 사례에 맞게 트랜스크립트를 추가로 사용자 지정하려면 통화 분석 요청에 [사용자 지정 어휘](#) 또는 [사용자 지정 언어 모델](#)을 추가하세요. 트랜스크립션 결과에서 원하지 않는 욕설 등의 단어를 마스킹하거나 삭제하거나 태그를 지정하려면 [어휘 필터링](#)을 추가하세요.

다음 섹션에서는 실시간 통화 분석 트랜스크립션에 대한 JSON 출력의 예를 보여줍니다.

범주 이벤트

트랜스크립션 출력에서 범주 일치가 어떻게 보이는지는 다음과 같습니다. 이 예시는 19,010밀리초 타임스탬프에서 22,690밀리초 타임스탬프까지의 오디오가 '네트워크 불만 사항' 범주와 일치한다는 것을 보여줍니다. 이 경우 사용자 지정 '네트워크 불만 사항' 범주에서는 고객이 "네트워크 문제"(정확한 단어 일치)라고 말해야 합니다.

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

```
}
},
```

문제 감지

트랜스크립션 출력의 문제 감지 일치 결과는 다음과 같습니다. 이 예시는 문자 26에서 문자 62까지의 텍스트가 문제를 설명한다는 것을 보여줍니다.

```
"UtteranceEvent": {
  ...
  "Transcript": "Wang Xiulan I'm tired of the network issues my phone is having.",
  ...
  "IssuesDetected": [
    {
      "CharacterOffsets": {
        "BeginOffsetChar": 26,
        "EndOffsetChar": 62
      }
    }
  ]
},
```

감정

트랜스크립션 출력의 감정 분석은 다음과 같습니다.

```
"UtteranceEvent": {
  ...
  "Sentiment": "NEGATIVE",
  "Items": [{
    ...
  ]
},
```

PII 식별

트랜스크립션 출력의 PII 식별은 다음과 같습니다.

```
"Entities": [
  {
    "Content": "Wang Xiulan",
    "Category": "PII",
    "Type": "NAME",
    "BeginOffsetMillis": 7999375,
```

```

        "EndOffsetMillis": 199375,
        "Confidence": 0.9989
    }
],

```

PII 수정

트랜스크립션 출력의 PII 수정은 다음과 같습니다.

```

"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
        }
    ]
},

```

언어 식별

트랜스크립션 출력의 언어 식별은 다음과 같습니다.

```

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

```

}

컴파일된 실시간 통화 분석 출력

다음 트랜스크립션 출력에서는 일부 내용을 간결하게 줄임표로 대체했습니다.

```
{
  "CallAnalyticsTranscriptResultStream": {
    "BadRequestException": {},
    "ConflictException": {},
    "InternalFailureException": {},
    "LimitExceededException": {},
    "ServiceUnavailableException": {},
    "UtteranceEvent": {
      "UtteranceId": "58c27f92-7277-11ec-90d6-0242ac120003",
      "ParticipantRole": "CUSTOMER",
      "IsPartial": false,
      "Transcript": "Wang Xiulan I'm tired of the network issues my phone is
having.",
      "BeginOffsetMillis": 19010,
      "EndOffsetMillis": 22690,
      "Sentiment": "NEGATIVE",
      "Items": [{
        "Content": "Wang",
        "BeginOffsetMillis": 379937,
        "EndOffsetMillis": 299375,
        "Type": "pronunciation",
        "Confidence": 0.9961,
        "VocabularyFilterMatch": false
      },
      {
        "Content": "Xiulan",
        "EndOffsetMillis": 5899375,
        "BeginOffsetMillis": 3899375,
        "Type": "pronunciation",
        "Confidence": 0.9961,
        "VocabularyFilterMatch": false
      },
      ...
      {
        "Content": "network",
        "EndOffsetMillis": 199375,
        "BeginOffsetMillis": 9299375,
```



```

        "Type": "pronunciation",
        "Confidence": 0.9961,
        "VocabularyFilterMatch": false
    },
    {
        "Content": "issues",
        "EndOffsetMillis": 7899375,
        "BeginOffsetMillis": 5999375,
        "Type": "pronunciation",
        "Confidence": 0.9961,
        "VocabularyFilterMatch": false
    },
    {
        "Content": "my",
        "EndOffsetMillis": 9199375,
        "BeginOffsetMillis": 7999375,
        "Type": "pronunciation",
        "Confidence": 0.9961,
        "VocabularyFilterMatch": false
    },
    {
        "Content": "phone",
        "EndOffsetMillis": 199375,
        "BeginOffsetMillis": 9299375,
        "Type": "pronunciation",
        "Confidence": 0.9961,
        "VocabularyFilterMatch": false
    },
    ...
],
"Entities": [{
    "Content": "Wang Xiulan",
    "Category": "PII",
    "Type": "NAME",
    "BeginOffsetMillis": 7999375,
    "EndOffsetMillis": 199375,
    "Confidence": 0.9989
}],
"IssuesDetected": [{
    "CharacterOffsets": {
        "BeginOffsetChar": 26,
        "EndOffsetChar": 62
    }
}],
],

```

```
    "LanguageCode": "en-US",
    "LanguageIdentification": [
      {
        "Code": "en-US",
        "Score": 0.823
      },
      {
        "Code": "de-DE",
        "Score": 0.177
      }
    ],
  },
  "CategoryEvent": {
    "MatchedCategories": [
      "network-complaint"
    ],
    "MatchedDetails": {
      "network issues" : {
        "TimestampRanges": [
          {
            "BeginOffsetMillis": 9299375,
            "EndOffsetMillis": 7899375
          }
        ]
      }
    }
  }
}
```

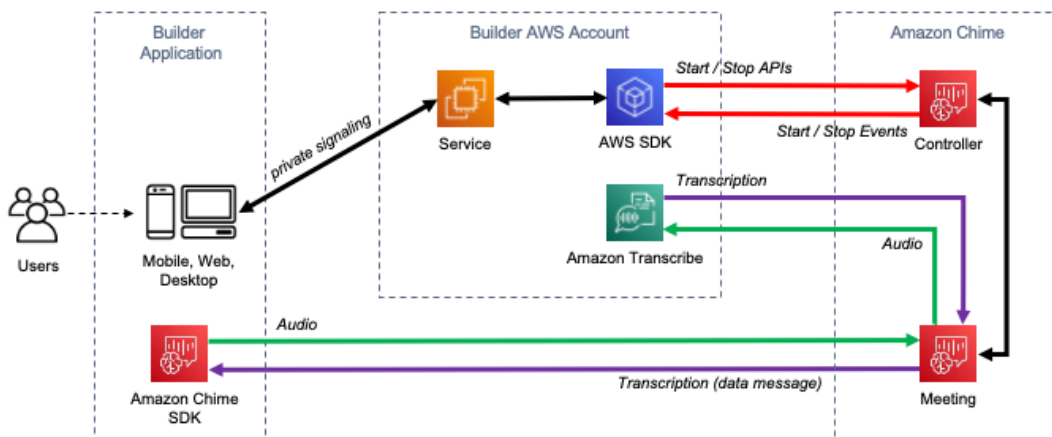
실시간으로 Amazon Chime 통화 트랜스크립션

Amazon Transcribe는 Amazon Chime SDK와 통합되어 Amazon Chime 통화의 실시간 트랜스크립션을 용이하게 합니다.

Amazon Chime SDK API를 사용하여 트랜스크립션을 요청하면 Amazon Chime은 Amazon Transcribe로 오디오 스트리밍을 시작하고 통화가 진행되는 동안 계속 스트리밍합니다.

Amazon Chime SDK는 '활성 화자' 알고리즘을 사용하여 상위 두 명의 활성 화자를 선택한 다음 단일 스트림을 통해 별도의 두 채널로 오디오를 Amazon Transcribe에 전송합니다. 회의 참가자는 Amazon Chime SDK 데이터 메시지를 통해 사용자 속성이 지정된 트랜스크립션을 받습니다. [Amazon Chime SDK 개발자 안내서](#)에서 전송 예시를 볼 수 있습니다.

Amazon Chime 트랜스크립션의 데이터 흐름은 다음 다이어그램에 나와 있습니다.



실시간 Amazon Chime 트랜스크립션을 설정하는 방법에 대한 추가 정보 및 자세한 지침은 Amazon Chime SDK 개발자 안내서의 [Amazon Chime SDK 라이브 트랜스크립션 사용](#)을 참조하세요. API 작업에 대한 내용은 [Amazon Chime SDK API 참조](#)를 참조하세요.

📘 AWS 기계 학습 블로그를 통해 더 자세히 알아보기

실시간 트랜스크립션으로 정확도를 높이는 방법에 대해 자세히 알아보려면 다음을 참조하세요.

- [Amazon Chime SDK 회의는 이제 Amazon Transcribe 및 Amazon Transcribe Medical을 통한 실시간 트랜스크립션을 지원합니다](#)

- [원격 진료 솔루션용 Amazon Chime SDK](#)

AWS SDKs를 사용한 Amazon Transcribe의 코드 예제

다음 코드 예제에서는 Amazon Transcribe를 AWS 소프트웨어 개발 키트(SDK)와 함께 사용하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접 호출하는 방법을 보여주며, 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

시나리오는 동일한 서비스 내에서 또는 다른 AWS 서비스와 결합된 상태에서 여러 함수를 직접적으로 호출하여 특정 태스크를 수행하는 방법을 보여주는 코드 예제입니다.

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 [섹션을 참조하세요](#) [AWS SDK에서 이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

코드 예시

- [AWS SDKs를 사용한 Amazon Transcribe의 기본 예제](#)
 - [AWS SDKs를 사용한 Amazon Transcribe 작업](#)
 - [AWS SDK 또는 CLI와 CreateVocabulary 함께 사용](#)
 - [AWS SDK 또는 CLI와 DeleteMedicalTranscriptionJob 함께 사용](#)
 - [AWS SDK 또는 CLI와 DeleteTranscriptionJob 함께 사용](#)
 - [AWS SDK 또는 CLI와 DeleteVocabulary 함께 사용](#)
 - [AWS SDK 또는 CLI와 GetTranscriptionJob 함께 사용](#)
 - [AWS SDK 또는 CLI와 GetVocabulary 함께 사용](#)
 - [AWS SDK 또는 CLI와 ListMedicalTranscriptionJobs 함께 사용](#)
 - [AWS SDK 또는 CLI와 ListTranscriptionJobs 함께 사용](#)
 - [AWS SDK 또는 CLI와 ListVocabularies 함께 사용](#)
 - [AWS SDK 또는 CLI와 StartMedicalTranscriptionJob 함께 사용](#)
 - [AWS SDK 또는 CLI와 StartTranscriptionJob 함께 사용](#)
 - [AWS SDK 또는 CLI와 UpdateVocabulary 함께 사용](#)
 - [AWS SDKs를 사용한 Amazon Transcribe 시나리오](#)
 - [Amazon Transcribe 스트리밍 앱 구축](#)
 - [AWS SDK를 사용하여 텍스트를 음성으로 변환하고 다시 텍스트로 변환](#)
 - [AWS SDK를 사용하여 Amazon Transcribe 사용자 지정 어휘 생성 및 구체화](#)
 - [AWS SDK를 사용하여 Amazon Transcribe로 오디오 트랜스크립션 및 작업 데이터 가져오기](#)

AWS SDKs를 사용한 Amazon Transcribe의 기본 예제

다음 코드 예제에서는 AWS SDK를 통해 Amazon Transcribe의 기본 기능을 사용하는 방법을 보여줍니다.

예제

- [AWS SDKs를 사용한 Amazon Transcribe 작업](#)
 - [AWS SDK 또는 CLI와 CreateVocabulary 함께 사용](#)
 - [AWS SDK 또는 CLI와 DeleteMedicalTranscriptionJob 함께 사용](#)
 - [AWS SDK 또는 CLI와 DeleteTranscriptionJob 함께 사용](#)
 - [AWS SDK 또는 CLI와 DeleteVocabulary 함께 사용](#)
 - [AWS SDK 또는 CLI와 GetTranscriptionJob 함께 사용](#)
 - [AWS SDK 또는 CLI와 GetVocabulary 함께 사용](#)
 - [AWS SDK 또는 CLI와 ListMedicalTranscriptionJobs 함께 사용](#)
 - [AWS SDK 또는 CLI와 ListTranscriptionJobs 함께 사용](#)
 - [AWS SDK 또는 CLI와 ListVocabularies 함께 사용](#)
 - [AWS SDK 또는 CLI와 StartMedicalTranscriptionJob 함께 사용](#)
 - [AWS SDK 또는 CLI와 StartTranscriptionJob 함께 사용](#)
 - [AWS SDK 또는 CLI와 UpdateVocabulary 함께 사용](#)

AWS SDKs를 사용한 Amazon Transcribe 작업

다음 코드 예제에서는 AWS SDKs를 사용하여 개별 Amazon Transcribe 작업을 수행하는 방법을 보여줍니다. 각 예제에는 GitHub에 대한 링크가 포함되어 있습니다. 여기에서 코드 설정 및 실행에 대한 지침을 찾을 수 있습니다.

이들 발췌문은 Amazon Transcribe API를 호출하며, 컨텍스트에서 실행되어야 하는 더 큰 프로그램에서 발췌한 코드입니다. [AWS SDKs를 사용한 Amazon Transcribe 시나리오](#)에서 컨텍스트에 맞는 작업을 볼 수 있습니다.

다음 예제에는 가장 일반적으로 사용되는 작업만 포함되어 있습니다. 전체 목록은 [Amazon Transcribe API 참조](#)를 참조하세요.

예제

- [AWS SDK 또는 CLI와 CreateVocabulary 함께 사용](#)

- [AWS SDK 또는 CLI와 DeleteMedicalTranscriptionJob 함께 사용](#)
- [AWS SDK 또는 CLI와 DeleteTranscriptionJob 함께 사용](#)
- [AWS SDK 또는 CLI와 DeleteVocabulary 함께 사용](#)
- [AWS SDK 또는 CLI와 GetTranscriptionJob 함께 사용](#)
- [AWS SDK 또는 CLI와 GetVocabulary 함께 사용](#)
- [AWS SDK 또는 CLI와 ListMedicalTranscriptionJobs 함께 사용](#)
- [AWS SDK 또는 CLI와 ListTranscriptionJobs 함께 사용](#)
- [AWS SDK 또는 CLI와 ListVocabularies 함께 사용](#)
- [AWS SDK 또는 CLI와 StartMedicalTranscriptionJob 함께 사용](#)
- [AWS SDK 또는 CLI와 StartTranscriptionJob 함께 사용](#)
- [AWS SDK 또는 CLI와 UpdateVocabulary 함께 사용](#)

AWS SDK 또는 CLI와 **CreateVocabulary** 함께 사용

다음 코드 예시는 CreateVocabulary의 사용 방법을 보여 줍니다.

작업 예제는 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 다음 코드 예제에서는 컨텍스트 내에서 이 작업을 확인할 수 있습니다.

- [사용자 지정 어휘 생성 및 세부 조정](#)

.NET

SDK for .NET

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
/// <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;
    }

```

- API 세부 정보는 AWS SDK for .NET API 참조의 [CreateVocabulary](#)를 참조하세요.

CLI

AWS CLI

사용자 지정 어휘를 생성하는 방법

다음 create-vocabulary 예시에서는 사용자 지정 어휘를 생성합니다. 사용자 지정 어휘를 생성하려면 더 정확하게 전사하려는 모든 용어가 포함된 텍스트 파일을 만들어야 합니다. vocabulary-file-uri의 경우 해당 텍스트 파일의 Amazon Simple Storage Service(Amazon S3) URI를 지정합니다. language-code에서 사용자 지정 어휘의 언어에 해당하는 언어 코드를 지정합니다. vocabulary-name에서 사용자 지정 어휘를 지칭할 이름을 지정합니다.

```

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

```

출력:


```
{
  "VocabularyName": "cli-vocab-example",
  "LanguageCode": "language-code",
  "VocabularyState": "PENDING"
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 [사용자 지정 어휘](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [CreateVocabulary](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [CreateVocabulary](#)를 참조하세요.

SAP ABAP

SDK for SAP ABAP API

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

TRY.

```

IF it_phrases IS NOT INITIAL.
    oo_result = lo_tnb->createvocabulary(
        iv_vocabularyname = iv_vocabulary_name
        iv_languagecode = iv_language_code
        it_phrases = it_phrases ).
ELSEIF iv_vocab_file_uri IS NOT INITIAL.
    oo_result = lo_tnb->createvocabulary(
        iv_vocabularyname = iv_vocabulary_name
        iv_languagecode = iv_language_code

```

```

        iv_vocabularyfileuri = iv_vocab_file_uri ).
    ENDIF.
    MESSAGE 'Custom vocabulary created.' TYPE 'I'.
    CATCH /aws1/cx_tnbbadrequestex INTO DATA(lo_bad_request_ex).
    MESSAGE lo_bad_request_ex TYPE 'I'.
    RAISE EXCEPTION lo_bad_request_ex.
    CATCH /aws1/cx_tnblimitexceededex INTO DATA(lo_limit_ex).
    MESSAGE lo_limit_ex TYPE 'I'.
    RAISE EXCEPTION lo_limit_ex.
    CATCH /aws1/cx_tnbinternalfailureex INTO DATA(lo_internal_ex).
    MESSAGE lo_internal_ex TYPE 'I'.
    RAISE EXCEPTION lo_internal_ex.
    CATCH /aws1/cx_tnbconflictexception INTO DATA(lo_conflict_ex).
    MESSAGE lo_conflict_ex TYPE 'I'.
    RAISE EXCEPTION lo_conflict_ex.
ENDTRY.

```

- API 세부 정보는 SDK for SAP ABAP API 참조의 [CreateVocabulary](#)를 참조하세요. AWS

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWS SDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK 또는 CLI와 **DeleteMedicalTranscriptionJob** 함께 사용

다음 코드 예시는 DeleteMedicalTranscriptionJob의 사용 방법을 보여 줍니다.

.NET

SDK for .NET

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```

/// <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;
    }

```

- API 세부 정보는 AWS SDK for .NET API 참조의 [DeleteMedicalTranscriptionJob](#)을 참조하세요.

CLI

AWS CLI

의료 전사 작업을 삭제하는 방법

다음 delete-medical-transcription-job 예시에서는 의료 전사 작업을 삭제합니다.

```

aws transcribe delete-medical-transcription-job \
    --medical-transcription-job-name medical-transcription-job-name

```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Transcribe 개발자 안내서의 [DeleteMedicalTranscriptionJob](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteMedicalTranscriptionJob](#)을 참조하세요.

JavaScript

SDK for JavaScript (v3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

클라이언트를 생성합니다.

```
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 };
```

의료 전사 작업을 삭제합니다.

```
// Import the required AWS SDK clients and commands for Node.js
import { DeleteMedicalTranscriptionJobCommand } from "@aws-sdk/client-transcribe";
import { transcribeClient } from "../libs/transcribeClient.js";

// Set the parameters
export const params = {
  MedicalTranscriptionJobName: "MEDICAL_JOB_NAME", // For example,
  'medical_transcription_demo'
};

export const run = async () => {
  try {
    const data = await transcribeClient.send(
      new DeleteMedicalTranscriptionJobCommand(params),
    );
    console.log("Success - deleted");
    return data; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
```

```

    }
};
run();

```

- 자세한 정보는 [AWS SDK for JavaScript 개발자 안내서](#)를 참조하세요.
- API 세부 정보는 AWS SDK for JavaScript API 참조의 [DeleteMedicalTranscriptionJob](#)을 참조하세요.

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWS SDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK 또는 CLI와 **DeleteTranscriptionJob** 함께 사용

다음 코드 예시는 DeleteTranscriptionJob의 사용 방법을 보여 줍니다.

작업 예제는 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 다음 코드 예제에서는 컨텍스트 내에서 이 작업을 확인할 수 있습니다.

- [사용자 지정 어휘 생성 및 세부 조정](#)

.NET

SDK for .NET

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```

    /// <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;
}
```

- API 세부 정보는 AWS SDK for .NET API 참조의 [DeleteTranscriptionJob](#)을 참조하세요.

CLI

AWS CLI

전사 작업 중 하나를 삭제하는 방법

다음 delete-transcription-job 예시에서는 전사 작업 중 하나를 삭제합니다.

```
aws transcribe delete-transcription-job \
    --transcription-job-name your-transcription-job
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Transcribe 개발자 안내서의 [DeleteTranscriptionJob](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteTranscriptionJob](#)을 참조하세요.

JavaScript

SDK for JavaScript (v3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

전사 작업을 삭제합니다.

```
// Import the required AWS SDK clients and commands for Node.js
import { DeleteTranscriptionJobCommand } from "@aws-sdk/client-transcribe";
import { transcribeClient } from "../libs/transcribeClient.js";

// Set the parameters
export const params = {
  TranscriptionJobName: "JOB_NAME", // Required. For example, 'transcription_demo'
};

export const run = async () => {
  try {
    const data = await transcribeClient.send(
      new DeleteTranscriptionJobCommand(params),
    );
    console.log("Success - deleted");
    return data; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};

run();
```

클라이언트를 생성합니다.

```
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 };
```

- 자세한 정보는 [AWS SDK for JavaScript 개발자 안내서](#)를 참조하세요.
- API 세부 정보는 AWS SDK for JavaScript API 참조의 [DeleteTranscriptionJob](#)을 참조하십시오.

Python

SDK for Python (Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [DeleteTranscriptionJob](#)을 참조하세요.

SAP ABAP

SDK for SAP ABAP API

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```

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

```

- API 세부 정보는 SDK for SAP ABAP API 참조의 [DeleteTranscriptionJob](#)을 참조하세요. AWS

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWS SDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK 또는 CLI와 **DeleteVocabulary** 함께 사용

다음 코드 예시는 DeleteVocabulary의 사용 방법을 보여 줍니다.

작업 예제는 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 다음 코드 예제에서는 컨텍스트 내에서 이 작업을 확인할 수 있습니다.

- [사용자 지정 어휘 생성 및 세부 조정](#)

.NET

SDK for .NET

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```

/// <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;
}

```

- API 세부 정보는 AWS SDK for .NET API 참조의 [DeleteVocabulary](#)를 참조하세요.

CLI

AWS CLI

사용자 지정 어휘를 삭제하는 방법

다음 delete-vocabulary 예시에서는 사용자 지정 어휘를 삭제합니다.

```

aws transcribe delete-vocabulary \
    --vocabulary-name vocabulary-name

```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Transcribe 개발자 안내서의 [사용자 지정 어휘](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [DeleteVocabulary](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [DeleteVocabulary](#)을 참조하세요.

SAP ABAP

SDK for SAP ABAP API

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

TRY.

```

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

```

- API 세부 정보는 SDK for SAP ABAP API 참조의 [DeleteVocabulary](#)를 참조하세요. AWS

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWS SDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK 또는 CLI와 **GetTranscriptionJob** 함께 사용

다음 코드 예시는 GetTranscriptionJob의 사용 방법을 보여 줍니다.

작업 예시는 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 다음 코드 예제에서는 컨텍스트 내에서 이 작업을 확인할 수 있습니다.

- [사용자 지정 어휘 생성 및 세부 조정](#)
- [오디오 전사 및 작업 데이터 가져오기](#)

.NET

SDK for .NET

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```

    /// <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;
    }

```

- API에 대한 세부 정보는 AWS SDK for .NET API 참조의 [GetTranscriptionJob](#)을 참조하세요.

CLI

AWS CLI

특정 전사 작업에 대한 정보를 가져오려면

다음 `get-transcription-job` 예시에서는 특정 전사 작업의 정보를 가져옵니다. 전사 결과에 액세스하려면 `TranscriptFileUri` 파라미터를 사용합니다. `MediaFileUri` 파라미터를 사용하면 이 작업에서 어떤 오디오 파일을 전사했는지 확인할 수 있습니다. 설정 객체를 사용하면 전사 작업에서 활성화한 선택적 기능을 확인할 수 있습니다.

```
aws transcribe get-transcription-job \
  --transcription-job-name your-transcription-job
```

출력:

```
{
  "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
}

```

자세한 내용은 Amazon Transcribe 개발자 안내서의 [시작하기\(AWS 명령줄 인터페이스\)](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetTranscriptionJob](#)을 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```

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

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [GetTranscriptionJob](#)을 참조하세요.

SAP ABAP

SDK for SAP ABAP API

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```

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

```



```

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

```

- API 세부 정보는 SDK for SAP ABAP API 참조의 [GetTranscriptionJob](#)을 참조하세요. AWS

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWS SDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK 또는 CLI와 **GetVocabulary** 함께 사용

다음 코드 예시는 GetVocabulary의 사용 방법을 보여 줍니다.

작업 예제는 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 다음 코드 예제에서는 컨텍스트 내에서 이 작업을 확인할 수 있습니다.

- [사용자 지정 어휘 생성 및 세부 조정](#)

.NET

SDK for .NET

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```

/// <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;
}
```

- API 세부 정보는 AWS SDK for .NET API 참조의 [GetVocabulary](#)를 참조하세요.

CLI

AWS CLI

사용자 지정 어휘에 대한 정보를 가져오려면

다음 `get-vocabulary` 예시에서는 이전에 생성한 사용자 지정 어휘의 정보를 가져옵니다.

```
aws transcribe get-vocabulary \
    --vocabulary-name cli-vocab-1
```

출력:

```
{
  "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"
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 [사용자 지정 어휘](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [GetVocabulary](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [GetVocabulary](#)를 참조하세요.

SAP ABAP

SDK for SAP ABAP API

 Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

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

- API 세부 정보는 SDK for SAP ABAP API 참조의 [GetVocabulary](#)를 참조하세요. AWS

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWS SDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK 또는 CLI와 **ListMedicalTranscriptionJobs** 함께 사용

다음 코드 예시는 ListMedicalTranscriptionJobs의 사용 방법을 보여 줍니다.

.NET

SDK for .NET

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
/// <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;
}
```

- API 세부 정보는 AWS SDK for .NET API 참조의 [ListMedicalTranscriptionJobs](#)를 참조하세요.

CLI

AWS CLI

의료 전사 작업을 나열하는 방법

다음 `list-medical-transcription-jobs` 예시에서는 AWS 계정 및 리전과 연결된 의료 트랜스크립션 작업을 나열합니다. 특정 전사 작업에 대한 자세한 정보를 얻으려면 전사 출력에서 `MedicalTranscriptionJobName` 파라미터의 값을 복사하고 `get-medical-transcription-job` 명령의 `MedicalTranscriptionJobName` 옵션에 해당 값을 지정합니다. 더 많은 전사 작업을 보려면 `NextToken` 파라미터의 값을 복사하고 `list-medical-transcription-jobs` 명령을 다시 실행한 다음 `--next-token` 옵션에 해당 값을 지정합니다.

```
aws transcribe list-medical-transcription-jobs
```

출력:

```
{
  "NextToken": "3/PblzkiGhzjER3KHuQt2fmbPLF7cDYafjFMEoGn440N/
gsuUSTIkGyanvRE6WMXfd/ZTEc2EZj+P9eii/
z102FDYli6RLI0WoRX4RwMisVrh9G0Kie0Y8ikBCdtqlZB10Wa9McC+eb0l
+LaDtZPC4u6ttoHLRlEfzqstHXSgapXg3tEBtm9piIaPB6MOM5BB6t86+qtmocTR/
qrteHZBBudhTfbCwhsxaqujHiiUvFdm3BQbKKWIW06yV9b+4f38oD2lVlan
+vfUs3gBYA15VTDmXXzQPBQ0HPjtwmFI+IWX15nSUjWuN3TUylHgPwzDaYT8qBtu0Z+3UG4V6b
+K2CC0XszXg5rBq9hYgNzy4XoFh/6s5DoSnzq49Q9xHgHdT2yBADFmvFK7myZBs75+2vQZ0SVpWUPy3WT/32zFAC
+mFYfUjtTZ8n/jq7aQEjQ42A
+X/7K6Jg0cdVPtEg8PlDr5kgYYG3q30mYXX37U3FZuJmnTI63VtIXsNn0U5eGoY0btpk00Nq9UkzgSJxqj84ZD5n
+S0EGy9ZUYBJRRcGeYUM3Q4DbSJfUwSAqcFdLIWZdp8qIREMQIBWy7BLwSdyqsQo2vRrd53hm5aWM7SVf6pPq6X/
IXR5+1eU00D8/coaTT4ES2DerbV6RkV4o0VT1d0SdVX/
MmtkNG8nYj8PqU07w7988quh1ZP6D80veJS1q73tUUR9MjnGernW2tAnvnLNhdefBcD
+sZVfYq3iBMFY7wTy1P1G6NqW9GrYDYox3tTPWlD7phpbVSyKrh/
PdYrps5UxnsGoA1b7L/FfAXDfUoGrGUB4N3JsPYXX9D++g+6gV1qBBs/
WfF934aKqfD6UTggm/zV3GA0WiBpfvAZRvEb924i6yGHYMC7y5401ZAwSBupmI
+FFd13CaP04kN1vJlth6aM5vUPXg4BpyUhtbRhWd/KxCvf9K0tLJGyL1A==",
  "MedicalTranscriptionJobSummaries": [
    {
      "MedicalTranscriptionJobName": "vocabulary-dictation-medical-
transcription-job",
      "CreationTime": "2020-09-21T21:17:27.016000+00:00",
      "StartTime": "2020-09-21T21:17:27.045000+00:00",
      "CompletionTime": "2020-09-21T21:17:59.561000+00:00",
      "LanguageCode": "en-US",
      "TranscriptionJobStatus": "COMPLETED",
      "OutputLocationType": "CUSTOMER_BUCKET",
      "Specialty": "PRIMARYCARE",
      "Type": "DICTATION"
    },
  ],
}
```

```

    {
      "MedicalTranscriptionJobName": "alternatives-dictation-medical-
transcription-job",
      "CreationTime": "2020-09-21T21:01:14.569000+00:00",
      "StartTime": "2020-09-21T21:01:14.592000+00:00",
      "CompletionTime": "2020-09-21T21:01:43.606000+00:00",
      "LanguageCode": "en-US",
      "TranscriptionJobStatus": "COMPLETED",
      "OutputLocationType": "CUSTOMER_BUCKET",
      "Specialty": "PRIMARYCARE",
      "Type": "DICTATION"
    },
    {
      "MedicalTranscriptionJobName": "alternatives-conversation-medical-
transcription-job",
      "CreationTime": "2020-09-21T19:09:18.171000+00:00",
      "StartTime": "2020-09-21T19:09:18.199000+00:00",
      "CompletionTime": "2020-09-21T19:10:22.516000+00:00",
      "LanguageCode": "en-US",
      "TranscriptionJobStatus": "COMPLETED",
      "OutputLocationType": "CUSTOMER_BUCKET",
      "Specialty": "PRIMARYCARE",
      "Type": "CONVERSATION"
    },
    {
      "MedicalTranscriptionJobName": "speaker-id-conversation-medical-
transcription-job",
      "CreationTime": "2020-09-21T18:43:37.157000+00:00",
      "StartTime": "2020-09-21T18:43:37.265000+00:00",
      "CompletionTime": "2020-09-21T18:44:21.192000+00:00",
      "LanguageCode": "en-US",
      "TranscriptionJobStatus": "COMPLETED",
      "OutputLocationType": "CUSTOMER_BUCKET",
      "Specialty": "PRIMARYCARE",
      "Type": "CONVERSATION"
    },
    {
      "MedicalTranscriptionJobName": "multichannel-conversation-medical-
transcription-job",
      "CreationTime": "2020-09-20T23:46:44.053000+00:00",
      "StartTime": "2020-09-20T23:46:44.081000+00:00",
      "CompletionTime": "2020-09-20T23:47:35.851000+00:00",
      "LanguageCode": "en-US",
      "TranscriptionJobStatus": "COMPLETED",

```

```

        "OutputLocationType": "CUSTOMER_BUCKET",
        "Specialty": "PRIMARYCARE",
        "Type": "CONVERSATION"
    }
]
}

```

자세한 내용은 Amazon Transcribe 개발자 안내서의 <https://docs.aws.amazon.com/transcribe/latest/dg/batch-med-transcription.html>을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListMedicalTranscriptionJobs](#)를 참조하세요.

JavaScript

SDK for JavaScript (v3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

클라이언트를 생성합니다.

```

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 };

```

의료 전사 작업을 나열합니다.

```

// 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();

```

- 자세한 정보는 [AWS SDK for JavaScript 개발자 안내서](#)를 참조하세요.
- API 세부 정보는 AWS SDK for JavaScript API 참조의 [ListMedicalTranscriptionJobs](#)를 참조하세요.

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWS SDK에서 이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK 또는 CLI와 **ListTranscriptionJobs** 함께 사용

다음 코드 예시는 ListTranscriptionJobs의 사용 방법을 보여 줍니다.

.NET

SDK for .NET

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
/// <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;
}
```

- API 세부 정보는 AWS SDK for .NET API 참조의 [ListTranscriptionJobs](#)를 참조하세요.

CLI

AWS CLI

전사 작업을 나열하는 방법

다음 `list-transcription-jobs` 예시에서는 AWS 계정 및 리전과 연결된 트랜스크립션 작업을 나열합니다.

aws transcribe list-transcription-jobs

출력:

```
{
  "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"
    }
  ]
}

```

자세한 내용은 Amazon Transcribe 개발자 안내서의 [시작하기\(AWS 명령줄 인터페이스\)](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListTranscriptionJobs](#)를 참조하세요.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```

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("-----");
    });
}
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [ListTranscriptionJobs](#)를 참조하십시오.

JavaScript

SDK for JavaScript (v3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

전사 작업을 나열합니다.

```
// 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();
```

클라이언트를 생성합니다.

```
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 };
```

- 자세한 정보는 [AWS SDK for JavaScript 개발자 안내서](#)를 참조하세요.
- API 세부 정보는 AWS SDK for JavaScript API 참조의 [ListTranscriptionJobs](#)를 참조하십시오.

Python

SDK for Python (Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [ListTranscriptionJobs](#)를 참조하세요.

SAP ABAP

SDK for SAP ABAP API

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```

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

```

- API 세부 정보는 SDK for SAP ABAP API 참조의 [ListTranscriptionJobs](#)를 참조하세요. AWS

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWS SDK에서 이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK 또는 CLI와 **ListVocabularies** 함께 사용

다음 코드 예시는 ListVocabularies의 사용 방법을 보여 줍니다.

작업 예제는 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 다음 코드 예제에서는 컨텍스트 내에서 이 작업을 확인할 수 있습니다.

- [사용자 지정 어휘 생성 및 세부 조정](#)

.NET

SDK for .NET

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```

    /// <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;  
}
```

- API 세부 정보는 AWS SDK for .NET API 참조의 [ListVocabularies](#)를 참조하세요.

CLI

AWS CLI

사용자 지정 어휘를 나열하는 방법

다음 `list-vocabularies` 예시에서는 AWS 계정 및 리전과 연결된 사용자 지정 어휘를 나열합니다.

```
aws transcribe list-vocabularies
```

출력:

```
{  
  "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"
    }
  ]
}

```

자세한 내용은 Amazon Transcribe 개발자 안내서의 [사용자 지정 어휘](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [ListVocabularies](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```

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

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [ListVocabularies](#)를 참조하세요.

SAP ABAP

SDK for SAP ABAP API

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```

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

```

```

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

```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 [ListVocabularies](#)를 참조하세요.

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWS SDK에서 이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK 또는 CLI와 **StartMedicalTranscriptionJob** 함께 사용

다음 코드 예시는 StartMedicalTranscriptionJob의 사용 방법을 보여 줍니다.

.NET

SDK for .NET

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```

/// <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;
    }

```

- API 세부 정보는 AWS SDK for .NET API 참조의 [StartMedicalTranscriptionJob](#)을 참조하세요.

CLI

AWS CLI

예 1: 오디오 파일로 저장된 의료 구술을 전사하는 방법

다음 start-medical-transcription-job 예시에서는 오디오 파일을 전사합니다. OutputBucketName 파라미터에는 전사 출력의 위치를 지정합니다.

```
aws transcribe start-medical-transcription-job \
  --cli-input-json file:///myfile.json
```

myfile.json의 콘텐츠:

```
{
  "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"
  }
}
```

출력:

```
{
  "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"
  }
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 [배치 전사 개요](#)를 참조하세요.

예시 2: 오디오 파일로 저장된 의사와 환자 간 대화 전사

다음 `start-medical-transcription-job` 예시에서는 의사와 환자 간 대화가 포함된 오디오 파일을 전사합니다. `OutputBucketName` 파라미터에는 전사 출력의 위치를 지정합니다.

```
aws transcribe start-medical-transcription-job \
  --cli-input-json file://mysecondfile.json
```

`mysecondfile.json`의 콘텐츠:

```
{
  "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"
  }
}
```

출력:

```
{
  "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"
  }
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 [배치 전사 개요](#)를 참조하세요.

예시 3: 의사와 환자 간 대화가 담긴 다중 채널 오디오 파일 전사

다음 `start-medical-transcription-job` 예시에서는 오디오 파일에 있는 각 채널의 오디오를 전사하고 각 채널의 개별 전사를 단일 전사 출력으로 병합합니다. `OutputBucketName` 파라미터에는 전사 출력의 위치를 지정합니다.

```
aws transcribe start-medical-transcription-job \
  --cli-input-json file://mythirdfile.json
```

`mythirdfile.json`의 콘텐츠:

```
{
  "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
  }
}
```

출력:

```
{
  "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"
  }
}
```

```
}
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 [채널 식별](#)을 참조하세요.

예시 4: 의사와 환자 간 대화의 오디오 파일을 전사하고 전사 출력에서 화자 식별

다음 start-medical-transcription-job 예시에서는 오디오 파일을 전사하고 전사 출력에서 각 화자의 음성에 레이블을 지정합니다. OutputBucketName 파라미터에는 전사 출력의 위치를 지정합니다.

```
aws transcribe start-medical-transcription-job \
  --cli-input-json file://myfourthfile.json
```

myfourthfile.json의 콘텐츠:

```
{
  "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
  }
}
```

출력:

```
{
  "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"
}
}

```

자세한 내용은 Amazon Transcribe 개발자 안내서의 [화자 식별](#)을 참조하세요.

예시 5: 최대 2개의 대체 전사를 사용하여 오디오 파일로 저장된 의료 대화 전사

다음 start-medical-transcription-job 예시에서는 단일 오디오 파일에서 최대 2개의 대체 전사를 생성합니다. 모든 전사에는 신뢰도가 있습니다. 기본적으로 Amazon Transcribe은 신뢰도가 가장 높은 전사를 반환합니다. Amazon Transcribe가 신뢰도가 낮은 추가 전사를 반환하도록 지정할 수도 있습니다. OutputBucketName 파라미터에는 전사 출력의 위치를 지정합니다.

```

aws transcribe start-medical-transcription-job \
  --cli-input-json file://myfifthfile.json

```

myfifthfile.json의 콘텐츠:

```

{
    "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
    }
}

```

```
}
```

출력:

```
{
  "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"
  }
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 [대체 전사](#)를 참조하세요.

예시 6: 최대 2개의 대체 전사를 사용하여 의료 구술 오디오 파일 전사

다음 start-medical-transcription-job 예시에서는 오디오 파일을 전사하고 어휘 필터를 사용하여 원하지 않는 단어를 마스킹합니다. OutputBucketName 파라미터에는 전사 출력의 위치를 지정합니다.

```
aws transcribe start-medical-transcription-job \
  --cli-input-json file://mysixthfile.json
```

mysixthfile.json의 콘텐츠:

```
{
  "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
    }
}

```

출력:

```

{
    "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"
    }
}

```

자세한 내용은 Amazon Transcribe 개발자 안내서의 [대체 전사](#)를 참조하세요.

예시 7: 사용자 지정 어휘로 정확도를 높여 의료 구술 오디오 파일 전사

다음 start-medical-transcription-job 예시에서는 오디오 파일을 전사하고 이전에 생성한 의료 사용자 지정 어휘를 사용하여 전사 정확도를 높입니다. OutputBucketName 파라미터에는 전사 출력의 위치를 지정합니다.

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

```
--cli-input-json file://myseventhfile.json
```

mysixthfile.json의 콘텐츠:

```
{
  "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"
  }
}
```

출력:

```
{
  "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"
  }
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 [의료 사용자 지정 어휘](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [StartMedicalTranscriptionJob](#)을 참조하세요.

JavaScript

SDK for JavaScript (v3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

클라이언트를 생성합니다.

```
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 };
```

의료 전사 작업을 시작합니다.

```
// 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();
```

- 자세한 정보는 [AWS SDK for JavaScript 개발자 안내서](#)를 참조하세요.
- API 세부 정보는 AWS SDK for JavaScript API 참조의 [StartMedicalTranscriptionJob](#)을 참조하세요.

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 [섹션을 참조하세요](#)[AWS SDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK 또는 CLI와 **StartTranscriptionJob** 함께 사용

다음 코드 예시는 StartTranscriptionJob의 사용 방법을 보여 줍니다.

작업 예시는 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 다음 코드 예제에서는 컨텍스트 내에서 이 작업을 확인할 수 있습니다.

- [사용자 지정 어휘 생성 및 세부 조정](#)
- [오디오 전사 및 작업 데이터 가져오기](#)

.NET

SDK for .NET

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```

    /// <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;
}
```

- API 세부 정보는 AWS SDK for .NET API 참조의 [StartTranscriptionJob](#)을 참조하세요.

CLI

AWS CLI

예 1: 오디오 파일을 전사하는 방법

다음 start-transcription-job 예시에서는 오디오 파일을 전사합니다.

```
aws transcribe start-transcription-job \  
  --cli-input-json file://myfile.json
```

myfile.json의 콘텐츠:

```
{  
  "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"  
  }  
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 [시작하기\(AWS 명령줄 인터페이스\)](#)를 참조하세요.

예시 2: 다중 채널 오디오 파일 전사

다음 start-transcription-job 예시에서는 다중 채널 오디오 파일을 전사합니다.

```
aws transcribe start-transcription-job \  
  --cli-input-json file://mysecondfile.json
```

mysecondfile.json의 콘텐츠:

```
{
  "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
  }
}
```

출력:

```
{
  "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
    }
  }
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 [다중 채널 오디오 전사](#)를 참조하세요.

예시 3: 오디오 파일을 전사하고 다양한 화자 식별

다음 start-transcription-job 예시에서는 오디오 파일을 전사하고 전사 출력에서 화자를 식별합니다.

```
aws transcribe start-transcription-job \
  --cli-input-json file://mythirdfile.json
```

mythirdfile.json의 콘텐츠:

```
{
  "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
  }
}
```

출력:

```
{
  "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
    }
  }
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 [화자 식별](#)을 참조하세요.

예시 4: 오디오 파일을 전사하고 전사 출력에서 원하지 않는 단어 마스킹

다음 start-transcription-job 예시에서는 오디오 파일을 전사하고 이전에 생성한 어휘 필터를 사용하여 원하지 않는 단어를 마스킹합니다.

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

```
--cli-input-json file://myfourthfile.json
```

myfourthfile.json의 콘텐츠:

```
{
  "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"
  }
}
```

출력:

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

자세한 내용은 Amazon Transcribe 개발자 안내서의 [전사 필터링](#)을 참조하세요.

예시 5: 오디오 파일을 전사하고 전사 출력에서 원하지 않는 단어 제거

다음 start-transcription-job 예시에서는 오디오 파일을 전사하고 이전에 생성한 어휘 필터를 사용하여 원하지 않는 단어를 마스킹합니다.

```
aws transcribe start-transcription-job \
  --cli-input-json file://myfifthfile.json
```

myfifthfile.json의 콘텐츠:

```
{
  "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"
  }
}
```

출력:

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

자세한 내용은 Amazon Transcribe 개발자 안내서의 [전사 필터링](#)을 참조하세요.

예시 6: 사용자 지정 어휘로 정확도를 높여 오디오 파일 전사

다음 start-transcription-job 예시에서는 오디오 파일을 전사하고 이전에 생성한 어휘 필터를 사용하여 원하지 않는 단어를 마스킹합니다.

```
aws transcribe start-transcription-job \
  --cli-input-json file:///mysixthfile.json
```

mysixthfile.json의 콘텐츠:

```
{
  "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"
  }
}
```

출력:

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

자세한 내용은 Amazon Transcribe 개발자 안내서의 [전사 필터링](#)을 참조하세요.

예시 7: 오디오 파일의 언어를 식별하고 전사

다음 start-transcription-job 예시에서는 오디오 파일을 전사하고 이전에 생성한 어휘 필터를 사용하여 원하지 않는 단어를 마스킹합니다.

```
aws transcribe start-transcription-job \
  --cli-input-json file://myseventhfile.json
```

myseventhfile.json의 콘텐츠:

```
{
  "TranscriptionJobName": "cli-identify-language-transcription-job",
  "IdentifyLanguage": true,
  "Media": {
    "MediaFileUri": "s3://amzn-s3-demo-bucket/Amazon-S3-prefix/your-media-
file-name.file-extension"
  }
}
```

출력:

```
{
  "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
  }
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 [언어 식별](#)을 참조하세요.

예시 8: 개인 식별 정보를 편집하여 오디오 파일 전사

다음 start-transcription-job 예시에서는 오디오 파일을 전사하고 전사 출력에서 개인 식별 정보를 편집합니다.

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



```
--cli-input-json file://myeighthfile.json
```

myeighthfile.json의 콘텐츠:

```
{
  "TranscriptionJobName": "cli-redaction-job",
  "LanguageCode": "language-code",
  "Media": {
    "MediaFileUri": "s3://Amazon-S3-Prefix/your-media-file.file-extension"
  },
  "ContentRedaction": {
    "RedactionOutput": "redacted",
    "RedactionType": "PII"
  }
}
```

출력:

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

자세한 내용은 Amazon Transcribe 개발자 안내서의 [자동 콘텐츠 편집](#)을 참조하세요.

예시 9: 개인 식별 정보(PII)가 편집된 기록과 편집되지 않은 기록 생성

다음 start-transcription-job 예시에서는 오디오 파일의 전사 2개를 생성합니다. 하나는 개인 식별 정보를 편집한 것이고 다른 하나는 편집하지 않은 것입니다.

```
aws transcribe start-transcription-job \
  --cli-input-json file://myninthfile.json
```

myninthfile.json의 콘텐츠:

```
{
  "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"
  }
}
```

출력:

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

자세한 내용은 Amazon Transcribe 개발자 안내서의 [자동 콘텐츠 편집](#)을 참조하세요.

예시 10: 이전에 생성한 사용자 지정 언어 모델을 사용하여 오디오 파일 전사

다음 `start-transcription-job` 예시에서는 이전에 생성한 사용자 지정 언어 모델을 사용하여 오디오 파일을 전사합니다.

```
aws transcribe start-transcription-job \  
  --cli-input-json file://mytenthfile.json
```

`mytenthfile.json`의 콘텐츠:

```
{  
  "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"  
  }  
}
```

출력:

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

자세한 내용은 Amazon Transcribe 개발자 안내서의 [사용자 지정 언어 모델을 사용한 도메인별 전사 정확도 개선](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조의 [StartTranscriptionJob](#)을 참조하세요.

JavaScript

SDK for JavaScript (v3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

전사 작업을 시작합니다.

```
// 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();
```

클라이언트를 생성합니다.

```
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 };
```

- 자세한 정보는 [AWS SDK for JavaScript 개발자 안내서](#)를 참조하세요.
- API 세부 정보는 AWS SDK for JavaScript API 참조의 [StartTranscriptionJob](#)을 참조하십시오.

Python

SDK for Python (Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartTranscriptionJob](#)을 참조하세요.

SAP ABAP

SDK for SAP ABAP API

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```

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

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

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

```

- API 세부 정보는 SDK for SAP ABAP API 참조의 [StartTranscriptionJob](#)을 참조하세요. AWS

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWS SDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK 또는 CLI와 **UpdateVocabulary** 함께 사용

다음 코드 예시는 UpdateVocabulary의 사용 방법을 보여 줍니다.

작업 예제는 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 다음 코드 예제에서는 컨텍스트 내에서 이 작업을 확인할 수 있습니다.

- [사용자 지정 어휘 생성 및 세부 조정](#)

.NET

SDK for .NET

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
/// <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;
}
```

- API 세부 정보는 AWS SDK for .NET API 참조의 [UpdateVocabulary](#)를 참조하세요.

CLI

AWS CLI

사용자 지정 어휘를 새 용어로 업데이트하는 방법

다음 update-vocabulary 예시에서는 사용자 지정 어휘를 생성하는 데 사용된 용어를 사용자가 제공한 새 용어로 덮어씁니다. 사전 조건: 사용자 지정 어휘의 용어를 바꾸려면 새 용어가 포함된 파일이 필요합니다.

```
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
```

출력:

```
{
  "VocabularyName": "custom-vocabulary",
  "LanguageCode": "language",
  "VocabularyState": "PENDING"
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 [사용자 지정 어휘](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 참조에서 [UpdateVocabulary](#)를 참조하세요.

Python

SDK for Python(Boto3)

 Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [UpdateVocabulary](#)를 참조하세요.

SAP ABAP

SDK for SAP ABAP API

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```

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

```

- API 세부 정보는 AWS SDK for SAP ABAP API 참조의 [UpdateVocabulary](#)를 참조하세요.

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWS SDK에서 이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDKs를 사용한 Amazon Transcribe 시나리오

다음 코드 예제에서는 Amazon Transcribe AWS SDKs에서 일반적인 시나리오를 구현하는 방법을 보여줍니다. 이러한 시나리오에서는 Amazon Transcribe 내에서 또는 다른 AWS 서비스와 결합된 상태에서 여러 함수를 직접적으로 호출하여 특정 작업을 수행하는 방법을 보여줍니다. 각 시나리오에는 전체 소스 코드에 대한 링크가 포함되어 있습니다. 여기에서 코드를 설정 및 실행하는 방법에 대한 지침을 찾을 수 있습니다.

시나리오는 컨텍스트에 맞는 서비스 작업을 이해하는 데 도움이 되도록 중급 수준의 경험을 대상으로 합니다.

예시

- [Amazon Transcribe 스트리밍 앱 구축](#)
- [AWS SDK를 사용하여 텍스트를 음성으로 변환하고 다시 텍스트로 변환](#)
- [AWS SDK를 사용하여 Amazon Transcribe 사용자 지정 어휘 생성 및 구체화](#)
- [AWS SDK를 사용하여 Amazon Transcribe로 오디오 트랜스크립션 및 작업 데이터 가져오기](#)

Amazon Transcribe 스트리밍 앱 구축

다음 코드 예시는 라이브 오디오를 실시간으로 기록, 변환 및 번역하고 결과를 이메일로 보내는 앱을 구축하는 방법을 보여줍니다.

JavaScript

SDK for JavaScript (v3)

Amazon Transcribe를 사용하여 라이브 오디오를 실시간으로 기록, 변환 및 번역하고 Amazon Simple Email Service(Amazon SES)를 사용하여 결과를 이메일로 전송하는 앱을 구축하는 방법을 보여줍니다.

전체 소스 코드와 설정 및 실행 방법에 대한 지침은 [GitHub](#)에서 전체 예시를 참조하세요.

이 예제에서 사용되는 서비스

- Amazon Comprehend
- Amazon SES
- Amazon Transcribe
- Amazon Translate

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 [섹션을 참조하세요](#)[AWS SDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK를 사용하여 텍스트를 음성으로 변환하고 다시 텍스트로 변환

다음 코드 예시는 다음과 같은 작업을 수행하는 방법을 보여줍니다.

- Amazon Polly를 사용하여 일반 텍스트(UTF-8) 입력 파일을 오디오 파일에 합성합니다.
- Amazon S3 버킷에 오디오 파일을 업로드합니다.
- Amazon Transcribe를 사용하여 오디오 파일을 텍스트로 변환합니다.
- 텍스트를 표시합니다.

Rust

SDK for Rust

Amazon Polly를 사용하여 일반 텍스트(UTF-8) 입력 파일을 오디오 파일에 합성하고, 오디오 파일을 Amazon S3 버킷에 업로드하고, Amazon Transcribe를 사용하여 해당 오디오 파일을 텍스트로 변환하고, 텍스트를 표시합니다.

전체 소스 코드와 설정 및 실행 방법에 대한 지침은 [GitHub](#)에서 전체 예시를 참조하세요.

이 예제에서 사용되는 서비스

- Amazon Polly
- Amazon S3
- Amazon Transcribe

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 [섹션을 참조하세요](#)[AWS SDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK를 사용하여 Amazon Transcribe 사용자 지정 어휘 생성 및 구체화

다음 코드 예제에서는 다음과 같은 작업을 수행하는 방법을 보여줍니다.

- Amazon S3에 오디오 파일을 업로드합니다.
- Amazon Transcribe 작업을 실행하여 파일을 전사하고 결과를 얻습니다.
- 사용자 지정 어휘를 생성하고 세부 조정하여 전사 정확도를 향상시킵니다.
- 사용자 지정 어휘와 함께 작업을 실행하고 결과를 얻습니다.

Python

SDK for Python(Boto3)

 Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

Lewis Carroll의 Jabberwocky 낭독이 포함된 오디오 파일을 트랜스크립션합니다. 먼저 Amazon Transcribe 작업을 래핑하는 함수를 생성하여 시작합니다.

```
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

```

래퍼 함수를 호출하여 사용자 지정 어휘 없이 오디오를 트랜스크립션한 다음 다른 버전의 사용자 지정 어휘를 사용하여 트랜스크립션하면 결과가 개선됩니다.

```

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!")

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 다음 주제를 참조하세요.
 - [CreateVocabulary](#)
 - [DeleteTranscriptionJob](#)
 - [DeleteVocabulary](#)
 - [GetTranscriptionJob](#)
 - [GetVocabulary](#)
 - [ListVocabularies](#)
 - [StartTranscriptionJob](#)
 - [UpdateVocabulary](#)

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWS SDK에서 이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK를 사용하여 Amazon Transcribe로 오디오 트랜스크립션 및 작업 데이터 가져오기

다음 코드 예제는 다음과 같은 작업을 수행하는 방법을 보여줍니다.

- Amazon Transcribe를 통해 전사 작업을 시작합니다.
- 작업이 완료될 때까지 기다립니다.
- 기록이 저장되는 URI를 가져옵니다.

자세한 내용은 [Amazon Transcribe 시작하기](#)를 참조하세요.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

PCM 파일을 전사합니다.

```
/**
 * 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();
    }
}

```

컴퓨터 마이크의 스트리밍 오디오를 전사합니다.

```

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();
    }
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 다음 주제를 참조하세요.
 - [GetTranscriptionJob](#)
 - [StartTranscriptionJob](#)

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS 코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

```
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()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 다음 주제를 참조하세요.
 - [GetTranscriptionJob](#)
 - [StartTranscriptionJob](#)

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWS SDK에서 이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

Amazon Transcribe의 보안

AWS에서는 클라우드 보안을 가장 중요하게 생각합니다. AWS 고객은 보안에 가장 보안에 민감한 조직의 요구 사항에 부합하도록 구축된 데이터 센터 및 네트워크 아키텍처의 혜택을 누릴 수 있습니다.

보안은 AWS와 사용자의 공동 책임입니다. [공동 책임 모델](#)은 이 사항을 클라우드의 보안 및 클라우드 내 보안으로 설명합니다.

- 클라우드의 보안: AWS는 AWS 클라우드 클라우드에서 AWS 서비스를 실행하는 인프라를 보호합니다. AWS는 또한 안전하게 사용할 수 있는 서비스를 제공합니다. 서드 파티 감사자는 [AWS 규정 준수 프로그램](#)의 일환으로 보안 효과를 정기적으로 테스트하고 검증합니다. Amazon Transcribe에 적용되는 규정 준수 프로그램에 대한 자세한 내용은 [규정 준수 프로그램의 범위에 속하는 AWS 서비스](#)를 참조하세요.
- 클라우드 내 보안: 귀하의 책임은 귀하가 사용하는 AWS 서비스에 의해 결정됩니다. 또한 귀하는 귀사의 데이터 민감도, 귀사의 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

이 설명서는 Amazon Transcribe 사용 시 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다. 다음 주제에서는 보안 및 규정 준수 목표를 충족하도록 Amazon Transcribe을(를) 구성하는 방법을 보여줍니다. 또한 Amazon Transcribe 리소스를 모니터링하고 보호하기 위해 다른 AWS 서비스를 사용하는 방법을 배웁니다.

주제

- [용 자격 증명 및 액세스 관리 Amazon Transcribe](#)
- [의 데이터 보호 Amazon Transcribe](#)
- [모니터링 Amazon Transcribe](#)
- [의 규정 준수 검증 Amazon Transcribe](#)
- [Amazon Transcribe의 복원성](#)
- [Amazon Transcribe에서 인프라 보안](#)
- [의 취약성 분석 및 관리 Amazon Transcribe](#)
- [의 보안 모범 사례 Amazon Transcribe](#)

용 자격 증명 및 액세스 관리 Amazon Transcribe

AWS Identity and Access Management (IAM)는 관리자가 AWS 리소스에 대한 액세스를 안전하게 제어하는 데 도움이 AWS 서비스 되는입니다. IAM 관리자는 누가 Amazon Transcribe 리소스를 사용할 수 있는 인증(로그인) 및 권한(권한 있음)을 받을 수 있는지 제어합니다. IAM은 추가 비용 없이 사용할 수 AWS 서비스 있는입니다.

주제

- [대상](#)
- [ID를 통한 인증](#)
- [정책을 사용하여 액세스 관리](#)
- [Amazon Transcribe 에서 IAM을 사용하는 방법](#)
- [교차 서비스 혼동된 대리인 방지](#)
- [Amazon Transcribe 자격 증명 기반 정책 예제](#)
- [Amazon Transcribe 자격 증명 및 액세스 문제 해결](#)

대상

AWS Identity and Access Management (IAM)를 사용하는 방법은 역할에 따라 다릅니다.

- 서비스 사용자 - 기능에 액세스할 수 없는 경우 관리자에게 권한 요청([참조 Amazon Transcribe 자격 증명 및 액세스 문제 해결](#))
- 서비스 관리자 - 사용자 액세스 결정 및 권한 요청 제출([Amazon Transcribe 에서 IAM을 사용하는 방법 참조](#))
- IAM 관리자 - 액세스를 관리하기 위한 정책 작성([Amazon Transcribe 자격 증명 기반 정책 예제 참조](#))

ID를 통한 인증

인증은 자격 증명 자격 증명을 AWS 사용하여 로그인하는 방법입니다. AWS 계정 루트 사용자, IAM 사용자 또는 IAM 역할을 수임하여 인증해야 합니다.

AWS IAM Identity Center (IAM Identity Center), Single Sign-On 인증 또는 Google/Facebook 자격 증명과 같은 자격 증명 소스의 자격 증명을 사용하여 페더레이션 자격 증명으로 로그인할 수 있습니다.

로그인하는 방법에 대한 자세한 내용은 AWS Sign-In 사용 설명서의 [AWS 계정에 로그인하는 방법](#) 섹션을 참조하세요.

프로그래밍 방식 액세스를 위해서는 요청에 암호화 방식으로 서명할 수 있는 SDK 및 CLI를 AWS 제공합니다. 자세한 내용은 IAM 사용 설명서의 [API 요청용 AWS Signature Version 4](#) 섹션을 참조하세요.

AWS 계정 루트 사용자

를 생성할 때 모든 AWS 서비스 및 리소스에 대한 완전한 액세스 권한이 있는 AWS 계정 theroot 사용자라는 하나의 로그인 자격 증명으로 AWS 계정시작합니다. 일상적인 태스크에 루트 사용자를 사용하지 않을 것을 강력히 권장합니다. 루트 사용자 자격 증명이 필요한 작업은 IAM 사용 설명서의 [루트 사용자 자격 증명이 필요한 작업](#) 섹션을 참조하세요.

페더레이션 ID

가장 좋은 방법은 인간 사용자에게 자격 증명 공급자와의 페더레이션을 사용하여 임시 자격 증명을 AWS 서비스 사용하여 액세스하도록 요구하는 것입니다.

페더레이션 자격 증명은 엔터프라이즈 디렉터리, 웹 자격 증명 공급자 또는 자격 증명 소스의 자격 증명을 AWS 서비스 사용하여 Directory Service 에 액세스하는 사용자입니다. 페더레이션 ID는 임시 자격 증명을 제공하는 역할을 수임합니다.

중앙 집중식 액세스 관리를 위해 AWS IAM Identity Center를 추천합니다. 자세한 정보는 AWS IAM Identity Center 사용 설명서의 [What is IAM Identity Center?](#)를 참조하세요.

IAM 사용자 및 그룹

[IAM 사용자](#)는 단일 개인 또는 애플리케이션에 대한 특정 권한을 가진 ID입니다. 장기 자격 증명에 있는 IAM 사용자 대신 임시 자격 증명을 사용하는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 [자격 증명 공급자와의 페더레이션을 사용하여 임시 자격 증명을 AWS 사용하여 액세스하도록 인간 사용자에게 요구](#)하기를 참조하세요.

[IAM 그룹](#)은 IAM 사용자 모음을 지정하고 대규모 사용자 집합에 대한 관리 권한을 더 쉽게 만듭니다. 자세한 내용은 IAM 사용 설명서의 [IAM 사용자 사용 사례](#) 섹션을 참조하세요.

IAM 역할

[IAM 역할](#)은 임시 자격 증명을 제공하는 특정 권한이 있는 자격 증명입니다. [사용자에서 IAM 역할\(콘솔\)로 전환하거나 또는 API 작업을 호출하여 역할을](#) 수임할 수 있습니다. AWS CLI AWS 자세한 내용은 IAM 사용 설명서의 [역할 수임 방법](#)을 참조하세요.

IAM 역할은 페더레이션 사용자 액세스, 임시 IAM 사용자 권한, 교차 계정 액세스, 교차 서비스 액세스 및 Amazon EC2에서 실행되는 애플리케이션에 유용합니다. 자세한 내용은 IAM 사용 설명서의 [교차 계정 리소스 액세스](#)를 참조하세요.

정책을 사용하여 액세스 관리

정책을 AWS 생성하고 자격 증명 또는 리소스에 연결하여 AWS 에서 액세스를 제어합니다. 정책은 자격 증명 또는 리소스와 연결될 때 권한을 정의합니다. 보안 주체가 요청할 때 이러한 정책을 AWS 평가합니다. 대부분의 정책은 JSON 문서 AWS 로 저장됩니다. JSON 정책 문서에 대한 자세한 내용은 IAM 사용 설명서의 [JSON 정책 개요](#) 섹션을 참조하세요.

정책을 사용하여 관리자는 어떤 보안 주체가 어떤 리소스에 대해 어떤 조건에서 작업을 수행할 수 있는지 정의하여 누가 무엇을 액세스할 수 있는지 지정합니다.

기본적으로 사용자 및 역할에는 어떠한 권한도 없습니다. IAM 관리자는 IAM 정책을 생성하고 사용자가 수임할 수 있는 역할에 추가합니다. IAM 정책은 작업을 수행하기 위해 사용하는 방법과 관계없이 작업에 대한 권한을 정의합니다.

ID 기반 정책

ID 기반 정책은 ID(사용자, 사용자 그룹 또는 역할)에 연결하는 JSON 권한 정책 문서입니다. 이러한 정책은 자격 증명이 수행할 수 있는 작업, 대상 리소스 및 이에 관한 조건을 제어합니다. ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서에서 [고객 관리형 정책으로 사용자 지정 IAM 권한 정의](#)를 참조하세요.

ID 기반 정책은 인라인 정책(단일 ID에 직접 포함) 또는 관리형 정책(여러 ID에 연결된 독립 실행형 정책)일 수 있습니다. 관리형 정책 또는 인라인 정책을 선택하는 방법을 알아보려면 IAM 사용 설명서의 [관리형 정책 및 인라인 정책 중에서 선택](#) 섹션을 참조하세요.

리소스 기반 정책

리소스 기반 정책은 리소스에 연결하는 JSON 정책 설명서입니다. 예를 들어 IAM 역할 신뢰 정책 및 Amazon S3 버킷 정책이 있습니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 리소스 기반 정책에서 [보안 주체를 지정](#)해야 합니다.

리소스 기반 정책은 해당 서비스에 있는 인라인 정책입니다. 리소스 기반 정책에서는 IAM의 AWS 관리형 정책을 사용할 수 없습니다.

기타 정책 유형

AWS 는 보다 일반적인 정책 유형에서 부여한 최대 권한을 설정할 수 있는 추가 정책 유형을 지원합니다.

- 권한 경계 - ID 기반 정책에서 IAM 엔터티에 부여할 수 있는 최대 권한을 설정합니다. 자세한 정보는 IAM 사용 설명서의 [IAM 엔터티의 권한 범위](#)를 참조하세요.
- 서비스 제어 정책(SCP) - AWS Organizations내 조직 또는 조직 단위에 대한 최대 권한을 지정합니다. 자세한 내용은 AWS Organizations 사용 설명서의 [서비스 제어 정책](#)을 참조하세요.
- 리소스 제어 정책(RCP) - 계정의 리소스에 사용할 수 있는 최대 권한을 설정합니다. 자세한 내용은 AWS Organizations 사용 설명서의 [리소스 제어 정책\(RCP\)](#)을 참조하세요.
- 세션 정책 - 역할 또는 페더레이션 사용자에게 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 자세한 내용은 IAM 사용 설명서의 [세션 정책](#)을 참조하세요.

여러 정책 유형

여러 정책 유형이 요청에 적용되는 경우, 결과 권한은 이해하기가 더 복잡합니다. 에서 여러 정책 유형이 관련될 때 요청을 허용할지 여부를 AWS 결정하는 방법을 알아보려면 IAM 사용 설명서의 [정책 평가 로직](#)을 참조하세요.

Amazon Transcribe 에서 IAM을 사용하는 방법

IAM을 사용하여에 대한 액세스를 관리하기 전에 사용할 수 있는 IAM 기능에 대해 Amazon Transcribe 알아봅니다 Amazon Transcribe.

IAM 에서 사용할 수 있는 기능 Amazon Transcribe

IAM 기능	Amazon Transcribe 지원
자격 증명 기반 정책	예
리소스 기반 정책	아니요
정책 작업	예
정책 리소스	예
정책 조건 키(서비스별)	예

IAM 기능	Amazon Transcribe 지원
ACL	아니요
ABAC(정책 내 태그)	부분적
임시 자격 증명	예
엔터티 권한	예
서비스 역할	예
서비스 연결 역할	아니요

Amazon Transcribe 및 기타 AWS 서비스에서 대부분의 IAM 기능을 사용하는 방법을 개괄적으로 알아보려면 IAM 사용 설명서의 [AWS 로 작업하는 서비스를 IAM](#) 참조하세요.

에 대한 자격 증명 기반 정책 Amazon Transcribe

ID 기반 정책 지원: 예

ID 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 ID에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자 및 역할이 어떤 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 제어합니다. 자격 증명 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서에서 [고객 관리형 정책으로 사용자 지정 IAM 권한 정의](#)를 참조하세요.

IAM ID 기반 정책을 사용하면 허용되거나 거부되는 작업과 리소스뿐 아니라 작업이 허용되거나 거부되는 조건을 지정할 수 있습니다. JSON 정책에서 사용할 수 있는 모든 요소에 대해 알아보려면 IAM 사용 설명서의 [IAM JSON 정책 요소 참조](#)를 참조하세요.

에 대한 자격 증명 기반 정책 예제 Amazon Transcribe

자격 Amazon Transcribe 증명 기반 정책의 예를 보려면 섹션을 참조하세요 [Amazon Transcribe 자격 증명 기반 정책 예제](#).

내의 리소스 기반 정책 Amazon Transcribe

리소스 기반 정책 지원: 아니요

리소스 기반 정책은 리소스에 연결하는 JSON 정책 설명서입니다. 리소스 기반 정책의 예제는 IAM 역할 신뢰 정책과 Amazon S3 버킷 정책입니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자

는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 정책이 연결된 리소스의 경우 정책은 지정된 보안 주체가 해당 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 [보안 주체를 지정](#)해야 합니다. 보안 주체에는 계정, 사용자, 역할, 페더레이션 사용자 또는 이 포함될 수 있습니다 AWS 서비스.

교차 계정 액세스를 활성화하려는 경우, 전체 계정이나 다른 계정의 IAM 개체를 리소스 기반 정책의 보안 주체로 지정할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 [IAM에서 교차 계정 리소스 액세스](#)를 참조하세요.

에 대한 정책 작업 Amazon Transcribe

정책 작업 지원: 예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

JSON 정책의 Action요소는 정책에서 액세스를 허용하거나 거부하는 데 사용할 수 있는 작업을 설명합니다. 연결된 작업을 수행할 수 있는 권한을 부여하기 위한 정책에 작업을 포함하세요.

Amazon Transcribe 작업 목록을 보려면 서비스 승인 참조의에서 [정의한 작업을 Amazon Transcribe](#) 참조하세요.

의 정책 작업은 작업 앞에 transcribe 접두사를 Amazon Transcribe 사용합니다. 단일 문에서 여러 작업을 지정하려면 쉼표로 구분합니다.

```
"Action": [
    "transcribe:action1",
    "transcribe:action2"
]
```

와일드카드(*)를 사용하여 여러 작업을 지정할 수 있습니다. 예를 들어, List라는 단어로 시작하는 모든 작업을 지정하려면 다음 작업을 포함합니다.

```
"Action": "transcribe:List*"
```

자격 Amazon Transcribe 증명 기반 정책의 예를 보려면 섹션을 참조하세요 [Amazon Transcribe 자격 증명 기반 정책 예제](#).

에 대한 정책 리소스 Amazon Transcribe

정책 리소스 지원: 예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

Resource JSON 정책 요소는 작업이 적용되는 하나 이상의 객체를 지정합니다. 모범 사례에 따라 [Amazon 리소스 이름\(ARN\)](#)을 사용하여 리소스를 지정합니다. 리소스 수준 권한을 지원하지 않는 작업의 경우, 와일드카드(*)를 사용하여 해당 문이 모든 리소스에 적용됨을 나타냅니다.

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

Amazon Transcribe 리소스 유형 및 해당 ARNs 목록을 보려면 서비스 승인 참조의에서 [정의한 리소스를 Amazon Transcribe](#) 참조하세요. 각 리소스의 ARN을 지정할 수 있는 작업을 알아보려면 [Amazon Transcribe에서 정의한 작업](#)을 참조하세요.

자격 Amazon Transcribe 증명 기반 정책의 예를 보려면 섹션을 참조하세요 [Amazon Transcribe 자격 증명 기반 정책 예제](#).

에 대한 정책 조건 키 Amazon Transcribe

서비스별 정책 조건 키 지원: 예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

Condition 요소는 정의된 기준에 따라 문이 실행되는 시기를 지정합니다. 같음(equals) 또는 미만(less than)과 같은 [조건 연산자](#)를 사용하여 정책의 조건을 요청의 값과 일치시키는 조건식을 생성할 수 있습니다. 모든 AWS 전역 조건 키를 보려면 IAM 사용 설명서의 [AWS 전역 조건 컨텍스트 키](#)를 참조하세요.

Amazon Transcribe 조건 키 목록을 보려면 서비스 승인 참조의에 [대한 조건 키를 Amazon Transcribe](#) 참조하세요. 조건 키를 사용할 수 있는 작업과 리소스를 알아보려면 [에서 정의한 작업을 Amazon Transcribe](#) 참조하세요.

자격 Amazon Transcribe 증명 기반 정책의 예를 보려면 섹션을 참조하세요 [Amazon Transcribe 자격 증명 기반 정책 예제](#).

ACLs Amazon Transcribe

ACL 지원: 아니요

액세스 제어 목록(ACL)은 어떤 보안 주체(계정 멤버, 사용자 또는 역할)가 리소스에 액세스할 수 있는 권한을 가지고 있는지를 제어합니다. ACL은 JSON 정책 문서 형식을 사용하지 않지만 리소스 기반 정책과 유사합니다.

를 사용한 ABAC Amazon Transcribe

ABAC 지원(정책의 태그): 부분적

속성 기반 액세스 제어(ABAC)는 태그라고 불리는 속성을 기반으로 권한을 정의하는 권한 부여 전략입니다. IAM 엔터티 및 AWS 리소스에 태그를 연결한 다음 보안 주체의 태그가 리소스의 태그와 일치할 때 작업을 허용하는 ABAC 정책을 설계할 수 있습니다.

태그에 근거하여 액세스를 제어하려면 `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` 또는 `aws:TagKeys` 조건 키를 사용하여 정책의 [조건 요소](#)에 태그 정보를 제공합니다.

서비스가 모든 리소스 유형에 대해 세 가지 조건 키를 모두 지원하는 경우, 값은 서비스에 대해 예입니다. 서비스가 일부 리소스 유형에 대해서만 세 가지 조건 키를 모두 지원하는 경우, 값은 부분적입니다.

ABAC에 대한 자세한 내용은 IAM 사용 설명서의 [ABAC 권한 부여를 통한 권한 정의](#)를 참조하세요. ABAC 설정 단계가 포함된 자습서를 보려면 IAM 사용 설명서의 [속성 기반 액세스 제어\(ABAC\) 사용](#)을 참조하세요.

Amazon Transcribe 리소스 태그 지정에 대한 자세한 내용은 [섹션을 참조하세요 리소스에 태그 지정](#). 태그 기반 액세스 제어에 대한 자세한 내용은 [태그를 사용한 AWS 리소스 액세스 제어](#)를 참조하세요.

에서 임시 자격 증명 사용 Amazon Transcribe

임시 자격 증명 지원: 예

임시 자격 증명은 AWS 리소스에 대한 단기 액세스를 제공하며 페더레이션 또는 전환 역할을 사용할 때 자동으로 생성됩니다. 장기 액세스 키를 사용하는 대신 임시 자격 증명을 동적으로 생성하는 것이 AWS 좋습니다. 자세한 내용은 IAM 사용 설명서의 [IAM의 임시 보안 자격 증명](#) 및 [IAM으로 작업하는 AWS 서비스](#) 섹션을 참조하세요.

에 대한 교차 서비스 보안 주체 권한 Amazon Transcribe

전달 액세스 세션(FAS) 지원: 예

전달 액세스 세션(FAS)은 호출하는 보안 주체의 권한을 다운스트림 서비스에 AWS 서비스 대한 요청과 AWS 서비스 함께 사용합니다. FAS 요청 시 정책 세부 정보는 [전달 액세스 세션](#)을 참조하세요.

에 대한 서비스 역할 Amazon Transcribe

서비스 역할 지원: 예

서비스 역할은 서비스가 사용자를 대신하여 작업을 수행하는 것으로 가정하는 [IAM 역할](#)입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 [AWS 서비스 AWS에 권한을 위임할 역할 생성](#)을 참조하세요.

Warning

서비스 역할에 대한 권한을 변경하면 Amazon Transcribe 기능이 중단될 수 있습니다. 에서 관련 지침을 Amazon Transcribe 제공하는 경우에만 서비스 역할을 편집합니다.

에 대한 서비스 연결 역할 Amazon Transcribe

서비스 연결 역할 지원: 아니요

서비스 연결 역할은 연결된 서비스 역할의 한 유형입니다 AWS 서비스. 서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은 표시 AWS 계정 되며 서비스가 소유합니다. IAM 관리자는 서비스 연결 역할의 권한을 볼 수 있지만 편집은 할 수 없습니다.

Amazon Transcribe 는 서비스 연결 역할을 지원하지 않습니다.

다른 서비스에 대한 서비스 연결 역할을 생성하거나 관리하는 방법에 대한 자세한 내용은 [AWS 에서 작업하는 서비스를 IAM](#) 참조하세요. 서비스 연결 역할 열에서 Yes가 포함된 서비스를 테이블에서 찾습니다. 해당 서비스에 대한 서비스 연결 역할 설명서를 보려면 예(Yes) 링크를 선택합니다.

교차 서비스 혼동된 대리인 방지

혼동된 대리인은 다른 엔터티에 의해 작업을 수행하도록 강요받는 엔터티(서비스 또는 계정)입니다. 이러한 유형의 가장은 교차 계정과 교차 서비스에서 발생할 수 있습니다.

혼동된 대리자를 방지하기 위해 AWS에서는 AWS 계정의 리소스에 대한 액세스 권한이 부여된 서비스 보안 주체를 사용하여 모든 서비스에 대한 데이터를 보호하는 데 도움이 되는 도구를 제공합니다. 이 섹션에서는 Amazon Transcribe과 관련된 교차 서비스 혼동 대리자 예방을 주로 설명합니다. 그러나 IAM 사용 설명서의 [혼동된 대리자 문제](#) 섹션에서 이 주제를 자세히 알아볼 수 있습니다.

리소스에 액세스하기 위해 IAM이 Amazon Transcribe에 부여하는 권한을 제한하려면 리소스 정책에 전역 조건 컨텍스트 키 [aws:SourceArn](#) 및 [aws:SourceAccount](#)을 사용하는 것이 좋습니다.

이러한 두 전역 조건 컨텍스트 키와 AWS 계정 ID를 포함하는 `aws:SourceArn` 값을 모두 사용하는 경우, `aws:SourceAccount` 값 및 `aws:SourceArn`의 AWS 계정은 동일한 정책 명령문에서 사용할 경우 반드시 동일한 AWS 계정 ID를 사용해야 합니다.

하나의 리소스만 교차 서비스 액세스와 연결되게 하려는 경우 `aws:SourceArn`을 사용합니다. 해당 AWS 계정의 모든 리소스를 교차 서비스 액세스와 연결하려면 하려면 `aws:SourceAccount`을 사용합니다.

Note

혼동된 대리자 문제로부터 보호하는 가장 효과적인 방법은 리소스의 전체 ARN이 포함된 `aws:SourceArn` 전역 조건 컨텍스트 키를 사용하는 것입니다. 전체 ARN을 모를 경우 또는 여러 리소스를 지정하는 경우, ARN의 알 수 없는 부분에 와일드 카드(*)를 포함한 `aws:SourceArn` 전역 조건 컨텍스트 키를 사용합니다. 예를 들어 `arn:aws:transcribe::123456789012:*`입니다.

혼동된 대리자 문제를 예방하는 방법을 보여주는 수임 역할 정책의 예는 [대리자 혼동 방지 정책](#)을 참조하세요.

Amazon Transcribe 자격 증명 기반 정책 예제

기본적으로 사용자 및 역할에는 Amazon Transcribe 리소스를 생성하거나 수정할 수 있는 권한이 없습니다. 사용자에게 사용자가 필요한 리소스에서 작업을 수행할 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성하면 됩니다.

이러한 예제 JSON 정책 문서를 사용하여 IAM ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 [IAM 정책 생성\(콘솔\)](#)을 참조하세요.

각 리소스 유형에 대한 ARN 형식을 포함하여 ACM에서 정의한 작업 및 리소스 유형에 대한 자세한 내용은 서비스 권한 부여 참조의 [Amazon Transcribe에 대한 작업, 리소스 및 조건 키](#)를 참조하세요.

주제

- [정책 모범 사례](#)
- [사용 AWS Management Console](#)

- [IAM 역할에 필요한 권한](#)
- [Amazon S3 암호화 키에 필요한 권한](#)
- [사용자가 자신의 고유한 권한을 볼 수 있도록 허용](#)
- [AWS KMS 암호화 컨텍스트 정책](#)
- [대리자 혼동 방지 정책](#)
- [태그를 기준으로 트랜스크립션 작업 보기](#)

정책 모범 사례

자격 증명 기반 정책에 따라 계정에서 사용자가 Amazon Transcribe 리소스를 생성, 액세스 또는 삭제할 수 있는지 여부가 결정됩니다. 이 작업으로 인해 AWS 계정에 비용이 발생할 수 있습니다. ID 기반 정책을 생성하거나 편집할 때는 다음 지침과 권장 사항을 따르세요.

- AWS 관리형 정책을 시작하고 최소 권한으로 전환 - 사용자 및 워크로드에 권한 부여를 시작하려면 많은 일반적인 사용 사례에 대한 권한을 부여하는 AWS 관리형 정책을 사용합니다. 에서 사용할 수 있습니다 AWS 계정. 사용 사례에 맞는 AWS 고객 관리형 정책을 정의하여 권한을 추가로 줄이는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 [AWS 관리형 정책](#) 또는 [AWS 직무에 대한 관리형 정책](#)을 참조하세요.
- 최소 권한 적용 - IAM 정책을 사용하여 권한을 설정하는 경우, 작업을 수행하는 데 필요한 권한만 부여합니다. 이렇게 하려면 최소 권한으로 알려진 특정 조건에서 특정 리소스에 대해 수행할 수 있는 작업을 정의합니다. IAM을 사용하여 권한을 적용하는 방법에 대한 자세한 정보는 IAM 사용 설명서에 있는 [IAM의 정책 및 권한](#)을 참조하세요.
- IAM 정책의 조건을 사용하여 액세스 추가 제한 - 정책에 조건을 추가하여 작업 및 리소스에 대한 액세스를 제한할 수 있습니다. 예를 들어, SSL을 사용하여 모든 요청을 전송해야 한다고 지정하는 정책 조건을 작성할 수 있습니다. AWS 서비스와 같은 특징을 통해 사용되는 경우 조건을 사용하여 서비스 작업에 대한 액세스 권한을 부여할 수도 있습니다 CloudFormation. 자세한 내용은 IAM 사용 설명서의 [IAM JSON 정책 요소: 조건](#)을 참조하세요.
- IAM Access Analyzer를 통해 IAM 정책을 확인하여 안전하고 기능적인 권한 보장 - IAM Access Analyzer에서는 IAM 정책 언어(JSON)와 모범 사례가 정책에서 준수되도록 새로운 및 기존 정책을 확인합니다. IAM Access Analyzer는 100개 이상의 정책 확인 항목과 실행 가능한 추천을 제공하여 안전하고 기능적인 정책을 작성하도록 돕습니다. 자세한 내용은 IAM 사용 설명서의 [IAM Access Analyzer에서 정책 검증](#)을 참조하세요.
- 다중 인증(MFA) 필요 -에서 IAM 사용자 또는 루트 사용자가 필요한 시나리오가 있는 경우 추가 보안을 위해 MFA를 AWS 계정입니다. API 작업을 직접적으로 호출할 때 MFA가 필요하다면 정책에 MFA 조건을 추가합니다. 자세한 내용은 IAM 사용 설명서의 [MFA를 통한 보안 API 액세스](#)를 참조하세요.

IAM의 모범 사례에 대한 자세한 내용은 IAM 사용 설명서의 [IAM의 보안 모범 사례](#)를 참조하세요.

사용 AWS Management Console

Amazon Transcribe 콘솔에 액세스하려면 최소한의 권한 집합이 있어야 합니다. 이러한 권한은의 Amazon Transcribe 리소스에 대한 세부 정보를 나열하고 볼 수 있도록 허용해야 합니다 AWS 계정. 최소 필수 권한보다 더 제한적인 ID 기반 정책을 생성하는 경우, 콘솔이 해당 정책에 연결된 엔티티(사용자 또는 역할)에 대해 의도대로 작동하지 않습니다.

AWS CLI 또는 AWS API만 호출하는 사용자에게는 최소 콘솔 권한을 허용할 필요가 없습니다. 대신, 수행하려는 API 작업과 일치하는 작업에만 액세스할 수 있도록 합니다.

엔티티(사용자 및 역할)가를 사용할 수 있도록 하려면 다음 AWS관리형 정책 중 [AWS Management Console](#) 하나를 연결합니다.

- `AmazonTranscribeFullAccess`: 모든 Amazon Transcribe 리소스를 생성, 읽기, 업데이트, 삭제 및 실행할 수 있는 전체 액세스 권한을 부여합니다. 버킷 이름에 `transcribe`가 있는 Amazon S3 버킷에 대한 액세스도 허용합니다.
- `AmazonTranscribeReadOnlyAccess`: Amazon Transcribe 리소스에 대한 읽기 전용 액세스 권한을 부여하면 사용자가 트랜스크립션 작업과 사용자 지정 어휘를 가져와서 나열할 수 있습니다.

Note

IAM AWS Management Console 에 로그인하고 정책 이름으로 검색하여 관리형 권한 정책을 검토할 수 있습니다. "transcribe"를 검색하면 위에 나열된 두 정책 (`AmazonTranscribeReadOnly` 및 `AmazonTranscribeFullAccess`)이 모두 반환됩니다.

자체 사용자 지정 IAM 정책을 생성하여 Amazon Transcribe API 작업에 대한 권한을 허용할 수도 있습니다. 지정된 권한이 필요한 엔티티에 이러한 사용자 지정 정책을 연결할 수 있습니다.

IAM 역할에 필요한 권한

호출할 IAM 역할을 생성하는 경우 Amazon S3 버킷에 액세스할 수 있는 권한이 Amazon Transcribe 있어야 합니다. 해당하는 경우 KMS key 를 사용하여 버킷의 콘텐츠를 암호화해야 합니다. 절차 예시는 다음 섹션을 참조하세요.

신뢰 정책

트랜스크립션 요청을 하는 데 사용하는 IAM 엔터티에는가 해당 역할을 수임 Amazon Transcribe 할 수 있도록 하는 신뢰 정책이 있어야 합니다. 다음 Amazon Transcribe 신뢰 정책을 사용합니다. 단, 통화 후 분석을 활성화한 상태에서 실시간 Call Analytics를 요청하려면 '실시간 Call Analytics에 대한 신뢰 정책'을 사용해야 합니다.

에 대한 신뢰 정책 Amazon Transcribe

JSON

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

실시간 Call Analytics에 대한 신뢰 정책

JSON

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

Amazon S3 입력 버킷 정책

다음 정책은 IAM 역할에 지정된 입력 버킷의 파일에 액세스할 수 있는 권한을 부여합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "s3:GetObject",
      "s3:ListBucket"
    ],
```

```

    "Resource": [
      "arn:aws:s3:::DOC-EXAMPLE-INPUT-BUCKET",
      "arn:aws:s3:::DOC-EXAMPLE-INPUT-BUCKET/*"
    ]
  }
}

```

Amazon S3 출력 버킷 정책

다음 정책은 지정된 출력 버킷에 파일을 쓸 수 있는 권한을 IAM 역할에 부여합니다.

JSON

```

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

```

Amazon S3 암호화 키에 필요한 권한

KMS key 를 사용하여 Amazon S3 버킷을 암호화하는 경우 KMS key 정책에 다음을 포함합니다. 이렇게 하면 버킷의 콘텐츠에 Amazon Transcribe 액세스할 수 있습니다. 에 대한 액세스 허용에 대한 자세한 내용은 AWS KMS 개발자 안내서의 [외부에서 액세스 AWS 계정 허용 KMS key](#)를 KMS keys 참조 하세요.

JSON

```

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

```

```

    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
    },
    "Action": [
      "kms:Decrypt"
    ],
    "Resource": "arn:aws:kms:us-west-2:111122223333:key/KMS-Example-KeyId"
  }
]
}

```

사용자가 자신의 고유한 권한을 볼 수 있도록 허용

이 예제는 IAM 사용자가 자신의 사용자 ID에 연결된 인라인 및 관리형 정책을 볼 수 있도록 허용하는 정책을 생성하는 방법을 보여줍니다. 이 정책에는 콘솔에서 또는 AWS CLI 또는 AWS API를 사용하여 프로그래밍 방식으로 이 작업을 완료할 수 있는 권한이 포함됩니다.

```

{
  "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 암호화 컨텍스트 정책

다음 정책은 이 특징에 대해 AWS KMS 암호 해독 및 암호화 작업을 사용할 수 있는 권한을 IAM 역할 "ExampleRole"에 부여합니다 KMS key. 이 정책은 하나 이상의 암호화 컨텍스트 쌍(이 경우 "color:indigoBlue")이 있는 요청에만 적용됩니다. AWS KMS 암호화 컨텍스트에 대한 자세한 내용은 섹션을 참조하세요 [AWS KMS 암호화 컨텍스트](#).

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
      },
      "Action": [
        "kms:Decrypt",
        "kms:Encrypt",
        "kms:GenerateDataKey*",
        "kms:ReEncrypt*"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:EncryptionContext:color": "indigoBlue"
        }
      }
    },
    {
      "Effect": "Allow",
      "Principal": {

```

```

        "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
    },
    "Action": "kms:DescribeKey",
    "Resource": "*"
  }
]
}

```

대리자 혼동 방지 정책

다음은 혼동된 대리자 문제를 방지하기 Amazon Transcribe 위해 `aws:SourceArn` 및 `aws:SourceAccount` 함께 사용하는 방법을 보여주는 역할 수임 정책의 예입니다. 대리자 혼동 방지에 대한 자세한 내용은 [교차 서비스 혼동된 대리인 방지](#) 섹션을 참조하세요.

JSON

```

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

```

태그를 기준으로 트랜스크립션 작업 보기

자격 증명 기반 정책의 조건을 사용하여 태그를 기반으로 Amazon Transcribe 리소스에 대한 액세스를 제어할 수 있습니다. 이 예시에서는 트랜스크립션 작업 보기를 허용하는 정책을 생성할 수 있는 방법을 보여줍니다. 그러나 트랜스크립션 작업 태그 Owner에 해당 사용자의 사용자 이름 값이 있는 경우에만 권한이 부여됩니다. 이 정책은 AWS Management Console을 사용하여 이 작업을 완료하는 데 필요한 권한도 부여합니다.

이 정책을 계정의 IAM 엔터티에 연결할 수 있습니다. test-role 역할이 트랜스크립션 작업을 보려고 시도하는 경우, 트랜스크립션 작업에 Owner=test-role 또는 owner=test-role 태그(조건 키 이름은 대소문자를 구분하지 않음)를 지정해야 합니다. 그렇지 않으면 액세스가 거부됩니다. 자세한 내용은 IAM 사용 설명서의 [IAM JSON 정책 요소: 조건](#)을 참조하세요.

태그 지정에 대한 자세한 내용은 섹션을 Amazon Transcribe참조하세요 [리소스에 태그 지정](#).

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ListTranscriptionJobsInConsole",
      "Effect": "Allow",
      "Action": "transcribe:ListTranscriptionJobs",
      "Resource": "*"
    },
    {
      "Sid": "ViewTranscriptionJobsIfOwner",
      "Effect": "Allow",
      "Action": "transcribe:GetTranscriptionJob",
      "Resource": "arn:aws:transcribe:us-west-2:111122223333:transcription-
job/*",
      "Condition": {
        "StringEquals": {"aws:ResourceTag/Owner": "${aws:username}"}
      }
    }
  ]
}
```

Amazon Transcribe 자격 증명 및 액세스 문제 해결

다음 정보를 사용하여 Amazon Transcribe 및 AWS Identity and Access Management (IAM)로 작업할 때 발생할 수 있는 일반적인 문제를 진단하고 수정할 수 있습니다.

주제

- [에서 작업을 수행할 권한이 없음 Amazon Transcribe](#)
- [iam:PassRole을 수행하도록 인증되지 않음](#)
- [내 외부의 사람이 내 Amazon Transcribe 리소스에 액세스 AWS 계정 하도록 허용하고 싶음](#)

에서 작업을 수행할 권한이 없음 Amazon Transcribe

작업을 수행할 권한이 없다는 오류가 표시되면 작업을 수행할 수 있도록 정책을 업데이트해야 합니다.

다음의 예제 오류는 mateojackson IAM 사용자가 콘솔을 사용하여 가상 *my-example-widget* 리소스에 대한 세부 정보를 보려고 하지만 가상 *transcribe:GetWidget* 권한이 없을 때 발생합니다.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
transcribe:GetWidget on resource: my-example-widget
```

이 경우, *transcribe:GetWidget* 작업을 사용하여 *my-example-widget* 리소스에 액세스할 수 있도록 mateojackson 사용자 정책을 업데이트해야 합니다.

도움이 필요한 경우 AWS 관리자에게 문의하세요. 관리자는 로그인 자격 증명을 제공한 사람입니다.

iam:PassRole을 수행하도록 인증되지 않음

iam:PassRole 작업을 수행할 수 있는 권한이 없다는 오류가 수신되면 Amazon Transcribe에 역할을 전달할 수 있도록 정책을 업데이트해야 합니다.

일부 AWS 서비스에서는 새 서비스 역할 또는 서비스 연결 역할을 생성하는 대신 기존 역할을 해당 서비스에 전달할 수 있습니다. 이렇게 하려면 역할을 서비스에 전달할 권한이 있어야 합니다.

다음 예 오류는 marymajor라는 IAM 사용자가 콘솔을 사용하여 Amazon Transcribe에서 작업을 수행하려고 하는 경우에 발생합니다. 하지만 작업을 수행하려면 서비스 역할이 부여한 권한이 서비스에 있어야 합니다. Mary는 서비스에 역할을 전달할 권한이 없습니다.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

이 경우, Mary가 iam:PassRole작업을 수행할 수 있도록 Mary의 정책을 업데이트해야 합니다.

도움이 필요한 경우 AWS 관리자에게 문의하세요. 관리자는 로그인 자격 증명을 제공한 사람입니다.

내 외부의 사람이 내 Amazon Transcribe 리소스에 액세스 AWS 계정 하도록 허용하고 싶음

다른 계정의 사용자 또는 조직 외부의 사람이 리소스에 액세스할 때 사용할 수 있는 역할을 생성할 수 있습니다. 역할을 수임할 신뢰할 수 있는 사람을 지정할 수 있습니다. 리소스 기반 정책 또는 액세스 제어 목록(ACL)을 지원하는 서비스의 경우, 이러한 정책을 사용하여 다른 사람에게 리소스에 대한 액세스 권한을 부여할 수 있습니다.

자세한 내용은 다음을 참조하세요.

- 에서 이러한 기능을 Amazon Transcribe 지원하는지 여부를 알아보려면 섹션을 참조하세요 [Amazon Transcribe 에서 IAM을 사용하는 방법](#).
- 소유 AWS 계정 한의 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 [IAM 사용 설명서의 소유한 다른의 IAM 사용자에게 액세스 권한 제공을 참조 AWS 계정 하세요](#).
- 리소스에 대한 액세스 권한을 타사에 제공하는 방법을 알아보려면 IAM 사용 설명서의 [타사 AWS 계정 소유에 대한 액세스 권한 제공을](#) AWS 계정참조하세요.
- ID 페더레이션을 통해 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 [외부에서 인증된 사용자에게 액세스 권한 제공\(ID 페더레이션\)](#)을 참조하세요.
- 크로스 계정 액세스에 대한 역할과 리소스 기반 정책 사용의 차이점을 알아보려면 IAM 사용 설명서의 [IAM의 크로스 계정 리소스 액세스](#)를 참조하세요.

의 데이터 보호Amazon Transcribe

AWS [공동 책임 모델](#)은 Amazon Transcribe의 데이터 보호에 적용됩니다. 이 모델에서 설명하는 것처럼 AWS는 모든 AWS 클라우드를 실행하는 글로벌 인프라를 보호할 책임이 있습니다. 사용자는 인프라에서 호스팅되는 콘텐츠를 관리해야 합니다. 사용하는 AWS 서비스의 보안 구성과 관리 태스크에 대한 책임도 사용자에게 있습니다. 데이터 프라이버시에 대한 자세한 내용은 [데이터 프라이버시 FAQ](#)를 참조하세요. 유럽의 데이터 보호에 대한 자세한 내용은 AWS 보안 블로그의 [AWS 공동 책임 모델 및 GDPR](#) 블로그 게시물을 참조하세요.

데이터를 보호하려면 AWS 계정 자격 증명을 보호하고 AWS IAM Identity Center 또는 AWS Identity and Access Management(IAM)를 통해 개별 사용자 계정을 설정하는 것이 좋습니다. 이렇게 하면 개별 사용자에게 자신의 직무를 충실히 이행하는 데 필요한 권한만 부여됩니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정에 다중 인증(MFA)을 사용하세요.
- SSL/TLS를 사용하여 AWS 리소스와 통신하세요. TLS 1.2는 필수이며 TLS 1.3을 권장합니다.
- AWS CloudTrail로 API 및 사용자 활동 로깅을 설정하세요. AWS 활동 캡처에 CloudTrail 추적을 사용하는 방법에 대한 자세한 내용은 AWS CloudTrail 사용 설명서의 [CloudTrail 추적 작업을](#) 참조하세요.
- AWS 암호화 솔루션을 AWS 서비스 내의 모든 기본 보안 컨트롤과 함께 사용하세요.
- Amazon S3에 저장된 민감한 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용하세요.
- 명령줄 인터페이스 또는 API를 통해 AWS에 액세스할 때 FIPS 140-3 검증된 암호화 모듈이 필요한 경우, FIPS 엔드포인트를 사용합니다. 사용 가능한 FIPS 엔드포인트에 대한 자세한 내용은 [Federal Information Processing Standard\(FIPS\) 140-3](#)을 참조하세요.

고객의 이메일 주소와 같은 기밀 정보나 중요한 정보는 태그나 이름 필드와 같은 자유 형식 텍스트 필드에 입력하지 않는 것이 좋습니다. 여기에는 Amazon Transcribe 또는 기타 AWS 서비스에서 콘솔, API, AWS CLI 또는 AWS SDK를 사용하여 작업하는 경우가 포함됩니다. 이름에 사용되는 태그 또는 자유 형식 텍스트 필드에 입력하는 모든 데이터는 청구 또는 진단 로그에 사용될 수 있습니다. 외부 서버에 URL을 제공할 때 해당 서버에 대한 요청을 검증하기 위해 자격 증명을 URL에 포함해서는 안 됩니다.

인터넷워크 트래픽 개인 정보 보호

Amazon Virtual Private Cloud에 대한 Amazon VPC(Amazon Transcribe) 엔드포인트는 Amazon Transcribe에 대한 연결만 허용하는 VPC 내의 논리적 엔티티입니다. Amazon VPC는 요청을 Amazon Transcribe로 라우팅하고 응답을 다시 VPC로 라우팅합니다. 자세한 내용은 [AWS PrivateLink 개념](#)을 참조하세요. Amazon Transcribe와 Amazon VPC 엔드포인트 함께 사용에 대한 자세한 내용은 [Amazon Transcribe 및 인터페이스 VPC 엔드포인트\(AWS PrivateLink\)](#) 단원을 참조하세요.

데이터 암호화

데이터 암호화란 전송 중 데이터와 저장 데이터를 보호하는 것을 말합니다. 전송 중 표준 전송 계층 보안(TLS)과 함께 Amazon S3관리형 키를 사용하거나 유휴 KMS keys상태에서 데이터를 보호할 수 있습니다.

저장 시 암호화

Amazon Transcribe는 Amazon S3버킷에 배치된 트랜스크립트의 서버 측 암호화에 기본 Amazon S3 키(SSE-S3)를 사용합니다.

[StartTranscriptionJob](#) 작업을 사용할 때 트랜스크립션 작업의 출력을 암호화KMS key하도록 자체를 지정할 수 있습니다.

Amazon Transcribe는 기본 키로 암호화된 Amazon EBS볼륨을 사용합니다.

전송 중 암호화

Amazon Transcribe는 AWS인증서와 함께 TLS 1.2를 사용하여 전송 중인 데이터를 암호화합니다. 여기에는 스트리밍 트랜스크립션이 포함됩니다.

키 관리

Amazon Transcribe는와 함께 작동KMS keys하여 데이터에 대한 향상된 암호화를 제공합니다. 를 사용하면 트랜스크립션 작업을 생성할 때 입력 미디어를 암호화Amazon S3할 수 있습니다. 와 통합AWS KMS하면 [StartTranscriptionJob](#) 요청의 출력을 암호화할 수 있습니다.

를 지정하지 않으면 KMS key트랜스크립션 작업의 출력이 기본 Amazon S3키(SSE-S3)로 암호화됩니다.

에 대한 자세한 내용은 [AWS Key Management Service개발자 안내서](#)를 AWS KMS참조하세요.

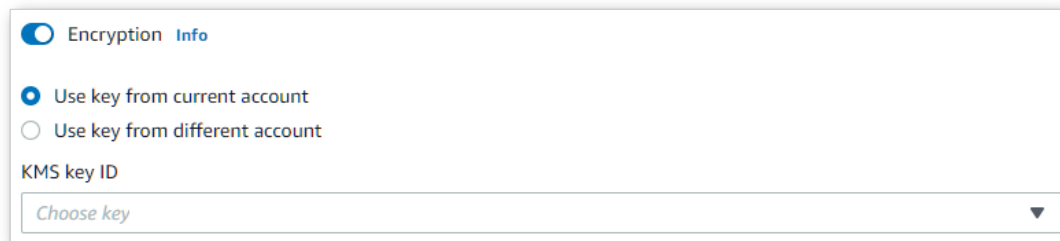
를 사용한 키 관리AWS Management Console

트랜스크립션 작업의 출력을 암호화하려면 요청을 AWS 계정수행하는에 KMS key대해를 사용하거나 다른 KMS key에서를 사용하도록 선택할 수 있습니다AWS 계정.

를 지정하지 않으면 KMS key트랜스크립션 작업의 출력이 기본 Amazon S3키(SSE-S3)로 암호화됩니다.

출력 암호화를 활성화하려면:

1. 출력 데이터에서 암호화를 선택합니다.



2. KMS key가 현재 사용 중인 AWS 계정의 인지 아니면 다른의 인지 선택합니다AWS 계정. 현재의 키를 사용하려면 KMS keyID에서 키를 AWS 계정선택합니다. 다른의 키를 사용하는 경우 키

의 ARN을 입력해야 AWS 계정입니다. 다른 키를 사용하려면 호출자에게 AWS 계정에 대한 kms:Encrypt 권한이 있어야 합니다 KMS key. 자세한 내용은 [키 정책 생성](#)을 참조하세요.

API를 사용하여 키 관리

API에서 출력 암호화를 사용하려면, [StartCallAnalyticsJob](#), [StartMedicalTranscriptionJob](#) 또는 [StartTranscriptionJob](#) 작업의 OutputEncryptionKMSKeyId 파라미터를 KMS key 사용하여 지정해야 합니다.

현재 AWS 계정에 있는 키를 사용하는 경우 다음 네 가지 방법 중 하나로 KMS key를 지정할 수 있습니다.

1. KMS keyID 자체를 사용합니다. 예를 들어 1234abcd-12ab-34cd-56ef-1234567890ab입니다.
2. KMS keyID에 별칭을 사용합니다. 예를 들어 alias/ExampleAlias입니다.
3. KMS keyID에 Amazon 리소스 이름(ARN)을 사용합니다. 예를 들어 arn:aws:kms:region:account-ID:key/1234abcd-12ab-34cd-56ef-1234567890ab입니다.
4. KMS key별칭에 ARN을 사용합니다. 예를 들어 arn:aws:kms:region:account-ID:alias/ExampleAlias입니다.

현재와 다른 AWS 계정에 있는 키를 사용하는 경우 다음 두 가지 방법 중 하나로 KMS key를 지정할 수 있습니다.

1. KMS keyID에 ARN을 사용합니다. 예를 들어 arn:aws:kms:region:account-ID:key/1234abcd-12ab-34cd-56ef-1234567890ab입니다.
2. KMS key별칭에 ARN을 사용합니다. 예를 들어 arn:aws:kms:region:account-ID:alias/ExampleAlias입니다.

단, 요청을 하는 주체는 지정한 KMS key를 사용할 권한이 있어야 합니다.

AWS KMS 암호화 컨텍스트

AWS KMS 암호화 컨텍스트는 일반 텍스트, 비밀이 아닌 키:값 페어의 맵입니다. 이 맵은 데이터에 대한 추가 보안 계층을 제공하는 암호화 컨텍스트 페어라고 하는 추가 인증 데이터를 나타냅니다.는 트랜스크립션 출력을 고객 지정 Amazon S3 버킷으로 암호화하려면 대칭 암호화 키가 Amazon Transcribe 필요합니다. 자세한 내용은 [AWS KMS의 비대칭 키](#)를 참조하세요.

암호화 컨텍스트 페어를 생성할 때 민감한 정보를 포함해서는 안 됩니다. 암호화 컨텍스트는 비밀이 아닙니다. CloudTrail로그 내에 일반 텍스트로 표시됩니다(암호화 작업을 식별하고 분류하는 데 사용할 수 있음).

암호화 컨텍스트 페어는 밑줄(_), 대시(-), 슬래시(/, \), 콜론(:) 같은 특수 문자를 포함할 수 있습니다.

Tip

암호화 컨텍스트 페어의 값을 암호화된 데이터와 연관시키는 것이 유용할 수 있습니다. 필수 사항은 아니지만 파일 이름, 헤더 값 또는 암호화되지 않은 데이터베이스 필드 등 암호화된 콘텐츠와 관련된 민감하지 않은 메타데이터를 사용하는 것이 좋습니다.

API와 함께 출력 암호화를 사용하려면 [StartTranscriptionJob](#) 작업의 `KMSEncryptionContext` 파라미터를 설정합니다. 출력 암호화 작업에 암호화 컨텍스트를 제공하려면 `OutputEncryptionKMSKeyId` 파라미터가 대칭 KMS key ID를 참조해야 합니다.

IAM정책과 함께 [AWS KMS조건 키](#)를 사용하여 암호화 [작업](#) 요청에 사용된 암호화 컨텍스트를 KMS key기반으로 대칭 암호화에 대한 액세스를 제어할 수 있습니다. 암호화 컨텍스트 정책의 예는 [AWS KMS 암호화 컨텍스트 정책](#)을 참조하세요.

암호화 컨텍스트는 선택 사항이지만 권장됩니다. 자세한 정보는 [암호화 컨텍스트](#)를 참조하세요.

서비스 개선을 위한 데이터 사용 거부

기본적으로 Amazon Transcribe는 서비스를 개발하고 사용자 경험을 지속적으로 개선하기 위해 처리한 음성 입력을 저장하고 사용합니다. AWS Organizations 옵트아웃 정책을 사용하여 콘텐츠가 Amazon Transcribe 개발 및 개선에 사용되는 것을 옵트아웃할 수 있습니다. 옵트아웃 방법에 대한 자세한 내용은 [AI 서비스 옵트아웃 정책](#)을 참조하세요.

모니터링 Amazon Transcribe

모니터링은 Amazon Transcribe 및 다른 AWS 솔루션의 신뢰성, 가용성 및 성능을 유지하는 데 중요한 부분입니다.는 다음과 같은 모니터링 도구를 AWS 제공하여 모니터링 Amazon Transcribe, 보고 및 문제 발생 시 적절한 경우 자동 조치를 취합니다.

- Amazon CloudWatch는 리소스 AWS 와 AWS 에서 실시간으로 실행하는 애플리케이션을 모니터링합니다. 지표를 수집 및 추적하고, 사용자 지정 대시보드를 생성할 수 있으며, 지정된 지표가 지정된 임계값에 도달하면 사용자에게 알리거나 조치를 취하도록 경보를 설정할 수 있습니다. 예를 들어

Amazon EC2 인스턴스에서 CPU 사용량 또는 기타 지표를 CloudWatch 추적하고 필요한 경우 새 인스턴스를 자동으로 시작할 수 있습니다.

- Amazon CloudWatch Logs는 Amazon EC2 인스턴스 및 기타 소스에서 로그 파일을 모니터링 CloudTrail, 저장 및 액세스할 수 있습니다.는 로그 파일의 정보를 모니터링하고 특정 임계값이 충족 되면 사용자에게 알릴 CloudWatch Logs 수 있습니다. 또한 매우 내구성이 뛰어난 스토리지에 로그 데이터를 저장할 수 있습니다.
- AWS CloudTrail는에 의해 또는를 대신하여 수행된 API 호출 및 관련 이벤트를 캡처 AWS 계정 하고 지정한 버킷에 Amazon S3 로그 파일을 전송합니다. 호출한 사용자 및 계정 AWS, 호출이 수행된 소스 IP 주소, 호출이 발생한 시기를 식별할 수 있습니다.

자세한 내용은 [Amazon CloudWatch 사용 설명서](#)를 참조하십시오.

Amazon EventBridge는 이벤트를 사용하여 애플리케이션 구성 요소를 함께 연결하는 서버리스 서비스로, 확장 가능한 이벤트 기반 애플리케이션을 더 쉽게 빌드할 수 있습니다.는 자체 애플리케이션, 서비스형 소프트웨어(SaaS) 애플리케이션 및 AWS 서비스에서 실시간 데이터 스트림을 EventBridge 제공하고 해당 데이터를와 같은 대상으로 라우팅합니다 Lambda. 서비스에서 발생하는 이벤트를 모니터링하고 이벤트 기반 아키텍처를 구축할 수 있습니다. 자세한 내용은 [Amazon EventBridge 사용 설명서](#)를 참조하십시오.

주제

- [Amazon Transcribe 를 사용한 모니터링 Amazon CloudWatch](#)
- [Amazon Transcribe 를 사용한 모니터링 AWS CloudTrail](#)
- [Amazon Transcribe과 함께 Amazon EventBridge 사용](#)

Amazon Transcribe 를 사용한 모니터링 Amazon CloudWatch

원시 데이터를 수집 CloudWatch하여 읽기 가능하며 실시간에 가까운 지표로 처리하는를 Amazon Transcribe 사용하여 모니터링할 수 있습니다. 이러한 통계는 15개월간 보관되므로 기록 정보에 액세스하고 웹 애플리케이션 또는 서비스가 어떻게 실행되고 있는지 전체적으로 더 잘 파악할 수 있습니다. 특정 임계값을 주시하다가 해당 임계값이 충족될 때 알림을 전송하거나 조치를 취하도록 경보를 설정할 수도 있습니다. 자세한 내용은 [CloudWatch 사용 설명서](#)를 참조하세요.

에서 Amazon CloudWatch 지표 및 차원 사용 Amazon Transcribe

Amazon Transcribe 는 성능 모니터링에 도움이 되는 데이터인 CloudWatch 지표와 차원을 지원합니다. 지원되는 지표 범주에는 트랜스크립션 작업과 관련된 트래픽, 오류, 데이터 전송, 지연 시간이 포함됩니다. 지원되는 지표는 AWS/Transcribe 네임스페이스 CloudWatch 에서를 통해 찾을 수 있습니다.

Note

CloudWatch 모니터링 지표는 무료이며 CloudWatch 서비스 할당량에 포함되지 않습니다.

CloudWatch 지표에 대한 자세한 내용은 [Amazon CloudWatch 지표 사용](#)을 참조하세요.

Amazon Transcribe 를 사용한 모니터링 AWS CloudTrail

Amazon Transcribe 는 AWS Identity and Access Management (IAM) 사용자 또는 역할 또는 AWS service. CloudTrail capture가 Amazon Transcribe 에 대해 수행한 작업의 레코드를 제공하는 AWS CloudTrail서비스와 통합됩니다 Amazon Transcribe. 여기에는의 호출 AWS Management Console 과 Amazon Transcribe APIs에 대한 코드 호출이 이벤트로 포함됩니다. 추적을 생성하면에 대한 CloudTrail 이벤트를 포함한 이벤트를 버킷 Amazon Transcribe에 지속적으로 전달할 수 있습니다 Amazon S3 . 추적을 생성하지 않은 경우 이벤트 기록에서 CloudTrail AWS Management Console 콘솔의 최신 이벤트를 볼 수도 있습니다. 에서 수집한 정보를 사용하여 수행된 각 요청 CloudTrail, 요청이 수행된 Amazon Transcribe IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 추가 세부 정보를 볼 수 있습니다.

자세한 내용은 [AWS CloudTrail 사용 설명서](#)를 CloudTrail참조하세요.

Amazon Transcribe 및 CloudTrail

CloudTrail 는 계정을 생성할 AWS 계정 때에서 활성화됩니다. 활동이에서 발생하면 Amazon Transcribe해당 활동은 CloudTrail 이벤트 CloudTrail 기록의 다른 AWS 서비스 이벤트와 함께 이벤트에 기록됩니다. 에서 최근 이벤트를 보고 검색하고 다운로드할 수 있습니다 AWS 계정. 자세한 내용은 [CloudTrail 이벤트 기록에서 이벤트 보기](#)를 참조하세요.

에 대한 이벤트를 AWS 계정포함하여에 이벤트를 지속적으로 기록하려면 추적을 Amazon Transcribe 생성합니다. 추적은가 CloudTrail 이벤트를 로그 파일로 지정된 Amazon S3 bucket. CloudTrail log 파일에 하나 이상의 로그 항목을 포함할 수 있도록 하는 구성입니다. 이벤트는 원본의 단일 요청을 나타냅니다. 여기에는 요청된 작업, 작업 날짜 및 시간, 요청 파라미터 등에 대한 정보가 포함됩니다. CloudTrail log 파일은 퍼블릭 API 호출의 순서가 지정된 스택 추적이 아니므로 특정 순서로 표시되지 않습니다.

기본적으로에서 추적을 생성하면 추적 AWS Management Console이 모든에 적용됩니다 AWS 리전. 추적은 AWS 파티션 AWS 리전 의 모든에서 이벤트를 로깅하고 지정한 Amazon S3 버킷으로 로그 파일을 전송합니다. 또한 CloudTrail 로그에 수집된 이벤트 데이터를 추가로 분석하고 조치를 취 AWS 서비스 하도록 다른를 구성할 수 있습니다. 자세한 내용은 다음을 참조하세요.

- [추적 생성 개요](#)
- [CloudTrail 지원되는 서비스 및 통합](#)
- [에 대한 Amazon SNS 알림 구성 CloudTrail](#)
- [여러 리전에서 CloudTrail 로그 파일 받기](#) 및 [여러 계정에서 CloudTrail 로그 파일 받기](#)

CloudTrail 는 [API 참조](#)에 설명된 모든 Amazon Transcribe 작업을 기록합니다. 예를 들어, [CreateVocabulary](#) [GetTranscriptionJob](#) 및 [StartTranscriptionJob](#) 작업은 CloudTrail 로그 파일에 항목을 생성합니다. CloudTrail이 Amazon Transcribe API 작업을 로깅하면 CloudTrail 로그 항목은 Amazon S3 URI 값과 같은 요청 및 응답 파라미터의 민감한 정보에 빈 문자열을 사용합니다.

모든 이벤트 또는 로그 항목에는 요청을 생성했던 사용자에게 관한 정보가 포함됩니다. 이 정보를 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청이 루트 또는 IAM 사용자 자격 증명으로 이루어지는지 여부
- IAM 역할 또는 페더레이션 사용자에게 대한 임시 보안 자격 증명을 통해 요청한 것인지 여부
- 요청이 다른에 의해 이루어지는지 여부 AWS 서비스

자세한 내용은 [CloudTrail userIdentity 요소](#)를 참조하세요.

여러 AWS 리전 및 여러의 Amazon Transcribe 로그 파일을 단일 Amazon S3 버킷 AWS 계정으로 집계할 수도 있습니다. 자세한 내용은 [여러 리전에서 CloudTrail 로그 파일 수신](#) 및 [여러 계정에서 CloudTrail 로그 파일 수신을 참조하세요](#).

예: Amazon Transcribe 로그 파일 항목

추적은 이벤트를 지정된 Amazon S3 bucket. CloudTrail log 파일로 전송할 수 있도록 하는 구성입니다. 로그 파일에는 하나 이상의 로그 항목이 포함됩니다. 이벤트는 원본의 단일 요청을 나타냅니다. 여기에는 작업의 날짜 및 시간과 같은 요청된 작업에 대한 정보와 요청 parameters. CloudTrail log 파일이 퍼블릭 API 호출의 정렬된 스택 추적이 아니므로 특정 순서로 표시되지 않습니다.

[StartTranscriptionJob](#) 및 [GetTranscriptionJob](#) API 작업을 호출하면 다음 항목이 생성됩니다.

Note

CloudTrail이 Amazon Transcribe API 작업을 로깅하면 CloudTrail 로그 항목은 Amazon S3 URI 값과 같은 요청 및 응답 파라미터의 민감한 정보에 빈 문자열을 사용합니다.

```

{
  "Records": [
    {
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "111122223333",
        "arn": "arn:aws:iam:us-west-2:111122223333:user/my-user-name",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "my-user-name"
      },
      "eventTime": "2022-03-07T15:03:45Z",
      "eventSource": "transcribe.amazonaws.com",
      "eventName": "StartTranscriptionJob",
      "awsRegion": "us-west-2",
      "sourceIPAddress": "127.0.0.1",
      "userAgent": "[]",
      "requestParameters": {
        "mediaFormat": "flac",
        "languageCode": "en-US",
        "transcriptionJobName": "my-first-transcription-job",
        "media": {
          "mediaFileUri": ""
        }
      },
      "responseElements": {
        "transcriptionJob": {
          "transcriptionJobStatus": "IN_PROGRESS",
          "mediaFormat": "flac",
          "creationTime": "2022-03-07T15:03:44.229000-08:00",
          "transcriptionJobName": "my-first-transcription-job",
          "languageCode": "en-US",
          "media": {
            "mediaFileUri": ""
          }
        }
      },
      "requestID": "47B8E8D397DCE7A6",
      "eventID": "cdc4b7ed-e171-4cef-975a-ad829d4123e8",
      "eventType": "AwsApiCall",
      "recipientAccountId": "111122223333"
    },
  ],

```



```

{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "111122223333",
    "arn": "arn:aws:iam:us-west-2:111122223333:user/my-user-name",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "my-user-name"
  },
  "eventTime": "2022-03-07T15:07:11Z",
  "eventSource": "transcribe.amazonaws.com",
  "eventName": "GetTranscriptionJob",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "[]",
  "requestParameters": {
    "transcriptionJobName": "my-first-transcription-job"
  },
  "responseElements": {
    "transcriptionJob": {
      "settings": {

      },
      "transcriptionJobStatus": "COMPLETED",
      "mediaFormat": "flac",
      "creationTime": "2022-03-07T15:03:44.229000-08:00",
      "transcriptionJobName": "my-first-transcription-job",
      "languageCode": "en-US",
      "media": {
        "mediaFileUri": ""
      },
      "transcript": {
        "transcriptFileUri": ""
      }
    }
  },
  "requestID": "BD8798EACDD16751",
  "eventID": "607b9532-1423-41c7-b048-ec2641693c47",
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
]

```

```
}
```

Amazon Transcribe과 함께 Amazon EventBridge 사용

Amazon EventBridge를 사용하면 다른 AWS 서비스에서 이벤트를 트리거하여 Amazon Transcribe 작업의 상태 변경에 대응할 수 있습니다. 트랜스크립션 작업으로 상태가 변경되면 EventBridge에서는 이벤트 스트림으로 이벤트를 자동으로 전송합니다. 이벤트 스트림에서 모니터링하고자 하는 이벤트를 정의하고, 이벤트 발생 시 EventBridge에서 취해야 할 조치를 정의하는 규칙을 생성합니다. 조치를 취할 수 있는 또 다른 서비스(또는 대상)로 이벤트를 라우팅하는 경우를 예로 들 수 있습니다. 예를 들어, 트랜스크립션 작업이 성공적으로 완료되면 AWS Lambda 함수로 이벤트를 라우팅하는 규칙을 구성할 수 있습니다. [EventBridge 규칙](#)을 정의하려면 다음 섹션을 참조하세요.

이메일, [채팅 애플리케이션의 Amazon Q Developer](#) 채팅 알림 또는 [AWS Console Mobile Application](#) 푸시 알림을 비롯한 여러 채널을 통해 이벤트에 대한 알림을 받을 수 있습니다. [콘솔 알림 센터](#)에서도 알림을 볼 수 있습니다. 알림을 설정하려는 경우 [AWS User Notifications](#)를 사용할 수 있습니다. AWS User Notifications에서는 집계를 지원하므로 특정 이벤트 중에 받는 알림 수를 줄일 수 있습니다.

EventBridge 규칙 정의

EventBridge 규칙을 정의하려면 [AWS Management Console](#)을 사용하세요. 규칙을 정의할 때 Amazon Transcribe를 서비스 이름으로 사용하세요. EventBridge 규칙을 만드는 방법에 대한 예는 [Amazon EventBridge 규칙](#)을 참조하세요.

EventBridge를 사용하기 전에 다음 정의에 유의하세요.

- 이벤트-이벤트는 트랜스크립션 작업 중 하나의 상태 변화를 나타냅니다. 작업의 TranscriptionJobStatus가 IN_PROGRESS에서 COMPLETED로 변경되는 경우를 예로 들 수 있습니다.
- 대상-대상은 이벤트를 처리하는 또 다른 AWS 서비스입니다. 예를 들면 AWS Lambda 또는 Amazon Simple Notification Service(Amazon SNS)가 있습니다. 대상은 JSON 형식으로 이벤트를 수신합니다.
- 규칙-규칙은 EventBridge에서 감시하도록 하려는 수신 이벤트를 매칭하여 처리를 위해 하나 또는 여러 개의 대상으로 이를 라우팅합니다. 규칙이 하나의 이벤트를 여러 대상으로 라우팅하는 경우, 모든 대상은 이벤트를 병렬 방식으로 처리합니다. 규칙은 대상으로 전송된 JSON을 사용자 지정할 수 있습니다.

Amazon EventBridge 이벤트는 최선의 작업을 기반으로 발생합니다. EventBridge에서 이벤트를 만들고 관리하는 방법에 대한 자세한 내용은 Amazon EventBridge 사용 설명서의 [Amazon EventBridge 이벤트를 참조하세요](#).

다음은 트랜스크립션 작업의 상태가 COMPLETED 또는 FAILED로 변경될 때 시작되는 Amazon Transcribe의 EventBridge 규칙 예시입니다.

```
{
  "source": [
    "aws.transcribe"
  ],
  "detail-type": [
    "Transcribe Job State Change"
  ],
  "detail": {
    "TranscriptionJobStatus": [
      "COMPLETED",
      "FAILED"
    ]
  }
}
```

규칙에는 다음 필드가 포함됩니다.

- **source**–이벤트의 원본. Amazon Transcribe의 경우 항상 `aws.transcribe`입니다.
- **detail-type**–이벤트 세부 정보에 대한 식별자. Amazon Transcribe의 경우 항상 `Transcribe Job State Change`입니다.
- **detail**–트랜스크립션 작업의 새 상태. 이 예시에서 규칙은 작업 상태가 COMPLETED 또는 FAILED로 변경될 때 이벤트를 시작합니다.

Amazon Transcribe 이벤트

Amazon EventBridge는 여러 Amazon Transcribe 이벤트를 로깅합니다.

- [트랜스크립션 작업 이벤트](#)
- [언어 식별 이벤트](#)
- [통화 분석 이벤트](#)
- [통화 분석 통화 후 이벤트](#)
- [어휘 이벤트](#)

이러한 이벤트는 모두 다음과 같은 공유 필드를 포함합니다.

- **version**: 이벤트 데이터의 버전. 이 값은 항상 0입니다.
- **id**: EventBridge에서 이벤트에 대해 생성한 고유 식별자.
- **detail-type**: 이벤트 세부 정보에 대한 식별자. 예를 들어 Transcribe Job State Change입니다.
- **source**: 이벤트의 원본. Amazon Transcribe의 경우 항상 `aws.transcribe`입니다.
- **account**: API 호출을 생성한 계정의 AWS 계정.
- **time**: 이벤트가 전달된 날짜 및 시간.
- **region**: 요청이 이루어진 AWS 리전.
- **resources**: API 호출에서 사용한 리소스. Amazon Transcribe의 경우 이 필드는 항상 비어 있습니다.
- **detail**: 이벤트에 대한 추가 세부 정보.
 - **FailureReason**: 이 필드는 상태 또는 상태가 FAILED로 변경될 경우 표시되며 FAILED 상태의 이유를 설명합니다.
 - 각 이벤트 유형에는 `detail` 아래에 표시되는 고유한 추가 필드가 있습니다. 이러한 고유 필드는 각 이벤트 예시 후 다음 섹션에서 정의합니다.

트랜스크립션 작업 이벤트

작업의 상태가 `IN_PROGRESS`에서 `COMPLETED` 또는 `FAILED`로 바뀌면 Amazon Transcribe에서 이벤트를 생성합니다. 상태가 바뀐 작업을 식별하고 대상에서 이벤트를 시작하려면 이벤트의 `TranscriptionJobName` 필드를 사용합니다. Amazon Transcribe 이벤트에는 다음 정보가 포함됩니다. 트랜스크립션 작업 상태가 `detail`인 경우 `FailureReason` 필드가 `FAILED` 아래에 추가됩니다.

단, 이 이벤트는 [StartTranscriptionJob](#) API 작업에만 적용됩니다.

```
{
  "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**: 트랜스크립션 작업에 대해 선택한 고유한 이름.
- **TranscriptionJobStatus** : 트랜스크립션 작업의 새 상태. 이는 COMPLETED 또는 FAILED일 수 있습니다.

언어 식별 이벤트

[자동 언어 식별](#)을 활성화하면 언어 식별 상태가 COMPLETED 또는 FAILED일 때 Amazon Transcribe에서 이벤트를 생성합니다. 상태가 바뀐 작업을 식별하고 대상에서 이벤트를 시작하려면 이벤트의 JobName 필드를 사용합니다. Amazon Transcribe 이벤트에는 다음 정보가 포함됩니다. 언어 식별 상태가 FAILED인 경우 FailureReason 필드가 detail 아래에 추가됩니다.

단, 이 이벤트는 [LanguageIdSettings](#) 파라미터가 포함된 경우 [StartTranscriptionJob](#) API 작업에만 적용됩니다.

```

{
  "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**: 트랜스크립션 작업에서 이 값은 TranscriptionJob이어야 합니다.
- **JobName**: 트랜스크립션 작업의 고유 이름.
- **LanguageIdentificationStatus**: 트랜스크립션 작업의 언어 식별 상태. 이는 COMPLETED 또는 FAILED일 수 있습니다.

통화 분석 이벤트

통화 분석 작업의 상태가 IN_PROGRESS에서 COMPLETED 또는 FAILED로 바뀌면 Amazon Transcribe에서 이벤트를 생성합니다. 상태가 바뀐 통화 분석 작업을 식별하고 대상에서 이벤트를 시작하려면 이벤트의 JobName 필드를 사용합니다. Amazon Transcribe 이벤트에는 다음 정보가 포함됩니다. 통화 분석 작업의 상태가 FAILED인 경우 FailureReason 필드가 detail 아래에 추가됩니다.

단, 이 이벤트는 [StartCallAnalyticsJob](#) API 작업에만 적용됩니다.

```
{
  "version": "0",
  "id": "event ID",
  "detail-type": "Call Analytics Job State Change",
  "source": "aws.transcribe",
  "account": "111122223333",
  "time": "timestamp",
  "region": "us-west-2",
  "resources": [],
  "detail": {
    "JobName": "my-first-analytics-job",
    "JobStatus": "COMPLETED" (or "FAILED"),
    "FailureReason": "failure reason", // only present when JobStatus is FAILED
    "AnalyticsJobDetails": { // only when you enable optional features such as
      Generative Call Summarization
      "Skipped": []
    }
  }
}
```

- JobName: 통화 분석 트랜스크립션 작업의 고유한 이름.
- JobStatus: 통화 분석 트랜스크립션 작업의 상태. 이는 COMPLETED 또는 FAILED일 수 있습니다.
- FailureReason: 이 필드는 JobStatus가 FAILED인 경우에만 존재하며 실패 이유를 설명합니다.
- AnalyticsJobDetails: 건너뛴 분석 기능에 대한 정보를 포함하여 통화 분석 트랜스크립션 작업의 세부 정보입니다.

통화 분석 통화 후 이벤트

통화 후 분석 트랜스크립션의 상태가 IN_PROGRESS에서 COMPLETED 또는 FAILED로 바뀌면 Amazon Transcribe에서 이벤트를 생성합니다. 상태가 바뀐 통화 분석 통화 후 작업을 식별하고 대상에서 이벤트를 시작하려면 이벤트의 StreamingSessionId 필드를 사용합니다.

단, 이 이벤트는 [PostCallAnalyticsSettings](#) 파라미터가 포함된 경우 [StartCallAnalyticsStreamTranscription](#) API 작업에만 적용됩니다.

COMPLETED 이벤트에는 다음 정보가 포함됩니다.

```
{
  "version": "0",
  "id": "event ID",
  "detail-type": "Call Analytics Post Call Job State Change",
  "source": "aws.transcribe",
  "account": "111122223333",
  "time": "timestamp",
  "region": "us-west-2",
  "resources": [],
  "detail": {
    "StreamingSessionId": "session-id",
    "PostCallStatus": "COMPLETED",
    "Transcript": {
      "RedactedTranscriptFileUri": "s3://amzn-s3-demo-bucket/my-output-files/my-redacted-file.JSON",
      "TranscriptFileUri": "s3://amzn-s3-demo-bucket/my-output-files/my-file.JSON"
    },
    "Media": {
      "MediaFileUri": "s3://amzn-s3-demo-bucket/my-output-files/my-redacted-file.WAV",
      "RedactedMediaFileUri": "s3://amzn-s3-demo-bucket/my-output-files/my-redacted-file.WAV"
    }
  }
}
```

FAILED 이벤트에는 다음 정보가 포함됩니다.

```
{
  "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: 실시간 통화 분석 트랜스크립션 요청에 할당된 식별 번호.
- PostCallStatus: 통화 후 통화 분석 트랜스크립션의 상태. 이는 COMPLETED 또는 FAILED일 수 있습니다.
- Transcript: 수정된 트랜스크립트와 수정되지 않은 트랜스크립트의 URI.
- Media: 수정된 오디오 파일과 수정되지 않은 오디오 파일의 URI.

AWS HealthScribe 스트림 후 분석 이벤트

IN_PROGRESS에서 COMPLETED, AWS로 변경되는 [ClinicalNoteGenerationResult](#)와 같은 AWS HealthScribe 스트림 후 분석 작업의 상태가 변경되면 HealthScribe는 다음 정보가 포함된 이벤트를 생성합니다.

```

{
  "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**: 이벤트를 생성한 스트림 후 분석 작업의 유형입니다. 결과 객체의 내용은 UpdateType에 따라 달라집니다.
- **SessionId**: AWS HealthScribe 스트림의 식별 번호입니다. 이 ID를 사용하여 시작 스트리밍 세션을 식별한 다음 이벤트를 생성한 스트림 후 분석을 찾습니다.
- **Status**: 스트림 후 분석 작업의 상태입니다. IN_PROGRESS, COMPLETED 또는 FAILED 유형을 지정할 수 있습니다.
- **ClinicalNoteOutputLocation**:에 대한 출력 Amazon S3 버킷의 URI입니다
다ClinicalNoteGenerationResult.
- **TranscriptOutputLocation**: 트랜스크립트의 URI입니다.

어휘 이벤트

[사용자 지정 어휘](#)의 상태가 PENDING에서 READY 또는 FAILED로 바뀌면 Amazon Transcribe에서 이벤트를 생성합니다. 상태가 바뀐 사용자 지정 어휘를 식별하고 대상에서 이벤트를 시작하려면 이벤트의 VocabularyName 필드를 사용합니다. Amazon Transcribe 이벤트에는 다음 정보가 포함됩니다. 사용자 지정 어휘 상태가 FAILED인 경우 FailureReason 필드가 detail 아래에 추가됩니다.

Note

단, 이 이벤트는 [CreateVocabulary](#) API 작업에만 적용됩니다.

```
{
  "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**: 사용자 지정 어휘의 고유한 이름.

- VocabularyState: 사용자 지정 어휘의 처리 상태. 이는 READY 또는 FAILED일 수 있습니다.

의 규정 준수 검증Amazon Transcribe

AWS 서비스가 특정 규정 준수 프로그램의 범위에 포함되는지 알아보려면 [규정 준수 프로그램 제공 범위 내 AWS 서비스](#)를 참조하고 관심 있는 규정 준수 프로그램을 선택하십시오. 일반적인 정보는 [AWS 규정 준수 프로그램](#)을 참조하세요.

AWS Artifact(을)를 사용하여 타사 감사 보고서를 다운로드할 수 있습니다. 자세한 내용은 [AWS Artifact에서 보고서 다운로드](#)를 참조하세요.

AWS 서비스 사용 시 규정 준수 책임은 데이터의 민감도, 회사의 규정 준수 목표 및 관련 법률 및 규정에 따라 결정됩니다. AWS 서비스 사용 시 규정 준수 책임에 대한 자세한 내용은 [AWS 보안 설명서](#)를 참조하세요.

Amazon Transcribe의 복원성

AWS 글로벌 인프라는 AWS 리전 및 가용 영역을 중심으로 구축됩니다. AWS 리전에서는 물리적으로 분리되고 격리된 다수의 가용 영역을 제공하며 이러한 가용 영역은 짧은 대기 시간, 높은 처리량 및 높은 중복성을 갖춘 네트워크에 연결되어 있습니다. 가용 영역을 사용하면 중단 없이 영역 간에 자동으로 장애 극복 조치가 이루어지는 애플리케이션 및 데이터베이스를 설계하고 운영할 수 있습니다. 가용 영역은 기존의 단일 또는 다중 데이터 센터 인프라보다 가용성, 내결함성, 확장성이 뛰어납니다.

AWS 리전 및 가용 영역에 대한 자세한 내용은 [AWS 글로벌 인프라](#) 섹션을 참조하세요.

Amazon Transcribe에서 인프라 보안

관리형 서비스인 Amazon Transcribe은 AWS 글로벌 네트워크 보안으로 보호됩니다. AWS 보안 서비스와 AWS의 인프라 보호 방법에 대한 자세한 내용은 [AWS 클라우드 보안](#)을 참조하세요. 인프라 보안에 대한 모범 사례를 사용하여 AWS 환경을 설계하려면 보안 원칙 AWS Well-Architected Framework의 [인프라 보호](#)를 참조하세요.

AWS에서 게시한 API 직접 호출을 사용하여 네트워크를 통해 Amazon Transcribe에 액세스합니다. 고객은 다음을 지원해야 합니다.

- Transport Layer Security(TLS) TLS 1.2는 필수이며 TLS 1.3을 권장합니다.

- DHE(Ephemeral Diffie-Hellman) 또는 ECDHE(Elliptic Curve Ephemeral Diffie-Hellman)와 같은 완전 전송 보안(PFS)이 포함된 암호 제품군 Java 7 이상의 최신 시스템은 대부분 이러한 모드를 지원합니다.

의 취약성 분석 및 관리 Amazon Transcribe

구성 및 IT 제어는 AWS 와 고객 간의 공동 책임입니다. 자세한 내용은 AWS [공동 책임 모델](#)을 참조하세요.

Amazon Transcribe 및 인터페이스 VPC 엔드포인트(AWS PrivateLink)

인터페이스 VPC 엔드포인트를 생성 Amazon Transcribe 하여 VPC와 간에 프라이빗 연결을 설정할 수 있습니다. 인터페이스 엔드포인트는 인터넷 게이트웨이 [AWS PrivateLink](#), NAT 디바이스, VPN 연결 또는 Direct Connect 연결 없이 비공개로 Amazon Transcribe APIs에 액세스하는 데 사용할 수 있는 기술인 로 구동됩니다. VPC의 인스턴스는 Amazon Transcribe APIs. VPC와 간의 트래픽 Amazon Transcribe 은 Amazon 네트워크를 벗어나지 않습니다.

각 인터페이스 엔드포인트는 서브넷에서 하나 이상의 [Elastic Network Interfaces](#)로 표현됩니다.

자세한 내용은 Amazon VPC 사용 설명서의 [인터페이스 VPC 엔드포인트\(AWS PrivateLink\)](#)를 참조하세요.

Amazon Transcribe VPC 엔드포인트에 대한 고려 사항

에 대한 인터페이스 VPC 엔드포인트를 설정하기 전에 Amazon VPC 사용 설명서의 [인터페이스 엔드포인트 속성 및 제한 사항](#)을 검토해야 Amazon Transcribe합니다.

Amazon Transcribe 는 VPC에서 모든 API 작업을 호출할 수 있도록 지원합니다.

에 대한 인터페이스 VPC 엔드포인트 생성 Amazon Transcribe

AWS Management Console 또는를 사용하여 Amazon Transcribe 서비스에 대한 VPC 엔드포인트를 Amazon VPC 생성할 수 있습니다 AWS CLI. 자세한 내용은 Amazon VPC 사용 설명서의 [인터페이스 엔드포인트 생성](#)을 참조하세요.

에서 배치 트랜스크립션의 경우 다음 서비스 이름을 사용하여 VPC 엔드포인트를 Amazon Transcribe 생성합니다.

- com.amazonaws.us-west-2.transcribe

에서 스트리밍 트랜스크립션의 경우 다음 서비스 이름을 사용하여 VPC 엔드포인트를 Amazon Transcribe 생성합니다.

- `com.amazonaws.us-west-2.transcribestreaming`

엔드포인트에 대해 프라이빗 DNS를 활성화하는 경우 AWS 리전과 같은에 대한 기본 DNS 이름을 Amazon Transcribe 사용하여 API 요청을 할 수 있습니다 `transcribestreaming.us-east-2.amazonaws.com`.

자세한 내용은 Amazon VPC 사용 설명서의 [인터페이스 엔드포인트를 통해 서비스 액세스](#)를 참조하세요.

에 대한 VPC 엔드포인트 정책 생성 Amazon Transcribe

스트리밍 서비스 또는 배치 트랜스크립션 서비스에 대한 액세스를 제어하는 엔드포인트 정책을 VPC 엔드포인트에 연결할 수 있습니다 Amazon Transcribe. 이 정책은 다음 정보를 지정합니다.

- 작업을 수행할 수 있는 위탁자.
- 수행할 수 있는 작업.
- 작업을 수행할 수 있는 리소스

자세한 내용은 Amazon VPC 사용 설명서의 [VPC 엔드포인트를 통해 서비스에 대한 액세스 제어](#)를 참조하세요.

예: Amazon Transcribe 배치 트랜스크립션 작업에 대한 VPC 엔드포인트 정책

다음은 Amazon Transcribe의 배치 트랜스크립션에 대한 엔드포인트 정책 예시입니다. 이 정책은 엔드포인트에 연결될 때 모든 리소스의 모든 보안 주체에 대한 액세스 권한을 나열된 Amazon Transcribe 작업에 부여합니다.

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Allow",
      "Action": [
        "transcribe:StartTranscriptionJob",
        "transcribe:ListTranscriptionJobs"
      ],
      "Resource": "*"
    }
  ]
}
```

```
    }
  ]
}
```

예: Amazon Transcribe 스트리밍 트랜스크립션 작업에 대한 VPC 엔드포인트 정책

다음은 Amazon Transcribe의 스트리밍 트랜스크립션에 대한 엔드포인트 정책 예시입니다. 이 정책은 엔드포인트에 연결될 때 모든 리소스의 모든 보안 주체에 대한 액세스 권한을 나열된 Amazon Transcribe 작업에 부여합니다.

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Allow",
      "Action": [
        "transcribe:StartStreamTranscription",
        "transcribe:StartStreamTranscriptionWebsocket"
      ],
      "Resource": "*"
    }
  ]
}
```

공유 서브넷

공유하는 서브넷의 VPC 엔드포인트는 생성, 설명, 수정 또는 삭제할 수 없습니다. 그러나 공유하는 서브넷의 VPC 엔드포인트를 사용할 수는 있습니다. VPC 공유에 대한 자세한 내용은 Amazon Virtual Private Cloud 가이드의 [다른 계정과 VPC 공유를 참조하세요](#).

의 보안 모범 사례Amazon Transcribe

다음 모범 사례는 일반적인 지침이며 완벽한 보안 솔루션을 나타내지는 않습니다. 이러한 모범 사례는 사용자의 환경에 적절하지 않거나 충분하지 않을 수 있으므로 규정이 아닌 참고용으로만 사용하세요.

- 데이터 암호화 사용(예: AWS KMS 암호화 컨텍스트)

AWS KMS 암호화 컨텍스트는 비밀이 아닌 일반 텍스트 키:값 페어의 맵입니다. 이 맵은 데이터에 대한 추가 보안 계층을 제공하는 암호화 컨텍스트 페어로 알려진 추가 인증 데이터를 나타냅니다.

자세한 정보는 [AWS KMS 암호화 컨텍스트](#) 단원을 참조하세요.

- **가능하면 항상 임시 보안 인증 사용**

가능하면 액세스 키와 같은 장기 보안 인증 대신 임시 보안 인증을 사용하는 것이 좋습니다. 프로그래밍 방식 액세스 권한과 장기 보안 인증이 있는 IAM 사용자가 필요한 시나리오의 경우 액세스 키를 교체하는 것이 좋습니다. 장기 보안 인증을 정기적으로 교체하면 프로세스에 익숙해지는 데 도움이 됩니다. 이는 직원이 퇴사할 때와 같이 보안 인증을 교체해야 하는 상황에 유용합니다. IAM 액세스에서 마지막으로 사용한 정보를 사용하여 액세스 키를 안전하게 교체 및 제거하는 것이 좋습니다.

자세한 내용은 [의 액세스 키 교체](#) 및 [IAM의 보안 모범 사례](#)를 참조하세요.

- **Amazon Transcribe 액세스를 요구하는 애플리케이션 및 AWS 서비스에 IAM 역할 사용**

IAM 역할을 사용하여 Amazon Transcribe에 액세스해야 하는 애플리케이션 또는 서비스의 임시 자격 증명을 관리하세요. 역할을 사용할 때 Amazon EC2 인스턴스나 AWS 서비스에 암호나 액세스 키 같은 장기 자격 증명을 배포하지 않아도 됩니다. 애플리케이션에서 AWS 리소스에 요청할 때 사용할 수 있는 임시 권한을 IAM 역할이 제공할 수 있습니다.

자세한 정보는 [IAM 역할](#) 및 [역할에 대한 일반 시나리오: 사용자, 애플리케이션 및 서비스](#)를 참조하세요.

- **태그 기반 액세스 제어**

태그를 사용하여 AWS 계정 내의 액세스를 제어할 수 있습니다. Amazon Transcribe에서 태그는 트랜스크립션 작업, 사용자 지정 어휘, 사용자 지정 어휘 필터 및 사용자 지정 언어 모델에 추가할 수 있습니다.

자세한 정보는 [태그 기반 액세스 제어](#) 단원을 참조하세요.

- **AWS 모니터링 도구 사용**

모니터링은 Amazon Transcribe와 사용자 AWS 솔루션의 안정성, 보안, 가용성 및 성능을 유지하는 중요한 역할을 합니다. CloudTrail를 사용하여 Amazon Transcribe를 모니터링할 수 있습니다.

자세한 정보는 [Amazon Transcribe 를 사용한 모니터링 AWS CloudTrail](#) 단원을 참조하세요.

- **AWS Config 사용 설정**

AWS Config는 AWS 리소스의 구성을 평가, 감사 및 심사할 수 있습니다. AWS Config를 사용하면 구성 변경 사항 및 AWS 리소스 간의 관계를 검토할 수 있습니다. 또한 자세한 리소스 구성 기록을 조사하고 내부 지침에 지정되어 있는 구성에 대한 전반적인 규정 준수 여부를 확인할 수 있습니다. 이를 사용하면 규정 준수 감사, 보안 분석, 변경 관리 및 운영 문제 해결 작업을 간소화할 수 있습니다.

자세한 내용은 [AWS Config란 무엇입니까?](#)를 참조하세요.

Amazon Transcribe Medical

Amazon Transcribe Medical은 의사가 구술한 메모, 약물 안전 모니터링, 원격 진료 예약 또는 의사와 환자의 대화 등 의료 관련 음성을 트랜스크립션하려는 의료 전문가를 위해 설계된 자동 음성 인식(ASR) 서비스입니다. Amazon Transcribe Medical은 실시간 스트리밍(마이크 사용) 또는 업로드된 파일의 트랜스크립션(배치)을 통해 제공됩니다.

Important

Amazon Transcribe Medical은 전문적인 의학적 조언, 진단 또는 치료를 대체할 수 없습니다. 사용 사례에 적합한 신뢰 임계값을 식별하고, 높은 정확도가 필요한 상황에서는 높은 신뢰 임계값을 사용합니다. 특정 사용 사례의 경우 적절한 교육을 받은 인간 검토자가 결과를 검토하고 확인해야 합니다. Amazon Transcribe Medical 트랜스크립션은 숙련된 의료 전문가로부터 정확성과 적절한 의학적 판단을 검토받은 후 환자 치료 시나리오에서만 사용해야 합니다.

Amazon Transcribe Medical은 공동 책임 모델에 따라 운영됩니다. 이에 따라 AWS는 Amazon Transcribe Medical을 실행하는 인프라를 보호할 책임이 있으며 사용자는 데이터를 관리할 책임이 있습니다. 자세한 내용은 [공동 책임 모델](#)을 참조하세요.

Amazon Transcribe Medical은 미국 영어(en-US)로 제공됩니다.

최상의 결과를 얻으려면 FLAC 또는 WAV와 같은 PCM 16비트 인코딩의 무손실 오디오 형식을 사용하세요. Amazon Transcribe Medical은 16,000Hz 이상의 샘플링 레이트를 지원합니다.

트랜스크립트 분석에는 [Amazon Comprehend 의료](#) 등의 다른 AWS 서비스를 사용할 수 있습니다.

지원되는 전문 분야

전문 분야	하위 전문 분야	오디오 입력
심장학	없음	스트리밍만 해당
신경학	없음	스트리밍만 해당
종양학	없음	스트리밍만 해당
1차 의료	가정의학	배치, 스트리밍
1차 의료	내과학	배치, 스트리밍

전문 분야	하위 전문 분야	오디오 입력
1차 의료	산부인과학	배치, 스트리밍
1차 의료	소아과학	배치, 스트리밍
방사선학	없음	스트리밍만 해당
비뇨기과	없음	스트리밍만 해당

리전 가용성 및 할당량

Call Analytics는 다음 AWS 리전에서 지원됩니다.

리전	트랜스크립션 유형
af-south-1(케이프타운)	일괄
ap-east-1(홍콩)	일괄
ap-northeast-1(도쿄)	배치, 스트리밍
ap-northeast-2(서울)	배치, 스트리밍
ap-south-1(뭄바이)	일괄
ap-southeast-1(싱가포르)	일괄
ap-southeast-2(시드니)	배치, 스트리밍
ca-central-1(캐나다, 중부)	배치, 스트리밍
eu-central-1(프랑크푸르트)	배치, 스트리밍
eu-north-1(스톡홀름)	일괄
eu-west-1(아일랜드)	배치, 스트리밍
eu-west-2(런던)	배치, 스트리밍

리전	트랜스크립션 유형
eu-west-3(파리)	일괄
me-south-1(바레인)	일괄
sa-east-1(상파울루)	배치, 스트리밍
us-east-1(버지니아 북부)	배치, 스트리밍
us-east-2(오하이오)	배치, 스트리밍
us-gov-east-1(GovCloud, 미국 동부)	배치, 스트리밍
us-gov-west-1,(GovCloud, 미국 서부)	배치, 스트리밍
us-west-1(샌프란시스코)	일괄
us-west-2(오리건)	배치, 스트리밍

[Amazon Transcribe](#), Amazon Transcribe Medical 및 [Call Analytics](#)에 대한 리전 지원은 다릅니다. .

지원되는 각 리전에 대한 엔드포인트를 얻으려면 AWS 일반 참조의 [서비스 엔드포인트](#)를 참조하세요.

트랜스크립션과 관련된 할당량 목록은 AWS 일반 참조의 [서비스 할당량](#)을 참조하세요. 일부 할당량은 요청 시 변경될 수 있습니다. 조정 가능 열에 '예'가 포함된 경우 증가를 요청할 수 있습니다. 이렇게 하려면 제공된 링크를 선택합니다.

의료 전문 분야 및 용어

의료 트랜스크립션 작업을 생성할 때 소스 파일의 언어, 의료 전문 분야 및 오디오 유형을 지정합니다. 언어 및 PRIMARYCARE 의료 전문 분야로 미국 영어(en-US)를 입력합니다. 1차 진료를 값으로 입력하면 다음 의학 전문 분야의 소스 오디오에서 트랜스크립션을 생성할 수 있습니다.

- 가정의학
- 내과학
- 산부인과학
- 소아과학

오디오 유형에 맞게 구술 및 대화 중에서 선택할 수 있습니다. 의사가 환자 방문 또는 절차에 대한 보고서를 제공하는 오디오 파일의 경우 구술을 선택합니다. 의사와 환자 간의 대화 또는 의사 간의 대화가 포함된 오디오 파일의 경우 대화를 선택합니다.

트랜스크립션 작업의 출력을 저장하려면 이미 생성한 Amazon S3 버킷을 선택합니다. Amazon S3 버킷에 대한 자세한 내용은 [시작하기를 Amazon Simple Storage Service](#) 참조하세요.

다음은 샘플 JSON에 입력할 요청 파라미터의 최소 수입니다.

```
{
  "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 Medical을 사용하면 대체 트랜스크립션을 생성할 수 있습니다. 자세한 내용은 [대체 트랜스크립션 생성](#) 단원을 참조하십시오.

화자 파티셔닝을 활성화하거나 오디오에서 채널을 식별할 수도 있습니다. 자세한 내용은 [화자 파티셔닝 활성화](#) 및 [다중 채널 오디오 트랜스크립션](#) 섹션을 참조하세요.

의학 용어 및 측정치 트랜스크립션

Amazon Transcribe Medical은 의학 용어 및 측정값을 트랜스크립션할 수 있습니다. Amazon Transcribe 의학은 말한 용어의 약어를 출력합니다. 예를 들어 “blood pressure”는 BP로 트랜스크립션됩니다. Amazon Transcribe Medical이 의료 용어 및 측정에 사용하는 규칙 목록은 이 페이지의 표에서 확인할 수 있습니다. Spoken Term(발언한 용어) 열은 소스 오디오에서 말한 용어를 나타냅니다. Output(출력) 열은 트랜스크립션 결과에 표시되는 약어를 나타냅니다.

소스 오디오에서 말한 용어가 여기의 트랜스크립션 출력과 어떻게 일치하는지 확인할 수 있습니다.

소스 오디오에서 말한 용어	출력에 사용되는 약어	출력 예시
Centigrade	C	The patient's temperature is 37.4 C.(환자의 체온은 37.4C입니다.)

소스 오디오에서 말한 용어	출력에 사용되는 약어	출력 예시
Celsius	C	The patient's temperature is 37.4 C.(환자의 체온은 37.4C입니다.)
Fahrenheit	F	The patient's temperature is 101 F.(환자의 체온은 101F입니다.)
grams	g	A mass of 100 g was extracted from the patient.(환자에게서 100g의 질량을 추출했습니다.)
meters	m	The patient is 1.8 m tall.(환자의 키는 1.8m입니다.)
feet	ft	The patient is 6 ft tall.(환자는 키는 6ft입니다.)
kilos	kg	The patient weighs 80 kg.(환자의 체중은 80kg입니다.)
kilograms	kg	The patient weighs 80 kg.(환자의 체중은 80kg입니다.)
c c	cc	Patient received 100 cc of saline solution.(환자는 100cc의 식염수를 수령했습니다.)
cubic centimeter	cc	Patient received 100 cc of saline solution.(환자는 100cc의 식염수를 수령했습니다.)
milliliter	mL	Patient excreted 100 mL urine.(환자는 100mL의 소변을 배설했습니다.)

소스 오디오에서 말한 용어	출력에 사용되는 약어	출력 예시
blood pressure	BP	Patient BP was elevated.(환자의 BP가 상승했습니다.)
b p	BP	Patient BP was elevated.(환자의 BP가 상승했습니다.)
X over Y	X/Y	Patient BP was 120/80.(환자의 BP는 120/80이었습니다.)
beats per min	BPM	Patient had atrial fibrillation with heart rate of 160 BPM.(환자는 160BPM의 심박수로 심방 세동이 있었습니다.)
beats per minute	BPM	Patient had atrial fibrillation with heart rate of 160 BPM.(환자는 160BPM의 심박수로 심방 세동이 있었습니다.)
O 2	O2	Patient O2 saturation was 98%.(환자의 O2 포화도는 98%였습니다.)
CO2	CO2	Patient required respiratory support for elevated CO2.(환자는 상승된 CO2로 인해 호흡기 지원이 필요했습니다.)
post operation	POSTOP	Patient came for POSTOP evaluation.(환자가 POSTOP 평가를 위해 방문했습니다.)
post op	POSTOP	Patient came for POSTOP evaluation.(환자가 POSTOP 평가를 위해 방문했습니다.)

소스 오디오에서 말한 용어	출력에 사용되는 약어	출력 예시
cat scan	CT Scan	Patient indication of cerebral hemorrhage required use of CT Scan.(뇌출혈 환자였기 때문에 CT 스캔 사용이 필요했습니다.)
Pulse 80	P 80	Patient vitals were P 80, R 17,...(환자의 바이탈은 P 80, R 17 등이었습니다.)
Respiration 17	R 17	Patient vitals were P 80, R 17,...(환자의 바이탈은 P 80, R 17 등이었습니다.)
in and out	I/O	Patient was I/O sinus rhythm. (환자는 I/O 부비동 리듬이었습니다.)
L five	L5	Lumbar puncture was performed between L4 and L5. (L4와 L5 사이에서 요추 천자가 수행되었습니다.)

번호 트랜스크립션

Amazon Transcribe Medical은 숫자를 단어 대신 숫자로 트랜스크립션합니다. 예를 들어 숫자 "천 이백 사십 이"를 말하면 1242로 트랜스크립션됩니다.

숫자는 다음 규칙에 따라 트랜스크립션됩니다.

규칙	설명
10보다 큰 기수를 숫자로 변환합니다.	"Fifty five" > 55 "a hundred" > 100 "One thousand and thirty one" > 1031

규칙	설명
	"One hundred twenty-three million four hundred fifty six thousand seven hundred eight nine" > 123,456,789
"million" 또는 "billion" 뒤에 숫자가 오지 않는 경우 뒤에 "million" 또는 "billion"이 있는 기수를 숫자로 변환합니다.	"one hundred million" > 100,000,000 "one billion" > 1,000,000,000 "two point three million" > 2,300,000
10보다 큰 서수를 숫자로 변환합니다.	"Forty third" > 43rd "twenty sixth avenue" > 26th avenue
분수를 숫자 형식으로 변환합니다.	"a quarter" > 1/4 "three sixteenths" > 3/16 "a half" > 1/2 "a hundredth" > 1/100
행에 하나 이상의 자릿수가 10 미만으로 숫자로 변환합니다.	"three four five" > 345 "My phone number is four two five five five five one two one two" > 4255551212
소수점은 "점"으로 표시됩니다.	"three hundred and three dot five" > 303.5 "three point twenty three" > 3.23 "zero point four" > 0.4 "point three" > 0.3
숫자 뒤의 "percent" 단어를 백분율 기호(%)로 변환합니다.	"twenty three percent" > 23% "twenty three point four five percent" > 23.45%
숫자 뒤의 "dollar," "US dollar," "Australian dollar," "AUD" 또는 "USD" 단어를 숫자 앞에 달러 기호로 변환합니다.	"one dollar and fifteen cents" > \$1.15 "twenty three USD" > \$23 "twenty three Australian dollars" > \$23

규칙	설명
“pounds” 또는 “milligrams” 단어를 각각 “lbs” 또는 “mg”로 변환합니다.	• "twenty three pounds" > 23 lbs • "forty-five milligrams" > 45 mg
숫자 뒤의 "rupees," "Indian rupees" 또는 "INR" 단어를 숫자 앞의 루피 기호(#)로 변환합니다.	• "twenty three rupees" > #23 • "fifty rupees thirty paise" > #50.30
시간을 숫자로 변환합니다.	• "seven a m eastern standard time" > 7 a.m. eastern standard time • "twelve thirty p m" > 12:30 p.m.
연도를 두 자리에서 네 자리로 표현합니다. 20세기, 21세기, 22세기에 한해 유효합니다.	• "nineteen sixty two" > 1962 • "the year is twenty twelve" > the year is 2012 • "twenty nineteen" > 2019 • "twenty one thirty" > 2130
날짜를 숫자로 변환합니다.	• "May fifth twenty twelve" > May 5th 2012 • "May five twenty twelve" > May 5 2012 • "five May twenty twelve" > 5 May 2012
숫자의 범위를 "to" 단어로 구분합니다.	• "twenty three to thirty seven" > 23 to 37

의료 대화 트랜스크립션

Amazon Transcribe Medical을 사용하여 배치 트랜스크립션 작업 또는 실시간 스트림을 사용하여 임상 의와 환자 간의 의료 대화를 트랜스크립션할 수 있습니다. 배치 트랜스크립션 작업을 사용하면 오디오 파일을 트랜스크립션할 수 있습니다. Amazon Transcribe Medical이 가능한 가장 높은 정확도로 트랜스크립션 결과를 생성하도록 하려면 트랜스크립션 작업 또는 스트림에 임상 의의 의료 전문 분야를 지정해야 합니다.

다음 의료 전문 분야에 있는 임상 의와 환자의 방문을 트랜스크립션할 수 있습니다.

- 심장학 – 스트리밍 트랜스크립션으로만 사용 가능
- 신경학 – 스트리밍 트랜스크립션으로만 사용 가능
- 종양학 – 스트리밍 트랜스크립션으로만 사용 가능

- 1차 의료 – 다음과 같은 유형의 의료 행위가 포함됩니다.
 - 가정의학
 - 내과학
 - 산부인과학
 - 소아과학
- 비뇨기과 – 스트리밍 트랜스크립션으로만 사용 가능

사용자 지정 의학 어휘를 사용하여 트랜스크립션 정확도를 높일 수 있습니다. 사용자 지정 의학 어휘의 작동 방식에 대한 자세한 내용은 [사용자 지정 의학 어휘를 통한 트랜스크립션 정확도 향상](#) 섹션을 참조하세요.

기본적으로 Amazon Transcribe Medical은 신뢰도가 가장 높은 트랜스크립션을 반환합니다. 대체 트랜스크립션을 반환하도록 구성하려면 [대체 트랜스크립션 생성](#) 섹션을 참조하세요.

트랜스크립션 출력에 숫자 및 의료 측정치가 어떻게 표시되는지에 대한 자세한 내용은 [번호 트랜스크립션](#) 및 [의학 용어 및 측정치 트랜스크립션](#) 섹션을 참조하세요.

주제

- [의료 대화의 오디오 파일 트랜스크립션](#)
- [의료 대화를 실시간 스트림으로 트랜스크립션](#)
- [화자 파티셔닝 활성화](#)
- [다중 채널 오디오 트랜스크립션](#)

의료 대화의 오디오 파일 트랜스크립션

배치 트랜스크립션 작업을 사용하면 의료 대화의 오디오 파일을 트랜스크립션할 수 있습니다. 이를 사용하면 임상가와 환자의 대화를 트랜스크립션할 수 있습니다. [StartMedicalTranscriptionJob](#) API 또는 AWS Management Console에서 배치 트랜스크립션 작업을 시작할 수 있습니다.

[StartMedicalTranscriptionJob](#) API로 의료 트랜스크립션 작업을 시작하는 경우 Specialty 파라미터의 값으로 PRIMARYCARE를 지정합니다.

AWS Management Console

임상의와 환자의 대화를 트랜스크립션하려면(AWS Management Console)

AWS Management Console 를 사용하여 임상의-환자 대화를 트랜스크립션하려면 트랜스크립션 작업을 생성하고 오디오 입력 유형에 대한 대화를 선택합니다.

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창의 Amazon Transcribe Medical에서 트랜스크립션 작업을 선택합니다.
3. 작업 생성을 선택합니다.
4. DB 세부 정보 지정 페이지의 작업 설정에서 다음을 지정합니다.
 - a. 이름 - 트랜스크립션 작업의 이름
 - b. 오디오 입력 유형 - 대화
5. 나머지 필드에는 오디오 파일의 Amazon S3 위치와 트랜스크립션 작업의 출력을 저장할 위치를 지정합니다.
6. 다음을 선택합니다.
7. 생성(Create)을 선택합니다.

API

배치 트랜스크립션 작업을 사용하여 의료 대화를 트랜스크립션하려면(API)

- [StartMedicalTranscriptionJob](#) API의 경우 다음을 지정하세요.
 - a. MedicalTranscriptionJobName에서 AWS 계정의 고유한 이름을 지정합니다.
 - b. LanguageCode에는 오디오 파일에서 사용하는 언어와 어휘 필터의 언어에 해당하는 언어 코드를 지정합니다.
 - c. Media 객체의 MediaFileUri 파라미터에서 트랜스크립션할 오디오 파일의 이름을 지정합니다.
 - d. Specialty에는 오디오 파일에서 말하는 임상의의 전문 분야를 PRIMARYCARE로 지정합니다.
 - e. Type에서 CONVERSATION를 지정합니다.
 - f. OutputBucketName에서 트랜스크립션 결과를 저장할 Amazon S3 버킷을 지정합니다.

다음은 AWS SDK for Python (Boto3) 를 사용하여 PRIMARYCARE 전문 임상가와 환자의 의료 대화를 트랜스크립션하는 요청의 예입니다.

```
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)
```

다음 예시 코드는 임상가와 환자 간 대화의 트랜스크립션 결과를 보여줍니다.

```
{
  "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

배치 트랜스크립션 작업을 사용하여 의료 대화를 트랜스크립션하려면(AWS CLI)

- 다음 코드를 실행합니다.

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

```
--region us-west-2 \
--cli-input-json file://example-start-command.json
```

다음 코드는 `example-start-command.json`의 내용을 보여줍니다.

```
{
  "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"
}
```

의료 대화를 실시간 스트림으로 트랜스크립션

HTTP/2 또는 [WebSocket](#) 프로토콜을 사용하여 의료 대화의 오디오 스트림을 트랜스크립션할 수 있습니다. WebSocket 프로토콜을 사용하여 스트림을 시작하는 방법에 대한 자세한 내용은 [WebSocket 스트림 설정](#) 섹션을 참조하세요. HTTP/2 요청을 시작하려면 [StartMedicalStreamTranscription](#) API를 사용하세요.

다음 의료 전문 분야의 스트리밍 오디오를 트랜스크립션할 수 있습니다.

- 심장학
- 신경학
- 종양학
- 1차 의료
- 비뇨기과

각 의료 전문 분야에는 다양한 유형의 시술과 예약이 포함됩니다. 따라서 임상 의는 다양한 유형의 메모를 작성합니다. WebSocket 요청의 `specialty` URI 파라미터 값 또는 [StartMedicalStreamTranscription](#) API의 `Specialty` 파라미터를 지정하는 데 도움이 되는 지침으로 다음 예시를 사용하세요.

- 전기생리학 또는 심장초음파 상담을 원하시면 CARDIOLOGY을 선택합니다.
- 의학 종양학, 수술 종양학 또는 방사선 종양학 상담을 원하시면 ONCOLOGY을 선택합니다.
- 일과성 허혈성 발작이나 뇌혈관 발작에 걸린 뇌졸중 환자에게 상담을 제공하는 의사를 원하시면 NEUROLOGY을 선택합니다.
- 요실금 관련 상담을 원하시면 UROLOGY을 선택합니다.
- 연간 검진 또는 긴급 진료 방문의 경우 PRIMARYCARE를 선택합니다.
- 입원 환자 진찰의 경우 PRIMARYCARE를 선택합니다.
- 불임, 난관 결찰, IUD 삽입 또는 낙태에 관한 상담을 원하시면 PRIMARYCARE를 선택합니다.

AWS Management Console

스트리밍 의학 대화를 트랜스크립션하려면(AWS Management Console)

AWS Management Console 를 사용하여 실시간 스트림에서 임상-환자 대화를 트랜스크립션하려면 의료 대화를 트랜스크립션하고 스트림을 시작하고 마이크에 말하기 시작하는 옵션을 선택합니다.

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창의 Amazon Transcribe Medical에서 실시간 트랜스크립션을 선택합니다.
3. 대화를 선택합니다.
4. 의료 전문 분야에서 임상-환자 대화의 전문 분야를 선택합니다.
5. 스트리밍 시작을 선택합니다.
6. 마이크에 대고 말합니다.

의료 대화를 HTTP/2 스트림으로 트랜스크립션

다음은 HTTP/2 요청 파라미터의 구문입니다.

의료 대화의 HTTP/2 스트림을 트랜스크립션하려면 [StartMedicalStreamTranscription](#) API를 사용하고 다음을 지정하세요.

- LanguageCode - 언어 코드. 유효한 값은 en-US입니다.
- MediaEncoding - 입력 오디오에 사용된 인코딩. 유효한 값은 pcm, ogg-opus, flac입니다.
- Specialty - 의료 전문가의 전문 분야.
- Type - CONVERSATION

실시간 스트림에서 특정 용어의 트랜스크립션 정확도를 높이려면 사용자 지정 어휘를 사용하세요. 사용자 지정 어휘를 활성화하려면 VocabularyName 파라미터 값을, 사용할 사용자 지정 어휘의 이름으로 설정하세요. 자세한 내용은 [사용자 지정 의학 어휘를 통한 트랜스크립션 정확도 향상](#) 단원을 참조하십시오.

다른 화자의 음성에서 레이블을 지정하려면 ShowSpeakerLabel 파라미터를 true로 설정합니다. 자세한 내용은 [화자 파티셔닝 활성화](#) 단원을 참조하십시오.

의료 대화를 트랜스크립션하기 위한 HTTP/2 스트림 설정에 대한 자세한 내용은 [HTTP/2 스트림 설정](#) 섹션을 참조하세요.

의료 대화를 WebSocket 스트림으로 트랜스크립션

WebSocket 요청을 사용하여 의료 대화를 트랜스크립션할 수 있습니다. WebSocket 요청을 할 때 미리 서명된 URI를 생성합니다. 이 URI에는 애플리케이션과 Amazon Transcribe Medical 간에 오디오 스트림을 설정하는 데 필요한 정보가 포함되어 있습니다. WebSocket 요청 생성에 대한 자세한 내용은 [WebSocket 스트림 설정](#) 섹션을 참조하세요.

다음 템플릿을 사용하여 미리 서명된 URI를 생성합니다.

```
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
```

실시간 스트림에서 특정 용어의 트랜스크립션 정확도를 높이려면 사용자 지정 어휘를 사용하세요. 사용자 지정 어휘를 활성화하려면 `vocabulary-name` 값을, 사용할 사용자 지정 어휘의 이름으로 설정하세요. 자세한 내용은 [사용자 지정 의학 어휘를 통한 트랜스크립션 정확도 향상](#) 단원을 참조하십시오.

다른 화자의 음성에서 레이블을 지정하려면 `show-speaker-label` 파라미터를 `true`로 설정합니다. 자세한 내용은 [화자 파티셔닝 활성화](#) 단원을 참조하십시오.

미리 서명된 URI 생성에 대한 자세한 내용은 [WebSocket 스트림 설정](#) 섹션을 참조하세요.

화자 파티셔닝 활성화

Amazon Transcribe Medical에서 화자 파티셔닝을 활성화하려면 화자 분할을 사용합니다. 이렇게 하면 트랜스크립트 출력에서 환자가 말한 내용과 임상이가 말한 내용을 구별할 수 있습니다.

화자 분할을 활성화하면 Amazon Transcribe Medical은 각 화자의 발화에 각 화자의 고유 식별자로 레이블을 지정합니다. 발화는 말의 단위로, 일반적으로 침묵을 기준으로 다른 발화와 구분됩니다. 배치 트랜스크립션의 경우, 임상이의 발화는 `spk_0`의 레이블을, 환자의 발화는 `spk_1`의 레이블을 받을 수 있습니다.

한 화자의 발화가 다른 화자의 발화와 겹치는 경우, Amazon Transcribe Medical 팀은 트랜스크립션에서 환자의 시작 시간을 기준으로 발화를 정리합니다. 입력 오디오에서 중첩되는 발화는 트랜스크립션 출력에서 중첩되지 않습니다.

배치 트랜스크립션 작업을 사용하거나 실시간 스트림으로 오디오 파일을 트랜스크립션할 때 화자 분할을 활성화할 수 있습니다.

주제

- [배치 트랜스크립션에서 화자 파티셔닝 활성화](#)
- [실시간 스트림에서 화자 파티셔닝 활성화](#)

배치 트랜스크립션에서 화자 파티셔닝 활성화

[StartMedicalTranscriptionJob](#) API 또는 AWS Management Console을 사용하여 배치 트랜스크립션 작업을 활성화할 수 있습니다. 이를 통해 의사와 환자의 대화에서 화자별로 텍스트를 분할하고 트랜스크립션 출력에서 누가 무슨 말을 했는지 확인할 수 있습니다.

AWS Management Console

AWS Management Console 를 사용하여 트랜스크립션 작업에서 화자 분할을 활성화하려면 오디오 식별을 활성화한 다음 화자 파티셔닝을 활성화합니다.

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창의 Amazon Transcribe Medical에서 트랜스크립션 작업을 선택합니다.
3. 작업 생성을 선택합니다.
4. 작업 세부 정보 지정 페이지에서 트랜스크립션 작업에 대한 정보를 제공합니다.
5. 다음을 선택합니다.
6. 오디오 식별을 활성화합니다.
7. 오디오 식별 유형에서 화자 파티셔닝을 선택합니다.
8. 최대 화자 수에는 오디오 파일에서 말하고 있다고 생각되는 최대 화자 수를 입력합니다.
9. 생성(Create)을 선택합니다.

API

배치 트랜스크립션 작업을 사용하여 화자 파티셔닝을 활성화하려면(API)

- [StartMedicalTranscriptionJob](#) API의 경우 다음을 지정하세요.
 - a. MedicalTranscriptionJobName에서 AWS 계정의 고유한 이름을 지정합니다.
 - b. LanguageCode에는 오디오 파일에서 사용하는 언어에 해당하는 언어 코드를 지정합니다.
 - c. Media 객체의 MediaFileUri 파라미터에서 트랜스크립션할 오디오 파일의 이름을 지정합니다.
 - d. Specialty에는 오디오 파일에서 말하는 임상치의 전문 분야를 지정합니다.
 - e. Type에서 CONVERSATION를 지정합니다.
 - f. 에서 트랜스크립션 결과를 저장할 Amazon S3 버킷을 OutputBucketName 지정합니다.
 - g. Settings 객체에서 다음을 지정합니다.
 - i. ShowSpeakerLabels - true.
 - ii. MaxSpeakerLabels - 오디오에서 말하는 것으로 생각되는 화자의 수를 나타내는 2에서 10 사이의 정수.

다음 요청은 AWS SDK for Python (Boto3) 를 사용하여 화자 파티셔닝이 활성화된 기본 의료 임상치의 환자 대화의 배치 트랜스크립션 작업을 시작합니다.

```
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)

```

다음 예시 코드는 화자 파티셔닝이 활성화된 상태에서 트랜스크립션 작업의 트랜스크립션 결과를 보여줍니다.

```

{
  "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

1차 의료를 담당하는 임상 의와 환자 간의 대화를 녹음한 오디오 파일을 트랜스크립션하려면(AWS CLI)

- 다음 코드를 실행합니다.

```

aws transcribe start-transcription-job \
--region us-west-2 \
--cli-input-json file://example-start-command.json

```

다음 코드는 *example-start-command.json*의 내용을 보여줍니다.

```

{
  "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
    }
}

```

실시간 스트림에서 화자 파티셔닝 활성화

화자를 분할하고 실시간 스트림에서 음성에 레이블을 지정하려면 AWS Management Console 또는 스트리밍 요청을 사용합니다. 화자 파티셔닝은 스트림에서 2~5명의 화자에게 가장 적합합니다. Amazon Transcribe Medical은 스트림에서 5명 이상의 화자를 분할할 수 있지만 해당 수를 초과하면 파티션의 정확도가 감소합니다.

HTTP/2 요청을 시작하려면 [StartMedicalStreamTranscription](#) API를 사용하세요. WebSocket 요청을 시작하려면 미리 서명된 URI를 사용하세요. URI에는 애플리케이션과 Amazon Transcribe Medical 간에 양방향 통신을 설정하는 데 필요한 정보가 포함되어 있습니다.

마이크에 대고 말하는 오디오의 화자 파티셔닝 활성화(AWS Management Console)

AWS Management Console 를 사용하여 임상-환자 대화의 실시간 스트림 또는 마이크에 실시간으로 표현되는 구술을 시작할 수 있습니다.

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창의 Amazon Transcribe Medical에서 실시간 트랜스크립션을 선택합니다.
3. 오디오 입력 유형에서 트랜스크립션할 의료 음성 유형을 선택합니다.
4. 추가 설정에서 화자 파티셔닝을 선택합니다.
5. 스트리밍 시작을 선택하여 실시간 오디오 트랜스크립션을 시작합니다.
6. 마이크에 대고 말합니다.

HTTP/2 스트림에서 화자 파티셔닝 활성화

의료 대화의 HTTP/2 스트림에서 화자 파티셔닝을 활성화하려면

[StartMedicalStreamTranscription](#) API를 사용하고 다음을 지정하세요.

- LanguageCode에는 오디오 파일에서 사용하는 언어에 해당하는 언어 코드를 지정합니다. 유효한 값은 en-US입니다.

- MediaSampleHertz에서 오디오의 샘플 속도를 지정합니다.
- Specialty에서 제공자의 전문 의료 분야를 지정합니다.
- ShowSpeakerLabel – true

의료 대화를 트랜스크립션하기 위한 HTTP/2 스트림 설정에 대한 자세한 내용은 [HTTP/2 스트림 설정](#) 섹션을 참조하세요.

WebSocket 요청에서 화자 파티셔닝 활성화

API를 사용하여 WebSocket 스트림에서 화자를 분할하려면 다음 형식을 사용하여 미리 서명된 URL을 만들고 WebSocket 요청을 시작하여 show-speaker-label을 true로 설정하세요.

```
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
```

다음 코드는 스트리밍 요청의 잘린 예시 응답을 보여줍니다.

```
{
  "Transcript": {
    "Results": [
      {
        "Alternatives": [
          {
```

```
"Items": [  
  {  
    "Confidence": 0.97,  
    "Content": "From",  
    "EndTime": 18.98,  
    "Speaker": "0",  
    "StartTime": 18.74,  
    "Type": "pronunciation",  
    "VocabularyFilterMatch": false  
  },  
  {  
    "Confidence": 1,  
    "Content": "the",  
    "EndTime": 19.31,  
    "Speaker": "0",  
    "StartTime": 19,  
    "Type": "pronunciation",  
    "VocabularyFilterMatch": false  
  },  
  {  
    "Confidence": 1,  
    "Content": "last",  
    "EndTime": 19.86,  
    "Speaker": "0",  
    "StartTime": 19.32,  
    "Type": "pronunciation",  
    "VocabularyFilterMatch": false  
  },  
  ...  
  {  
    "Confidence": 1,  
    "Content": "chronic",  
    "EndTime": 22.55,  
    "Speaker": "0",  
    "StartTime": 21.97,  
    "Type": "pronunciation",  
    "VocabularyFilterMatch": false  
  },  
  ...  
  {  
    "Confidence": 1,  
    "Content": "fatigue",  
    "EndTime": 24.42,  
    "Speaker": "0",  
    "StartTime": 23.95,
```

```

        "Type": "pronunciation",
        "VocabularyFilterMatch": false
    },
    {
        "EndTime": 25.22,
        "StartTime": 25.22,
        "Type": "speaker-change",
        "VocabularyFilterMatch": false
    },
    {
        "Confidence": 0.99,
        "Content": "True",
        "EndTime": 25.63,
        "Speaker": "1",
        "StartTime": 25.22,
        "Type": "pronunciation",
        "VocabularyFilterMatch": false
    },
    {
        "Content": ".",
        "EndTime": 25.63,
        "StartTime": 25.63,
        "Type": "punctuation",
        "VocabularyFilterMatch": false
    }
],
    "Transcript": "From the last note she still has mild sleep deprivation and chronic fatigue True."
}
],
    "EndTime": 25.63,
    "IsPartial": false,
    "ResultId": "XXXXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
    "StartTime": 18.74
}
]
}
}

```

Amazon Transcribe Medical은 화자 변경 또는 오디오 일시 중지나 같은 자연 음성 세그먼트를 기반으로 수신 오디오 스트림을 중단합니다. 트랜스크립션은 전체 세그먼트가 트랜스크립션될 때까지 애플리케이션에 점진적으로 반환되고, 각 응답에는 텍스트로 변환된 음성이 포함되어 있습니다. 위 코드는 완

전히 트랜스크립션 음성 세그먼트의 잘린 예입니다. 화자 레이블은 완전히 트랜스크립션된 세그먼트에만 표시됩니다.

다음 목록은 스트리밍 트랜스크립션 출력의 객체 및 파라미터 구성을 보여줍니다.

Transcript

각 음성 세그먼트에는 고유한 Transcript 객체가 있습니다.

Results

각 Transcript 객체에는 고유한 Results 객체가 있습니다. 이 객체는 `isPartial` 필드를 포함하고 있습니다. 값이 `false`인 경우 전체 음성 세그먼트에 대한 결과가 반환됩니다.

Alternatives

각 Results 객체에는 Alternatives 객체가 있습니다.

Items

각 Alternatives 객체에는 트랜스크립션 출력의 각 단어 및 문장 부호에 대한 정보가 들어 있는 고유한 Items 객체가 있습니다. 화자 파티셔닝을 활성화하면 각 단어에는 완전히 트랜스크립션된 스피치 세그먼트에 대한 Speaker 레이블이 있습니다. Amazon Transcribe Medical은 이 레이블을 사용하여 스트림의 각 화자에 고유한 정수를 할당합니다. 값이 `speaker-change`인 Type 파라미터는 한 사람이 말을 멈췄고 다른 사람이 곧 말을 시작하려 한다는 것을 나타냅니다.

Transcript

각 Items 객체에는 Transcript 필드의 값으로 트랜스크립션된 음성 세그먼트가 포함되어 있습니다.

WebSocket 요청에 대한 자세한 내용은 [WebSocket 스트림 설정](#) 섹션을 참조하세요.

다중 채널 오디오 트랜스크립션

여러 채널이 있는 오디오 파일 또는 스트림이 있는 경우 채널 식별을 사용하여 각 채널의 음성을 트랜스크립션할 수 있습니다. Amazon Transcribe Medical은 각 채널의 음성을 별도로 트랜스크립션합니다. 각 채널의 개별 트랜스크립션을 결합하여 단일 트랜스크립션 출력으로 만듭니다.

채널 식별을 사용하여 오디오의 개별 채널을 식별하고 각 채널의 음성을 트랜스크립션할 수 있습니다. 발신자와 에이전트가 나오는 시나리오 등의 상황에서 이 기능을 활성화하세요. 이 채널 식별을 사용하

여 약물 안전성 모니터링을 수행하는 고객 센터의 녹음이나 스트림에서 발신자와 에이전트를 구별할 수 있습니다.

배치 프로세싱과 실시간 스트리밍 모두에 채널 식별을 활성화할 수 있습니다. 다음 목록은 각 방법에서 채널 식별을 활성화하는 방법을 설명합니다.

- 배치 트랜스크립션 및 AWS Management Console [StartMedicalTranscriptionJob](#) API
- 스트리밍 트랜스크립션 - WebSocket 스트리밍 및 [StartMedicalStreamTranscription](#) API

다중 채널 오디오 파일 트랜스크립션

오디오 파일을 트랜스크립션하면 Amazon Transcribe Medical은 각 채널의 항목 목록을 반환합니다. 항목은 트랜스크립션된 단어 또는 문장 부호입니다. 각 단어에는 시작 시간과 종료 시간이 있습니다. 한 채널에 있는 사람이 별도의 채널에 있는 사람과 동시에 말하는 경우, 각 채널에 있는 항목의 시작 시간과 종료 시간은 개인이 서로 말하는 동안 서로 겹칩니다.

기본적으로 두 채널로 오디오 파일을 트랜스크립션할 수 있습니다. 채널이 3개 이상인 파일을 트랜스크립션해야 하는 경우 할당량 증가를 요청할 수 있습니다. 할당량 증가 요청에 대한 자세한 내용은 [AWS 서비스 할당량](#)을 참조하세요.

배치 트랜스크립션 작업에서 다중 채널 오디오를 트랜스크립션하려면 AWS Management Console 또는 [StartMedicalTranscriptionJob](#) API를 사용합니다.

AWS Management Console

AWS Management Console 를 사용하여 배치 트랜스크립션 작업에서 채널 식별을 활성화하려면 오디오 식별을 활성화한 다음 채널 식별을 활성화합니다. 채널 식별은에서 오디오 식별의 하위 집합입니다 AWS Management Console.

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창의 Amazon Transcribe Medical에서 트랜스크립션 작업을 선택합니다.
3. 작업 생성을 선택합니다.
4. 작업 세부 정보 지정 페이지에서 트랜스크립션 작업에 대한 정보를 제공합니다.
5. 다음을 선택합니다.
6. 오디오 식별을 활성화합니다.
7. 오디오 식별 유형에서 채널 식별을 선택합니다.
8. 생성(Create)을 선택합니다.

API

다중 채널 오디오 파일을 트랜스크립션하려면(API)

- [StartMedicalTranscriptionJob](#) API의 경우 다음을 지정하세요.
 - a. TranscriptionJobName에서 AWS 계정의 고유한 이름을 지정합니다.
 - b. LanguageCode에는 오디오 파일에서 사용하는 언어에 해당하는 언어 코드를 지정합니다. 유효한 값은 en-US입니다.
 - c. Media 객체의 MediaFileUri 파라미터에서 트랜스크립션할 미디어 파일의 이름을 지정합니다.
 - d. Settings 객체에서 ChannelIdentification을 true로 설정합니다.

다음은 AWS SDK for Python (Boto3)을 사용한 요청 예시입니다.

```
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

배치 트랜스크립션 작업을 사용하여 다중 채널 오디오 파일을 트랜스크립션하려면(AWS CLI)

- 다음 코드를 실행합니다.

```

aws transcribe start-medical-transcription-job \
--region us-west-2 \
--cli-input-json file://example-start-command.json

```

다음은 `example-start-command.json`의 코드입니다.

```

{
  "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
  }
}

```

다음 코드는 두 채널의 대화가 있는 오디오 파일의 트랜스크립션 출력을 보여줍니다.

```

{

```

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

```

        "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"
        },
        ...
    ]
}

```

}

다중 채널 오디오 스트림 트랜스크립션

[StartMedicalStreamTranscription](#) API를 사용하여 HTTP/2 또는 WebSocket 스트림의 개별 채널에서 오디오를 트랜스크립션할 수 있습니다.

기본적으로 두 채널로 스트림을 트랜스크립션할 수 있습니다. 채널이 3개 이상인 스트림을 트랜스크립팅해야 하는 경우 할당량 증가를 요청할 수 있습니다. 할당량 증가 요청에 대한 자세한 내용은 [AWS 서비스 할당량](#)을 참조하세요.

HTTP/2 스트림에서 다중 채널 트랜스크립션

HTTP/2 스트림에서 다중 채널 오디오를 트랜스크립션하려면 [StartMedicalStreamTranscription](#) API를 사용하고 다음을 지정하세요.

- LanguageCode - 오디오의 언어 코드. 유효한 값은 en-US입니다.
- MediaEncoding - 오디오의 인코딩. 유효한 값은 ogg-opus, flac, pcm입니다.
- EnableChannelIdentification - true
- NumberOfChannels - 스트리밍 오디오의 채널 수.

의료 대화를 트랜스크립션하기 위한 HTTP/2 스트림 설정에 대한 자세한 내용은 [HTTP/2 스트림 설정](#) 섹션을 참조하세요.

WebSocket 스트림에서 다중 채널 오디오 트랜스크립션

WebSocket 스트림에서 화자를 분할하려면 다음 형식을 사용하여 미리 서명된 URI를 만들고 WebSocket 요청을 시작하세요. enable-channel-identification을 true로 지정하고 number-of-channels에서 스트림의 채널 수를 지정합니다. 미리 서명된 URI에는 애플리케이션과 Amazon Transcribe Medical 간의 양방향 통신을 설정하는 데 필요한 정보가 포함되어 있습니다.

```
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
```

파라미터 정의는 [API 참조](#)에서 찾을 수 있습니다. 모든 AWS API 작업에 공통적인 파라미터는 [공통 파라미터](#) 섹션에 나열되어 있습니다.

WebSocket 요청에 대한 자세한 내용은 [WebSocket 스트림 설정](#) 섹션을 참조하세요.

다중 채널 스트리밍 출력

스트리밍 트랜스크립션의 출력은 HTTP/2 및 WebSocket 요청에서 동일합니다. 다음은 예시 출력입니다.

```
{
  "resultId": "XXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
  "startTime": 0.11,
  "endTime": 0.66,
  "isPartial": false,
  "alternatives": [
    {
      "transcript": "Left.",
      "items": [
        {
          "startTime": 0.11,
          "endTime": 0.45,
          "type": "pronunciation",
          "content": "Left",
          "vocabularyFilterMatch": false
        },
        {
          "startTime": 0.45,
          "endTime": 0.45,
          "type": "punctuation",
          "content": ".",
          "vocabularyFilterMatch": false
        }
      ]
    }
  ]
}
```



```

    }
  ]
},
"channelId": "ch_0"
}

```

각 음성 세그먼트에는 음성이 속한 채널을 나타내는 channelId 플래그가 있습니다.

의료 구술 트랜스크립션

Amazon Transcribe Medical을 사용하여 배치 트랜스크립션 작업 또는 실시간 스트림을 사용하여 임상 의가 지시한 의료 기록을 트랜스크립션할 수 있습니다. 배치 트랜스크립션 작업을 사용하면 오디오 파일을 트랜스크립션할 수 있습니다. Medical이 가능한 가장 높은 정확도로 트랜스크립션 결과를 생성할 수 있도록 트랜스크립션 작업 또는 스트림에 임상 의의 Amazon Transcribe 의료 전문 분야를 지정합니다.

다음과 같은 전문 분야의 의료 구술을 트랜스크립션할 수 있습니다.

- 심장학 – 스트리밍 트랜스크립션으로만 사용 가능
- 신경학 – 스트리밍 트랜스크립션으로만 사용 가능
- 종양학 – 스트리밍 트랜스크립션으로만 사용 가능
- 1차 의료 – 다음과 같은 유형의 의료 행위가 포함됩니다.
 - 가정의학
 - 내과학
 - 산부인과학
 - 소아과학
- 방사선학 - 스트리밍 트랜스크립션으로만 사용 가능
- 비뇨기과 – 스트리밍 트랜스크립션으로만 사용 가능

사용자 지정 어휘를 사용하여 트랜스크립션 정확도를 높일 수 있습니다. 사용자 지정 의학 어휘의 작동 방식에 대한 자세한 내용은 [사용자 지정 의학 어휘를 통한 트랜스크립션 정확도 향상](#) 섹션을 참조하세요.

기본적으로 Amazon Transcribe Medical은 신뢰도가 가장 높은 트랜스크립션을 반환합니다. 대체 트랜스크립션을 반환하도록 구성하려면 [대체 트랜스크립션 생성](#) 섹션을 참조하세요.

트랜스크립션 출력에 숫자 및 의료 측정치가 어떻게 표시되는지에 대한 자세한 내용은 [번호 트랜스크립션](#) 및 [의학 용어 및 측정치 트랜스크립션](#) 섹션을 참조하세요.

주제

- [의료 구술의 오디오 파일 트랜스크립션](#)
- [의료 구술을 실시간 스트림으로 트랜스크립션](#)

의료 구술의 오디오 파일 트랜스크립션

배치 트랜스크립션 작업을 사용하면 의료 대화의 오디오 파일을 트랜스크립션할 수 있습니다. 이를 사용하면 임상가와 환자의 대화를 트랜스크립션할 수 있습니다. [StartMedicalTranscriptionJob](#) API 또는 AWS Management Console에서 배치 트랜스크립션 작업을 시작할 수 있습니다.

[StartMedicalTranscriptionJob](#) API로 의료 트랜스크립션 작업을 시작하는 경우 Specialty 파라미터의 값으로 PRIMARYCARE를 지정합니다.

AWS Management Console

임상가와 환자의 대화를 트랜스크립션하려면(AWS Management Console)

AWS Management Console 를 사용하여 임상-환자 대화를 트랜스크립션하려면 트랜스크립션 작업을 생성하고 오디오 입력 유형에 대한 대화를 선택합니다.

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창의 Amazon Transcribe Medical에서 트랜스크립션 작업을 선택합니다.
3. 작업 생성을 선택합니다.
4. DB 세부 정보 지정 페이지의 작업 설정에서 다음을 지정합니다.
 - a. 이름 - 트랜스크립션 작업의 이름
 - b. 오디오 입력 유형 - 구술
5. 나머지 필드에는 오디오 파일의 Amazon S3 위치와 트랜스크립션 작업의 출력을 저장할 위치를 지정합니다.
6. 다음을 선택합니다.
7. 생성(Create)을 선택합니다.

API

배치 트랜스크립션 작업을 사용하여 의료 대화를 트랜스크립션하려면(API)

- [StartMedicalTranscriptionJob](#) API의 경우 다음을 지정하세요.
 - a. MedicalTranscriptionJobName에서 AWS 계정의 고유한 이름을 지정합니다.
 - b. LanguageCode에는 오디오 파일에서 사용하는 언어와 어휘 필터의 언어에 해당하는 언어 코드를 지정합니다.
 - c. Media 객체의 MediaFileUri 파라미터에서 트랜스크립션할 오디오 파일의 이름을 지정합니다.
 - d. Specialty에는 오디오 파일에서 말하는 임상 의학의 전문 분야를 지정합니다.
 - e. Type에서 DICTATION를 지정합니다.
 - f. OutputBucketName에서 트랜스크립션 결과를 저장할 Amazon S3 버킷을 지정합니다.

다음은를 사용하여 PRIMARYCARE 전문 임상 의학의 의료 구술을 AWS SDK for Python (Boto3) 트랜스크립션하는 요청의 예입니다.

```
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)
```

다음 예시 코드는 의료구술의 트랜스크립션 결과를 보여줍니다.

```
{
  "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

배치 트랜스크립션 작업에서 화자 파티셔닝을 활성화하려면(AWS CLI)

- 다음 코드를 실행합니다.

```

aws transcribe start-medical-transcription-job \
--region us-west-2 \
--cli-input-json file://example-start-command.json

```

다음 코드는 *example-start-command.json*의 내용을 보여줍니다.

```

{
  "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"
}

```

의료 구술을 실시간 스트림으로 트랜스크립션

WebSocket 스트림을 사용하여 의료 구술을 오디오 스트림으로 트랜스크립션할 수 있습니다. 또한 이를 사용하여 본인 또는 다른 사람이 마이크에 직접 말하는 음성을 트랜스크립션 AWS Management Console 할 수 있습니다.

HTTP/2 또는 WebSocket 스트림의 경우 다음과 같은 의료 전문 분야에서 오디오를 트랜스크립션할 수 있습니다.

- 심장학
- 종양학
- 신경학
- 1차 의료
- 방사선학
- 비뇨기과

각 의료 전문 분야에는 다양한 유형의 시술과 예약이 포함됩니다. 따라서 임상 의는 다양한 유형의 메모를 작성합니다. WebSocket 요청의 specialty URI 파라미터 값 또는 [StartMedicalStreamTranscription](#) API의 Specialty 파라미터를 지정하는 데 도움이 되는 지침으로 다음 예시를 사용하세요.

- 전기생리학 또는 심초음파 시술 후 구술을 원하시면 CARDIOLOGY를 선택합니다.
- 외과적 종양학 또는 방사선 종양학 시술 후 구술을 원하시면 ONCOLOGY를 선택합니다.
- 의사가 구술한 뇌염 진단에 관한 소견서를 원하시면 NEUROLOGY를 선택합니다.
- 방광 결석 분리를 위한 시술 기록의 구술을 원하시면 UROLOGY를 선택합니다.
- 내과 진료 후 임상 의의 소견서의 구술을 원하시면 PRIMARYCARE를 선택합니다.
- CT 스캔, PET 스캔, MRI 또는 방사선 촬영 결과를 전달하는 의사의 구술을 보려면 RADIOLOGY를 선택합니다.
- 부인과 상담 후 의사의 소견서의 구술을 원하시면 PRIMARYCARE를 선택합니다.

실시간 스트림에서 특정 용어의 트랜스크립션 정확도를 높이려면 사용자 지정 어휘를 사용하세요. 사용자 지정 어휘를 활성화하려면 vocabulary-name 값을, 사용할 사용자 지정 어휘의 이름으로 설정하세요.

를 사용하여 마이크에 적힌 구술 트랜스크립션 AWS Management Console

AWS Management Console 를 사용하여 의료 구술의 스트리밍 오디오를 트랜스크립션하려면 의료 구술을 트랜스크립션하고 스트림을 시작한 다음 마이크에 말하기 시작하는 옵션을 선택합니다.

의료 구술의 스트리밍 오디오를 트랜스크립션하려면(AWS Management Console)

1. [AWS Management Console](#)에 로그인합니다.

2. 탐색 창의 Amazon Transcribe Medical에서 실시간 트랜스크립션을 선택합니다.
3. 구술을 선택합니다.
4. 의료 전문 분야에는 스트림에서 말하는 임상치의 의료 전문 분야를 선택합니다.
5. 스트리밍 시작을 선택합니다.
6. 마이크에 대고 말합니다.

HTTP/2 스트림에서 구술 트랜스크립션

의료 구술의 HTTP/2 스트림을 트랜스크립션하려면 [StartMedicalStreamTranscription](#) API를 사용하고 다음을 지정하세요.

- LanguageCode - 언어 코드. 유효한 값은 en-US입니다.
- MediaEncoding - 입력 오디오에 사용된 인코딩. 유효한 값은 pcm, ogg-opus, flac입니다.
- Specialty - 의료 전문가의 전문 분야.
- Type - DICTATION

의료 구술을 트랜스크립션하기 위한 HTTP/2 스트림 설정에 대한 자세한 내용은 [HTTP/2 스트림 설정](#) 섹션을 참조하세요.

WebSocket 스트림 요청을 사용하여 의료 구술 트랜스크립션

WebSocket 요청을 사용하여 의료 구술을 실시간 스트림으로 트랜스크립션하려면 미리 서명된 URI를 생성해야 합니다. 이 URI에는 애플리케이션과 Amazon Transcribe Medical 간에 오디오 스트림을 설정하는 데 필요한 정보가 포함되어 있습니다. WebSocket 요청 생성에 대한 자세한 내용은 [WebSocket 스트림 설정](#) 섹션을 참조하세요.

다음 템플릿을 사용하여 미리 서명된 URI를 생성합니다.

```
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
```

미리 서명된 URI 생성에 대한 자세한 내용은 [WebSocket 스트림 설정](#) 섹션을 참조하세요.

사용자 지정 의학 어휘를 통한 트랜스크립션 정확도 향상

Amazon Transcribe Medical에서 트랜스크립션 정확도를 높이려면 하나 이상의 의료 사용자 지정 어휘를 생성하고 사용합니다. 사용자 지정 어휘는 도메인별 단어 또는 구절의 모음입니다. 이 모음은 해당 단어 또는 구문을 트랜스크립션하는 데 있어 Amazon Transcribe Medical의 성능을 개선하는 데 도움이 됩니다.

Amazon Transcribe Medical을 사용할 때 자체 데이터의 무결성에 대한 책임은 사용자에게 있습니다. 기밀 정보, 개인 정보(PII) 또는 보호 대상 건강 정보(PHI)를 사용자 지정 어휘에 입력하지 마세요.

최상의 결과를 얻으려면 각각 특정 오디오 녹음을 트랜스크립션하는 데 도움이 되는 작은 사용자 지정 어휘를 별도로 만드세요. 모든 녹음에 사용할 큰 사용자 지정 어휘를 하나 생성했을 때보다 필사 정확도가 크게 향상됩니다.

기본적으로에는 최대 100개의 사용자 지정 어휘가 있을 수 있습니다 AWS 계정. 사용자 지정 어휘의 용량은 50KB를 초과할 수 없습니다. 에 포함할 수 있는 사용자 지정 어휘 수 증가를 요청하는 방법에 대한 자세한 내용은 서비스 할당량을 AWS 계정참조하세요. [AWS](#)

사용자 지정 어휘는 미국 영어(en-US)로 제공됩니다.

주제

- [사용자 지정 의학 어휘를 위한 텍스트 파일 생성](#)
- [텍스트 파일을 사용하여 사용자 지정 의학 어휘 생성](#)
- [사용자 지정 의학 어휘를 사용한 오디오 파일 트랜스크립션](#)
- [사용자 지정 의학 어휘를 사용한 실시간 스트림의 트랜스크립션](#)
- [Amazon Transcribe Medical의 문자 집합](#)

사용자 지정 의학 어휘를 위한 텍스트 파일 생성

사용자 지정 어휘를 생성하려면 UTF-8 형식의 텍스트 파일을 만듭니다. 이 파일에서는 4열 테이블을 만들고 각 열에서 필드를 지정합니다. 각 필드는 도메인별 용어가 발음되는 방식 또는 트랜스크립션에 이러한 용어를 표시하는 방법을 Amazon Transcribe Medical에 알려줍니다. 이러한 필드가 포함된 텍스트 파일을 Amazon S3 버킷에 저장합니다.

텍스트 파일 형식 지정 방법 이해

사용자 지정 의학 어휘를 생성하려면 열 이름을 머리글 행으로 입력합니다. 머리글 행 아래에 각 열의 값을 입력합니다.

테이블에 있는 4개 열의 이름은 다음과 같습니다.

- **Phrase** - 열은 필수, 값은 필수
- **IPA** - 열은 필수, 값은 선택 사항일 수 있음
- **SoundsLike** - 열은 필수, 값은 선택 사항일 수 있음
- **DisplayAs** - 열은 필수, 값은 선택 사항일 수 있음

사용자 지정 어휘를 만드는 경우 다음 사항을 확인하세요.

- 각 열을 단일 Tab 문자로 구분합니다. 열을 공백 또는 여러 Tab 문자로 구분하려고 하면 오류 메시지가 Amazon Transcribe 표시됩니다.
- 열 내 각 값 뒤에 후행 공백이나 여백이 없어야 합니다.

각 열에 입력하는 값이 다음과 같은지 확인하세요.

- 256자 미만(하이픈 포함)
- 허용된 문자 집합의 문자만 사용하세요([Amazon Transcribe Medical의 문자 집합](#) 섹션 참조).

테이블의 열에 대한 값 입력

다음 정보는 테이블의 4개 열에 대한 값을 지정하는 방법을 보여줍니다.

- **Phrase** - 인식해야 하는 단어 또는 구절. 이 열에 값을 입력해야 합니다.

구절을 입력할 경우 단어는 하이픈(-)으로 구분합니다. 예를 들면 **cerebral autosomal dominant arteriopathy with subcortical infarcts and leukoencephalopathy**를

cerebral-autosomal-dominant-arteriopathy-with-subcortical-infarcts-and-leukoencephalopathy 형식으로 입력합니다.

머리글자어 또는 문자를 개별적으로 발음해야 하는 기타 단어를 점으로 구분된 단일 문자로 입력합니다(예: **D.N.A.** 또는 **S.T.E.M.I.**). “STEMIs”와 같은 복수 형태의 머리글자어를 입력하려면 하이픈을 사용하여 머리글자어에서 “s”를 구분합니다(예: “**S.T.E.M.I-s**”). 머리글자어는 대문자 또는 소문자를 사용할 수 있습니다.

Phrase 열은 필수 항목입니다. 허용되는 모든 문자를 입력 언어로 사용할 수 있습니다. 허용되는 문자는 [Amazon Transcribe Medical의 문자 집합](#) 단원을 참조하세요. DisplayAs 열을 지정하지 않으면 Amazon Transcribe Medical은 출력 파일의 Phrase 열 내용을 사용합니다.

- **IPA**(열은 필수, 값은 선택 사항일 수 있음) - 단어나 구절의 발음을 지정하려면 이 열의 [IPA\(International Phonetic Alphabet\)](#)에 문자를 포함하면 됩니다. IPA 열에는 선행 또는 후행 공백을 포함할 수 없으며 입력에서 단일 공백을 사용하여 각 음소를 구분해야 합니다. 예를 들어 영어에서는 **acute-respiratory-distress-syndrome** 구절을 **ə k j u t # # s p # # ə t # # i d # s t # # s s # n d # o # m** 형식으로 입력합니다. **A.L.L.** 구절은 **e # # l # l** 형식으로 입력합니다.

IPA 열의 콘텐츠를 지정하지 않아도 빈 IPA 열을 포함해야 합니다. IPA 열에 값을 포함하는 경우 SoundsLike 열의 값을 제공할 수 없습니다.

특정 언어에 대해 허용되는 IPA 문자 목록은 [Amazon Transcribe Medical의 문자 집합](#) 단원을 참조하세요. 미국 영어는 Amazon Transcribe Medical에서 사용할 수 있는 유일한 언어입니다.

- **SoundsLike**(열은 필수, 값은 선택 사항일 수 있음) - 단어나 구절을 보다 작은 조각으로 분리하고 언어의 표준 철자법을 사용하여 각 조각의 발음을 제공하여 단어의 소리를 모방할 수 있습니다. 예를 들어 **cerebral-autosomal-dominant-arteriopathy-with-subcortical-infarcts-and-leukoencephalopathy** 구절에 대한 발음 힌트는 **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**의 형식으로 제공할 수 있습니다. **atrioventricular-nodal-reentrant-tachycardia** 구절에 대한 힌트는 **ay-tree-o-ven-trick-u-lar-node-al-re-entr-ant-tack-ih-card-ia**의 형식으로 제공할 수 있습니다. 힌트의 각 부분은 하이픈(-)으로 구분합니다.

SoundsLike 열의 값을 제공하지 않아도 빈 SoundsLike 열을 포함해야 합니다. SoundsLike 열에 값을 포함하는 경우 IPA 열의 값을 제공할 수 없습니다.

허용되는 모든 문자를 입력 언어로 사용할 수 있습니다. 허용되는 문자 목록은 [Amazon Transcribe Medical의 문자 집합](#) 단원을 참조하세요.

- **DisplayAs**(열은 필수, 값은 선택 사항일 수 있음) - 출력 시 단어나 구절이 어떻게 보이는지 정의합니다. 예를 들어 단어 또는 구절이 **cerebral-autosomal-dominant-arteriopathy-with-subcortical-infarcts-and-leukoencephalopathy**일 경우 표시 형식을 cerebral autosomal dominant arteriopathy with subcortical infarcts and leukoencephalopathy로 지정하여 하이픈이 포함되지 않도록 할 수 있습니다. 출력에서 전체 용어 대신 머리글자어를 표시하려는 DisplayAs를 CADASIL 형식으로 지정할 수도 있습니다.

DisplayAs 열을 지정하지 않으면 Amazon Transcribe Medical은 출력의 입력 파일에서 Phrase 열을 사용합니다.

DisplayAs 열에 UTF-8 문자를 사용할 수 있습니다.

IPA 및 DisplayAs 열의 값에만 공백을 포함할 수 있습니다.

사용자 지정 어휘의 텍스트 파일을 생성하려면 텍스트 파일의 각 단어나 문구를 별도의 줄에 배치하세요. 탭 문자로 열을 구분합니다. IPA 및 DisplayAs 열의 값에만 공백을 포함합니다. 확장자가 있는 파일을 Amazon Transcribe Medical.txt을 사용하여 사용자 지정 어휘를 생성하는 AWS 리전 동일한의 Amazon S3 버킷에 저장합니다.

Windows에서 텍스트 파일을 편집하는 경우 파일 형식이 LF가 아닌 CRLF인지 확인합니다. 그렇지 않으면 사용자 지정 어휘를 작성할 수 없습니다. 일부 텍스트 편집기에서는 찾기 및 바꾸기 명령을 사용하여 서식을 변경할 수 있습니다.

다음 예시에서는 사용자 지정 어휘를 작성하는 데 사용할 수 있는 텍스트를 보여줍니다. 이러한 예제에서 사용자 정의 어휘를 작성하려면 예시를 텍스트 편집기에 복사하고 [TAB]을 탭 문자로 바꾸고 저장된 텍스트 파일을 Amazon S3에 업로드합니다.

```
Phrase[TAB]IPA[TAB]SoundsLike[TAB]DisplayAs
acute-respiratory-distress-syndrome[TAB][TAB][TAB]acute respiratory distress syndrome
A.L.L.[TAB]e# # 1 # 1[TAB][TAB]ALL
atrioventricular-nodal-reentrant-tachycardia[TAB][TAB]ay-tree-o-ven-trick-u-lar-node-
al-re-entr-ant-tack-ih-card-ia[TAB]
```

원하는 순서로 열을 입력할 수 있습니다. 다음 예시에서는 사용자 지정 어휘 입력 파일의 다른 유효한 구조를 보여줍니다.

```
Phrase[TAB]SoundsLike[TAB]IPA[TAB]DisplayAs
acute-respiratory-distress-syndrome[TAB][TAB][TAB]acute respiratory distress syndrome
A.L.L.[TAB][TAB]e# # 1 # 1[TAB]ALL
```

```
atrioventricular-nodal-reentrant-tachycardia[TAB]ay-tree-o-ven-trick-u-lar-node-al-re-entr-ant-tack-ih-card-ia[TAB][TAB]
```

```
DisplayAs[TAB]SoundsLike[TAB]IPA[TAB]Phrase
acute respiratory distress syndrome[TAB][TAB][TAB]acute-respiratory-distress-syndrome
ALL[TAB][TAB]e# # 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
```

읽기 쉽도록 다음 테이블은 위의 예시를 html 형식으로 더 명확하게 보여줍니다. 예시를 설명하기 위한 용도로만 사용됩니다.

구절	IPA	SoundsLike	DisplayAs
acute-respiratory-distress-syndrome			acute respiratory distress syndrome
A.L.L.	eɪ ɛ l ɛ l		ALL
atrioventricular-nodal-reentrant-tachycardia		ay-tree-o-ven-trick-u-lar-node-al-re-entr-ant-tack-ih-card-ia	

구절	SoundsLike	IPA	DisplayAs
acute-respiratory-distress-syndrome			acute respiratory distress syndrome
atrioventricular-nodal-reentrant-tachycardia	ay-tree-o-ven-trick-u-lar-node-al-re-entr-ant-tack-ih-card-ia		
A.L.L.		eɪ ɛ l ɛ l	ALL

DisplayAs	SoundsLike	IPA	구절
acute respiratory distress syndrome			acute-respiratory-distress-syndrome
ALL		eɪ ɛ l ɛ l	A.L.L.
	ay-tree-o-ven-trick-ular-node-al-re-entr-ant-tack-ih-card-ia		atrioventricular-nodal-reentrant-tachycardia

텍스트 파일을 사용하여 사용자 지정 의학 어휘 생성

사용자 지정 어휘를 생성하려면 단어 또는 문구 모음이 포함된 텍스트 파일을 준비해야 합니다. Amazon Transcribe Medical은 이 텍스트 파일을 사용하여 해당 단어 또는 문구의 트랜스크립션 정확도를 개선하는 데 사용할 수 있는 사용자 지정 어휘를 생성합니다. [CreateMedicalVocabulary](#) API 또는 Amazon Transcribe Medical 콘솔을 사용하여 사용자 지정 어휘를 생성할 수 있습니다.

AWS Management Console

AWS Management Console 를 사용하여 사용자 지정 어휘를 생성하려면 단어 또는 문구가 포함된 텍스트 파일의 Amazon S3 URI를 제공합니다.

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창의 Amazon Transcribe Medical에서 사용자 지정 어휘를 선택합니다.
3. 이름의 경우 어휘 설정에서 사용자 지정 어휘의 이름을 선택합니다.
4. Amazon S3에서 오디오 파일 또는 비디오 파일의 위치를 지정합니다.
 - 어휘 설정의 S3의 어휘 입력 파일 위치에서 사용자 지정 어휘를 생성하는 데 사용할 텍스트 파일을 식별하는 Amazon S3 URI를 지정합니다.
 - S3의 어휘 입력 파일 위치에서 Browse S3를 선택하여 텍스트 파일을 찾아 선택합니다.
5. 어휘 생성을 선택합니다.

AWS Management Console에서 사용자 지정 어휘의 처리 상태를 확인할 수 있습니다.

API

사용자 지정 의학 어휘를 생성하려면(API)

- [StartTranscriptionJob](#) API의 경우 다음을 지정하세요.
 - a. LanguageCode에서 en-US를 지정합니다.
 - b. 에서 사용자 지정 어휘를 정의하는 데 사용하는 텍스트 파일의 Amazon S3 위치를 VocabularyFileUri 지정합니다.
 - c. VocabularyName에서 사용자 지정 어휘의 이름을 지정합니다. 지정하는 이름은 내에서 고 유해야 합니다 AWS 계정.

사용자 지정 어휘의 처리 상태를 보려면 [GetMedicalVocabulary](#) API를 사용하세요.

다음은를 사용하여 사용자 지정 어휘를 AWS SDK for Python (Boto3) 생성하는 요청의 예입니다.

```
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

배치 트랜스크립션 작업에서 화자 파티셔닝을 활성화하려면(AWS CLI)

- 다음 코드를 실행합니다.

```
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
```

사용자 지정 의학 어휘를 사용한 오디오 파일 트랜스크립션

[StartMedicalTranscriptionJob](#) 또는를 사용하여 사용자 지정 어휘를 사용하여 트랜스크립션 정확도를 개선하는 트랜스크립션 작업을 AWS Management Console 시작합니다.

AWS Management Console

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창의 Amazon Transcribe Medical에서 트랜스크립션 작업을 선택합니다.
3. 작업 생성을 선택합니다.
4. 작업 세부 정보 지정 페이지에서 트랜스크립션 작업에 대한 정보를 제공합니다.
5. 다음을 선택합니다.
6. 사용자 지정에서 사용자 지정 어휘를 활성화합니다.
7. 어휘 선택에서 사용자 지정 어휘를 선택합니다.
8. 생성(Create)을 선택합니다.

API

배치 트랜스크립션 작업을 사용하여 오디오 파일의 화자 파티셔닝을 활성화하려면(API)

- [StartMedicalTranscriptionJob](#) API의 경우 다음을 지정하세요.
 - a. `MedicalTranscriptionJobName`에서 AWS 계정의 고유한 이름을 지정합니다.

- b. LanguageCode에는 오디오 파일에서 사용하는 언어와 어휘 필터의 언어에 해당하는 언어 코드를 지정합니다.
- c. Media 객체의 MediaFileUri 파라미터에서 트랜스크립션할 오디오 파일의 이름을 지정합니다.
- d. Specialty에는 오디오 파일에서 말하는 임상치의 전문 분야를 지정합니다.
- e. Type에는 오디오 파일이 대화인지 구술인지를 지정합니다.
- f. OutputBucketName에서 트랜스크립션 결과를 저장할 Amazon S3 버킷을 지정합니다.
- g. Settings 객체에서 다음을 지정합니다.
 - VocabularyName - 사용자 지정 어휘의 이름.

다음 요청은 AWS SDK for Python (Boto3) 를 사용하여 사용자 지정 어휘로 배치 트랜스크립션 작업을 시작합니다.

```
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)

```

사용자 지정 의학 어휘를 사용한 실시간 스트림의 트랜스크립션

실시간 스트림에서 트랜스크립션 정확도를 높이려면 HTTP/2 또는 WebSocket 스트림을 사용하여 사용자 지정 어휘를 사용할 수 있습니다. HTTP/2 요청을 시작하려면 [StartMedicalStreamTranscription](#) API를 사용하세요. 사용자 지정 어휘는, AWS Management Console [StartMedicalStreamTranscription](#) API 또는 WebSocket 프로토콜을 사용하여 실시간으로 사용할 수 있습니다.

마이크에 대고 말하는 구술의 트랜스크립션(AWS Management Console)

AWS Management Console 를 사용하여 의료 구술의 스트리밍 오디오를 트랜스크립션하려면 의료 구술을 트랜스크립션하고 스트림을 시작한 다음 마이크에 말하기 시작하는 옵션을 선택합니다.

의료 구술의 스트리밍 오디오를 트랜스크립션하려면(AWS Management Console)

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창의 Amazon Transcribe Medical에서 실시간 트랜스크립션을 선택합니다.
3. 의료 전문 분야에는 스트림에서 말하는 임상어의 의료 전문 분야를 선택합니다.
4. 오디오 입력 유형에서 대화 또는 구술을 선택합니다.
5. 추가 설정에서 사용자 지정 어휘를 선택합니다.
 - 어휘 선택에서 사용자 지정 어휘를 선택합니다.
6. 스트리밍 시작을 선택합니다.
7. 마이크에 대고 말합니다.

HTTP/2 스트림에서 화자 파티셔닝 활성화

다음은 HTTP/2 요청 파라미터의 구문입니다.

```
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

```

파라미터 설명:

- 호스트: 호출하려는 로 AWS 리전 (이전 예제의 'us-west-2')를 업데이트 AWS 리전 합니다. 유효한 목록은 및 엔드포인트를 AWS 리전참조하세요. [AWS 리전](#)
- 권한: 생성된 필드. 서명 생성에 대한 자세한 내용은 [서명 버전 4로 AWS 요청 서명을 참조하세요](#).
- x-amz-target: 이 필드를 변경하지 말고 이전 예시에 표시된 콘텐츠를 사용하세요.
- x-amz-content-sha256: 생성된 필드. 서명 계산에 대한 자세한 내용은 [서명 버전 4로 AWS 요청 서명을 참조하세요](#).
- x-amz-date: 서명이 생성된 날짜 및 시간. 형식은 YYYYMMDDTHHMMSSZ입니다. 여기서 YYYY=년, MM=월, DD=일, HH=시간, MM=분, SS=초이고, 'T'와 'Z'는 고정 문자입니다. 자세한 내용은 [서명 버전 4의 날짜 처리](#)를 참조하세요.
- x-amzn-transcribe-session-id: 스트리밍 세션의 이름.
- x-amzn-transcribe-language-code: 입력 오디오에 사용되는 인코딩. 유효한 값 목록은 [StartMedicalStreamTranscription](#) 또는 [지원되는 언어 및 언어별 기능](#) 섹션을 참조하세요.
- x-amzn-transcribe-media-encoding: 입력 오디오에 사용되는 인코딩. 유효한 값은 pcm, ogg-opus, flac입니다.
- x-amzn-transcribe-sample-rate: 입력 오디오의 샘플 속도(Hz)입니다.는 8,000Hz~48,000Hz의 범위를 Amazon Transcribe 지원합니다. 전화 오디오와 같은 저품질 오디오는 일반적으로 약 8,000Hz입니다. 고품질 오디오의 범위는 일반적으로 16,000Hz에서 48,000Hz입니다. 단, 지정하는 샘플 속도는 오디오의 샘플 속도와 반드시 일치해야 합니다.
- x-amzn-transcribe-vocabulary-name: 트랜스크립션에 사용하려는 어휘의 이름.

- `x-amzn-transcribe-specialty`: 트랜스크립션 대상 의료 전문 분야.
- `x-amzn-transcribe-type`: 구술인지 대화인지를 선택하세요.
- `x-amzn-transcribe-show-speaker-label`: 분할을 활성화하려면 이 값이 `true`여야 합니다.
- `content-type`: 이 필드를 변경하지 말고 이전 예시에 표시된 콘텐츠를 사용하세요.

WebSocket 요청에서 화자 파티셔닝 활성화

API를 사용하여 WebSocket 스트림에서 화자를 분할하려면 다음 형식을 사용하여 미리 서명된 URL을 만들고 WebSocket 요청을 시작하여 `vocabulary-name`을 사용자 지정 어휘의 이름으로 설정하세요.

```
GET wss://transcribestreaming.us-west-2.amazonaws.com:8443/medical-stream-
transcription-websocket
?language-code=en-US
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIAIOSFODNN7EXAMPLE%2F20220208%2Fus-
west-2%2Ftranscribe%2Faws4_request
&X-Amz-Date=20220208T235959Z
&X-Amz-Expires=300
&X-Amz-Security-Token=security-token
&X-Amz-Signature=Signature Version 4 signature
&X-Amz-SignedHeaders=host
&media-encoding=flac
&sample-rate=16000
&session-id=sessionId
&specialty=medicalSpecialty
&type=CONVERSATION
&vocabulary-name=vocabularyName
&show-speaker-label=boolean
```

Amazon Transcribe Medical의 문자 집합

Amazon Transcribe Medical에서 사용자 지정 어휘를 사용하려면 다음 문자 집합을 사용합니다.

영어 문자 집합

영어 사용자 지정 어휘의 경우 `Phrase` 및 `SoundsLike` 열에 다음 문자를 사용할 수 있습니다.

- a - z
- A - Z

- ' (아포스트로피)
- - (하이픈)
- . (마침표)

어휘 입력 파일의 IPA 열에 다음 IPA(International Phonetic Alphabet) 문자를 사용할 수 있습니다.

문자	코드	문자	코드
au	0061 028A	w	0077
aɪ	0061 026A	z	007A
b	0062	æ	00E6
d	0064	ð	00F0
eɪ	0065 026A	ŋ	014B
f	0066	ɑ	0251
g	0067	ɔ	0254
h	0068	ɔɪ	0254 026A
i	0069	ə	0259
j	006A	ɛ	025B
k	006B	ʒ	025D
l	006C	g	0261
ɹ	006C 0329	ɪ	026A
m	006D	ɹ	0279
n	006E	ʃ	0283
ŋ	006E 0329	ʊ	028A

문자	코드	문자	코드
ou	006F 028A	ʌ	028C
p	0070	m	028D
s	0073	ʒ	0292
t	0074	dʒ	02A4
u	0075	tʃ	02A7
v	0076	θ	03B8

트랜스크립션의 개인 건강 정보(PHI) 식별

개인 건강 정보 식별을 사용하여 트랜스크립션 결과의 개인 건강 정보(PHI)에 레이블을 지정합니다. 레이블을 검토하여 환자를 식별하는 데 사용할 수 있는 PHI를 찾을 수 있습니다.

실시간 스트림 또는 배치 트랜스크립션 작업을 사용하여 PHI를 식별할 수 있습니다.

자체 후처리를 사용하여 트랜스크립션 출력에서 확인된 PHI를 수정할 수 있습니다.

개인 건강 정보 식별을 사용하여 다음 유형의 PHI를 확인합니다.

- 개인 PHI:
 - 이름 – 성명 또는 성 및 이니셜
 - Gender
 - 나이
 - 전화번호
 - 환자와 직접 관련된 날짜(연도 제외)
 - 이메일 주소
- 지리적 PHI:
 - 실제 주소
 - 우편 번호
 - 의료 센터 또는 진료소 이름
- 계정 PHI:

- 팩스 번호
- 사회 보장 번호(SSN)
- 건강 보험 수혜자 번호
- 계정 번호
- 인증서 또는 면허 번호
- 차량 PHI:
 - 차량 식별 번호(VIN)
 - 번호판 번호
- 기타 PHI:
 - 웹 URL(Uniform Resource Location)
 - 인터넷 프로토콜(IP) 주소 번호

Amazon Transcribe Medical은 HIPAA(Health Insurance Portability and Accountability Act of 1996) 적격 서비스입니다. 자세한 내용은 [Amazon Transcribe Medical](#) 단원을 참조하십시오. 오디오 파일에서 PHI를 식별하는 방법에 대한 자세한 내용은 [오디오 파일에서 PHI 식별](#) 섹션을 참조하세요. 스트림에서 PHI를 식별하는 방법에 대한 자세한 내용은 [실시간 스트림에서 PHI 식별](#) 섹션을 참조하세요.

주제

- [오디오 파일에서 PHI 식별](#)
- [실시간 스트림에서 PHI 식별](#)

오디오 파일에서 PHI 식별

배치 트랜스크립션 작업을 사용하여 오디오 파일을 트랜스크립션하고 그 안에 있는 개인 건강 정보 (PHI)를 식별할 수 있습니다. 개인 건강 정보(PHI) 식별을 활성화하면 Amazon Transcribe Medical은 트랜스크립션 결과에서 식별한 PHI에 레이블을 지정합니다. Amazon Transcribe Medical에서 식별할 수 있는 PHI에 대한 자세한 내용은 섹션을 참조하세요 [트랜스크립션의 개인 건강 정보\(PHI\) 식별](#).

[StartMedicalTranscriptionJob](#) API 또는 AWS Management Console을 사용하여 배치 트랜스크립션 작업을 시작할 수 있습니다.

AWS Management Console

AWS Management Console 를 사용하여 임상의-환자 대화를 트랜스크립션하려면 트랜스크립션 작업을 생성하고 오디오 입력 유형에 대한 대화를 선택합니다.

오디오 파일을 트랜스크립션하고 해당 PHI를 식별하려면(AWS Management Console)

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창의 Amazon Transcribe Medical에서 트랜스크립션 작업을 선택합니다.
3. 작업 생성을 선택합니다.
4. DB 세부 정보 지정 페이지의 작업 설정에서 다음을 지정합니다.
 - a. 이름 - 고유한 트랜스크립션 작업의 이름입니다 AWS 계정.
 - b. 오디오 입력 유형 - 대화 또는 구술.
5. 나머지 필드에는 오디오 파일의 Amazon S3 위치와 트랜스크립션 작업의 출력을 저장할 위치를 지정합니다.
6. 다음을 선택합니다.
7. 오디오 설정에서 PHI 식별을 선택합니다.
8. 생성(Create)을 선택합니다.

API

배치 트랜스크립션 작업을 사용하여 오디오 파일을 트랜스크립션하고 해당 PHI를 식별하려면(API)

- [StartMedicalTranscriptionJob](#) API의 경우 다음을 지정하세요.
 - a. MedicalTranscriptionJobName에서 AWS 계정의 고유한 이름을 지정합니다.
 - b. LanguageCode에는 오디오 파일에서 사용하는 언어에 해당하는 언어 코드를 지정합니다.
 - c. Media 객체의 MediaFileUri 파라미터에서 트랜스크립션할 오디오 파일의 이름을 지정합니다.
 - d. Specialty에는 오디오 파일에서 말하는 임상사의 전문 분야를 PRIMARYCARE로 지정합니다.
 - e. Type에서 CONVERSATION 또는 DICTATION를 지정합니다.
 - f. OutputBucketName에서 트랜스크립션 결과를 저장할 Amazon S3 버킷을 지정합니다.

다음은를 사용하여 오디오 파일을 AWS SDK for Python (Boto3) 트랜스크립션하고 환자의 PHI를 식별하는 요청의 예입니다.

```
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)

```

다음 예시 코드는 환자 PHI를 식별한 트랜스크립션 결과를 보여줍니다.

```

{
  "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

배치 트랜스크립션 작업을 사용하여 오디오 파일을 트랜스크립션하고 PHI를 식별하려면(AWS CLI)

- 다음 코드를 실행합니다.

```

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
```

실시간 스트림에서 PHI 식별

HTTP/2 또는 WebSocket 스트림에서 개인 건강 정보(PHI)를 식별할 수 있습니다. PHI 식별을 활성화하면 Amazon Transcribe Medical은 트랜스크립션 결과에서 식별하는 PHI에 레이블을 지정합니다. Amazon Transcribe Medical에서 식별할 수 있는 PHI에 대한 자세한 내용은 섹션을 참조하세요 [트랜스크립션의 개인 건강 정보\(PHI\) 식별](#).

마이크에 대고 말하는 구술로 PHI 식별하기

를 사용하여 마이크가 선택한 음성 AWS Management Console 을 트랜스크립션하고 PHI를 식별하려면 오디오 입력 유형으로 구술을 선택하고 스트림을 시작한 다음 컴퓨터의 마이크에 대고 말하기 시작합니다.

를 사용하여 구술에서 PHI를 식별하려면 AWS Management Console

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창에서 실시간 트랜스크립션을 선택합니다.
3. 오디오 입력 유형에서 구술을 선택합니다.
4. 추가 설정에서 PHI 식별을 선택합니다.
5. 스트리밍 시작을 선택하고 마이크에 대고 말합니다.
6. 구술을 종료하려면 스트리밍 중지를 선택합니다.

HTTP/2 스트림에서 PHI 식별하기

PHI 식별이 활성화된 상태에서 HTTP/2 스트림을 시작하려면 [StartMedicalStreamTranscription](#) API를 사용하고 다음을 지정하세요.

- LanguageCode에는 스트림에서 사용하는 언어에 대한 언어 코드를 지정합니다. 미국 영어의 경우에는 en-US를 지정합니다.

- MediaSampleHertz에서 오디오의 샘플 속도를 지정합니다.
- content-identification-type에서 PHI를 지정합니다.

WebSocket 스트림에서 PHI 식별하기

PHI 식별이 활성화된 WebSocket 스트림을 시작하려면 다음 형식을 사용하여 미리 서명된 URL을 만드세요.

```
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
```

파라미터 정의는 [API 참조](#)에서 찾을 수 있습니다. 모든 AWS API 작업에 공통적인 파라미터는 [공통 파라미터](#) 섹션에 나열되어 있습니다.

대체 트랜스크립션 생성

Amazon Transcribe Medical을 사용하면 신뢰도가 가장 높은 트랜스크립션을 얻을 수 있습니다. 그러나 신뢰도가 낮은 추가 트랜스크립션을 반환하도록 Amazon Transcribe Medical을 구성할 수 있습니다.

대체 트랜스크립션을 사용하여 트랜스크립션된 오디오의 다른 해석을 볼 수 있습니다. 예를 들어, 사람이 트랜스크립션을 검토할 수 있도록 하는 애플리케이션에 사람이 선택할 수 있는 대체 트랜스크립션을 제공할 수 있습니다.

AWS Management Console 또는 [StartMedicalTranscriptionJob](#) API를 사용하여 대체 트랜스크립션을 생성할 수 있습니다.

AWS Management Console

를 사용하여 대체 트랜스크립션 AWS Management Console 을 생성하려면 작업을 구성할 때 대체 결과를 활성화합니다.

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창의 Amazon Transcribe Medical에서 트랜스크립션 작업을 선택합니다.
3. 작업 생성을 선택합니다.
4. 작업 세부 정보 지정 페이지에서 트랜스크립션 작업에 대한 정보를 제공합니다.
5. 다음을 선택합니다.
6. 대체 결과를 활성화합니다.
7. 최대 대안에는 출력에 포함할 대체 트랜스크립션 최대 개수를 2에서 10 사이의 정수 값으로 입력합니다.
8. 생성(Create)을 선택합니다.

API

배치 트랜스크립션 작업을 사용하여 오디오 파일에서 화자별로 텍스트를 구분하려면(API)

- [StartMedicalTranscriptionJob](#) API의 경우 다음을 지정하세요.
 - a. MedicalTranscriptionJobName에서 AWS 계정의 고유한 이름을 지정합니다.
 - b. LanguageCode에는 오디오 파일에서 사용하는 언어와 어휘 필터의 언어에 해당하는 언어 코드를 지정합니다.
 - c. Media 객체의 MediaFileUri 파라미터에서 트랜스크립션할 오디오 파일의 위치를 지정합니다.
 - d. Specialty에는 오디오 파일에서 말하는 임상 의의 전문 분야를 지정합니다.
 - e. Type에서 의료 대화 또는 구술을 트랜스크립션할지 여부를 지정합니다.
 - f. OutputBucketName에서 트랜스크립션 결과를 저장할 Amazon S3 버킷을 지정합니다.
 - g. Settings 객체에서 다음을 지정합니다.
 - i. ShowAlternatives – true.
 - ii. MaxAlternatives - 트랜스크립션 출력에 포함하려는 대체 트랜스크립트의 수를 나타내는 2에서 10 사이의 정수.

다음 요청은 AWS SDK for Python (Boto3) 를 사용하여 최대 2개의 대체 트랜스크립션을 생성하는 트랜스크립션 작업을 시작합니다.

```
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

1차 의료 임상가와 환자 간의 대화를 녹음한 오디오 파일을 오디오 파일에 트랜스크립션하려면(AWS CLI)

- 다음 코드를 실행합니다.

```
aws transcribe start-transcription-job \  
--cli-input-json file://filepath/example-start-command.json
```

다음 코드는 example-start-command.json의 내용을 보여줍니다.

```
{  
  "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 의료 및 인터페이스 VPC 엔드포인트(AWS PrivateLink)

인터페이스 VPC 엔드포인트를 생성하여 VPC와 Amazon Transcribe Medical 간에 프라이빗 연결을 설정할 수 있습니다. 인터페이스 엔드포인트는 인터넷 게이트웨이, [AWS PrivateLink](#), NAT 디바이스, VPN 연결 또는 Direct Connect 연결 없이 Medical Amazon Transcribe APIs에 비공개로 액세스할 수 있는 기술로 구동됩니다. VPC의 인스턴스는 Amazon Transcribe Medical APIs와 통신하는 데 퍼블릭 IP 주

소가 필요하지 않습니다. VPC와 Amazon Transcribe Medical 간의 트래픽은 Amazon 네트워크를 벗어나지 않습니다.

각 인터페이스 엔드포인트는 서브넷에서 하나 이상의 [Elastic Network Interfaces](#)로 표현됩니다.

자세한 내용은 Amazon VPC 사용 설명서의 [인터페이스 VPC 엔드포인트\(AWS PrivateLink\)](#)를 참조하세요.

Amazon Transcribe Medical VPC 엔드포인트에 대한 고려 사항

Amazon Transcribe Medical용 인터페이스 VPC 엔드포인트를 설정하기 전에 Amazon VPC 사용 설명서의 [인터페이스 엔드포인트 속성 및 제한 사항](#)을 검토해야 합니다.

Amazon Transcribe Medical은 VPC에서 모든 API 작업을 호출할 수 있도록 지원합니다.

Amazon Transcribe Medical용 인터페이스 VPC 엔드포인트 생성

AWS Management Console 또는를 사용하여 Amazon Transcribe Medical 서비스에 대한 VPC 엔드포인트를 생성할 수 있습니다 AWS CLI. 자세한 내용은 Amazon VPC 사용 설명서의 [인터페이스 엔드포인트 생성](#)을 참조하세요.

Amazon Transcribe Medical의 배치 트랜스크립션의 경우 다음 서비스 이름을 사용하여 VPC 엔드포인트를 생성합니다.

- `com.amazonaws.us-west-2.transcribe`

Amazon Transcribe Medical에서 스트리밍 트랜스크립션의 경우 다음 서비스 이름을 사용하여 VPC 엔드포인트를 생성합니다.

- `com.amazonaws.us-west-2.transcribestreaming`

엔드포인트에 대해 프라이빗 DNS를 활성화하는 경우와 AWS 리전같은의 기본 DNS 이름을 사용하여 Amazon Transcribe Medical에 API 요청을 할 수 있습니다 `transcribestreaming.us-east-2.amazonaws.com`.

자세한 내용은 Amazon VPC 사용 설명서의 [인터페이스 엔드포인트를 통해 서비스 액세스](#)를 참조하세요.

Amazon Transcribe Medical 스트리밍을 위한 VPC 엔드포인트 정책 생성

Amazon Transcribe Medical에 대한 액세스를 제어하는 VPC 엔드포인트에 엔드포인트 정책을 연결할 수 있습니다. 이 정책은 다음 정보를 지정합니다.

- 작업을 수행할 수 있는 위탁자.
- 수행할 수 있는 작업.
- 작업을 수행할 수 있는 리소스.

자세한 내용은 Amazon VPC 사용 설명서의 [VPC 엔드포인트를 통해 서비스에 대한 액세스 제어](#)를 참조하세요.

예: Amazon Transcribe Medical 스트리밍 트랜스크립션 작업에 대한 VPC 엔드포인트 정책

다음은 Amazon Transcribe Medical에서 스트리밍 트랜스크립션을 위한 엔드포인트 정책의 예입니다. 엔드포인트에 연결되면 이 정책은 모든 리소스의 모든 보안 주체에 대해 나열된 Amazon Transcribe Medical 작업에 대한 액세스 권한을 부여합니다.

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Allow",
      "Action": [
        "transcribe:StartMedicalStreamTranscription",
      ],
      "Resource": "*"
    }
  ]
}
```

예: Amazon Transcribe Medical 배치 트랜스크립션 작업에 대한 VPC 엔드포인트 정책

다음은 Amazon Transcribe Medical에서 배치 트랜스크립션을 위한 엔드포인트 정책의 예입니다. 엔드포인트에 연결되면 이 정책은 모든 리소스의 모든 보안 주체에 대해 나열된 Amazon Transcribe Medical 작업에 대한 액세스 권한을 부여합니다.

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

```
    "Principal": "*",
    "Effect": "Allow",
    "Action": [
        "transcribe:StartMedicalTranscriptionJob"
    ],
    "Resource": "*"
  }
]
```

공유 서브넷

공유하는 서브넷의 VPC 엔드포인트는 생성, 설명, 수정 또는 삭제할 수 없습니다. 그러나 공유하는 서브넷의 VPC 엔드포인트를 사용할 수는 있습니다. VPC 공유에 대한 자세한 내용은 Amazon Virtual Private Cloud 가이드의 [다른 계정과 VPC 공유를 참조하세요](#).

AWS HealthScribe

AWS HealthScribe는 음성 인식과 생성형 AI를 결합하여 환자-클리니션 대화를 트랜스크립션하고 easy-to-review 임상 노트를 생성하는 HIPAA 적격 기계 학습(ML) 기능입니다. AWS HealthScribe는 의료 소프트웨어 공급업체가 문서 부담을 줄이고 상담 경험을 개선하는 임상 애플리케이션을 구축할 수 있도록 지원합니다. 이 서비스는 풍부한 대화 기록을 자동으로 제공하고, 화자 역할을 식별하며, 대화를 분류하고, 의학 용어를 추출하고, 예비 임상 기록을 생성합니다. AWS HealthScribe는 이러한 기능을 결합하여 별도의 AI 서비스를 통합하고 최적화할 필요가 없으므로 구현을 가속화할 수 있습니다.

일반 사용 사례:

- 문서화 시간 단축 — 의료진이 애플리케이션에서 쉽게 검토, 조정 및 마무리할 수 있는 AI 생성형 임상 노트를 사용하여 임상 문서를 신속하게 작성할 수 있습니다.
- 의료 필사자의 효율성 향상 — 의료 필사자에게 AI 생성형 트랜스크립트와 임상 노트를 상담 오디오와 함께 제공하여 문서 처리 시간을 단축할 수 있습니다.
- 효율적인 환자 방문 요약 — 사용자가 애플리케이션에서 대화의 주요 내용을 빠르게 기억할 수 있습니다.

Important

AWS HealthScribe에서 생성된 결과는 확률적이며 다양한 요인으로 인해 항상 정확하지는 않을 수 있습니다. 오디오 품질, 배경 노이즈, 화자 명확성, 의학 용어의 복잡성, 컨텍스트별 언어 뉘앙스, 및 [기계 학습 및 생성형 AI의 특성](#). AWS HealthScribe는 임상 의사와 의료 담당자를 위한 보조 역할에 사용하도록 설계되었습니다. AWS HealthScribe 출력은 환자 치료 시나리오에서만 사용해야 합니다. 다음을 포함합니다. 전자 건강 기록의 일부로 훈련된 의료 전문가의 올바른 의학적 판단의 정확성 및 적용 검토 후. AWS HealthScribe 출력은 전문적인 의학적 조언을 대체하지 않습니다. 진단, 또는 처리 및는 완치용이 아니며, 처리, 완화, 방지, 또는 질병 또는 건강 상태를 진단합니다.

주제

- [보안](#)
- [서비스 가용성](#)
- [기술 요구 사항](#)
- [지원되는 의료 전문 분야](#)

- [워크플로](#)
- [AWS HealthScribe Transcript 파일](#)
- [AWS HealthScribe 임상 설명서 파일](#)
- [AWS HealthScribe 트랜스크립션 작업](#)
- [AWS HealthScribe 스트리밍](#)
- [AWS HealthScribe의 유휴 데이터 암호화](#)

보안

AWS HealthScribe는 공동 책임 모델에 따라 운영되며, 여기서 AWS 는 AWS HealthScribe를 실행하는 인프라를 보호할 책임이 있고 사용자는 데이터를 관리할 책임이 있습니다. 자세한 내용은 [공동 책임 모델](#)을 참조하세요.

기본적으로 AWS HealthScribe는 Amazon S3관리형 키를 사용하여 민감한 고객 데이터를 보호하기 위해 저장 시 암호화를 제공합니다. AWS HealthScribe 트랜스크립션 작업을 생성하거나 스트리밍을 시작할 때 고객 관리형 키를 지정할 수 있습니다. 이렇게 하면 두 번째 암호화 계층이 추가됩니다. 자세한 내용은 [AWS HealthScribe의 유휴 데이터 암호화](#) 단원을 참조하십시오.

서비스 가용성

AWS HealthScribe는 미국 동부(버지니아 북부) 리전에서 사용할 수 있습니다.

기술 요구 사항

- 지원되는 언어: 미국 영어(en-US)
- 권장 오디오 형식: 무손실 오디오(예: FLAC 또는 WAV)
- 인코딩: PCM 16비트
- 샘플 속도: 16,000Hz 이상

지원되는 의료 전문 분야

AWS HealthScribe는 현재 다음과 같은 전문 분야를 지원합니다.

- 알레르기 면역학
- 심장학

- 피부과학
- 내분비학
- 위장병학
- 혈액학/종양학
- 감염 질병
- 신장병학
- 신경학
- 산부인과
- 종양학
- 안과학
- 정형외과
- 이비인후과
- 통증 의학
- 소아과학
- 1차 의료
- 정신의학
- 호흡기학
- 류마티스학
- 외과
- 비뇨기과

워크플로

AWS HealthScribe 워크플로에는 트랜스크립션 작업 및 스트리밍이 포함됩니다. 트랜스크립션 작업을 실행하거나 스트림을 완료하면 AWS HealthScribe는 대화 turn-by-turn마다 트랜스크립션 출력과 인사이트가 포함된 트랜스크립션 파일을 생성합니다. 또한 요약 및 증거 링크가 포함된 임상 문서 파일을 생성합니다. 자세한 내용은 [AWS HealthScribe Transcript 파일](#) 및 [AWS HealthScribe 임상 설명서 파일](#) 섹션을 참조하세요.

- 트랜스크립션 작업 - 트랜스크립션 작업을 통해 AWS HealthScribe는 Amazon S3 버킷에서 완료된 의료 상담 미디어 파일을 분석합니다. 다음은 AWS HealthScribe 트랜스크립션 작업과 관련된 API 작업입니다.

- [StartMedicalScribeJob](#)
- [ListMedicalScribeJobs](#)
- [GetMedicalScribeJob](#)
- [DeleteMedicalScribeJob](#)

예제 코드를 포함한 자세한 내용은 [AWS HealthScribe 트랜스크립션 작업](#) 섹션을 참조하세요.

- 스트리밍 - AWS HealthScribe 스트리밍은 한 채널에서 오디오 스트림을 수락하고 다른 채널에서 오디오 트랜스크립션을 제공하는 실시간 HTTP2 기반 양방향 서비스입니다.

다음은 AWS HealthScribe 스트리밍과 관련된 API 작업입니다.

- [StartMedicalScribeStream](#)
- [GetMedicalScribeStream](#)

예제 코드를 포함한 자세한 내용은 [AWS HealthScribe 스트리밍](#) 섹션을 참조하세요.

AWS HealthScribe Transcript 파일

트랜스크립트 파일에서 단어 수준 타임스탬프가 있는 표준 turn-by-turn 트랜스크립션 출력 외에도 AWS HealthScribe는 다음을 제공합니다.

- 참가자 역할 감지를 통해 대화 트랜스크립트에 있는 임상가와 환자를 구분할 수 있습니다.
- 트랜스크립트 분할은 잡담, 주관적 정보, 객관적 정보 등 임상적 관련성을 기준으로 트랜스크립트 대화를 분류합니다. 이를 사용하여 트랜스크립트의 특정 부분을 표시할 수 있습니다.
- 임상 엔터티: 대화에서 언급한 약물, 질병 및 치료법과 같은 구조화된 정보가 포함됩니다.

또한 각 대화 턴에 대해 다음과 같은 인사이트가 제공됩니다.

- 참가자 역할 — 각 참가자는 임상가 또는 환자로 분류됩니다. 대화에 범주별로 한 명 이상의 참가자가 있는 경우 각 참가자에게 번호가 할당됩니다. 예를 들면 CLINICIAN_0, CLINICIAN_1 및 PATIENT_0, PATIENT_1입니다.
- 섹션 - 식별된 콘텐츠에 따라 가능한 네 개의 섹션 중 하나에 각 대화 턴이 할당됩니다.
 - 주관적 - 환자의 건강 문제에 대해 환자가 제공한 정보입니다.
 - 객관적 - 신체 검사, 랩, 영상 또는 진단 검사를 통해 의사가 관찰한 정보입니다.
 - 평가 및 계획 - 의사의 평가 및 치료 계획과 관련된 정보입니다.

- 방문 흐름 관리 - 잡담 또는 전환과 관련된 정보입니다.
- 인사이트 - 대화에 있는 임상 관련 엔터티(ClinicalEntity)를 추출합니다. AWS HealthScribe는 [Amazon Comprehend Medical](#)에서 지원하는 모든 임상 엔터티를 감지합니다.

트랜스크립션 작업의 트랜스크립트 예제는 [트랜스크립션 작업 출력 예제](#)의 트랜스크립트 출력을 참조하세요. 스트리밍의 트랜스크립트 예제는 [스트리밍 트랜스크립션 출력 예제](#)의 트랜스크립트 출력을 참조하세요.

AWS HealthScribe 임상 설명서 파일

AWS HealthScribe는 임상 노트 요약에 다음 템플릿 중 하나를 사용할 수 있습니다. 기본값은 HISTORY_AND_PHYSICAL입니다.

- HISTORY_AND_PHYSICAL: 임상 문서의 주요 섹션에 대한 요약을 제공합니다. 섹션의 예로는 주요 불만 제기, 현재 질병 기록, 시스템 검토, 과거 의학적 기록, 평가 및 계획이 있습니다.
- GIRPP: 목표를 향한 환자의 진행 상황을 기반으로 요약을 제공합니다. 섹션의 예로는 목표, 개입, 대응, 진행 상황 및 계획이 있습니다.
- BIRP: 환자의 행동 패턴과 응답에 중점을 둡니다. 섹션의 예로는 행동, 개입, 대응 및 계획이 있습니다.
- SIRP: 상황별 치료 컨텍스트를 강조합니다. 섹션의 예로는 상황, 개입, 대응 및 계획이 있습니다.
- DAP: 임상 문서에 대한 간소화된 형식을 제공합니다. 섹션의 예로는 데이터, 평가 및 계획이 있습니다.
- BH_SOAP: 동작 상태 중심 설명서 형식입니다. 섹션의 예로는 주관적, 목표, 평가 및 계획이 있습니다.
- PH_SOAP: 물리적 상태 중심의 설명서 형식입니다. 섹션의 예로는 주관적, 목표, 평가 및 계획이 있습니다.

사용할 템플릿을 지정하려면 다음을 수행합니다.

- 트랜스크립션 작업의 경우 [StartMedicalScribeJob](#) API 작업의 Settings에 있는 [ClinicalNoteGenerationSettings](#)의 NoteTemplate에 사용할 템플릿을 지정합니다.
- 스트리밍의 경우 [MedicalScribeConfigurationEvent](#)의 PostStreamAnalyticsSettings에 있는 [ClinicalNoteGenerationSettings](#)의 NoteTemplate에 사용할 템플릿을 지정합니다.

Medical Scribe 컨텍스트

MedicalScribeContext 객체에는 환자별 세부 정보로 임상 노트 생성을 사용자 지정하는 데 사용되는 컨텍스트 정보가 포함되어 있습니다.

MedicalScribeContext 객체에는 다음 구성 요소가 포함되어 있습니다.

- PatientContext - 임상 노트 생성을 사용자 지정하는 데 사용되는 환자별 정보를 포함합니다. 여기에는 다음이 포함됩니다.
 - 대명사 - 임상 노트 생성을 위한 컨텍스트로 제공하려는 환자가 선호하는 대명사입니다. 이러한 대명사는 생성된 임상 출력에서 환자를 참조할 때 사용됩니다.

Medical scribe 컨텍스트를 설정하는 방법에 대한 자세한 내용은 [MedicalScribeContext](#)를 참조하세요.

주제

- [HISTORY_AND_PHYSICAL 템플릿 섹션](#)
- [GIRPP 템플릿 섹션](#)
- [BIRP 템플릿 섹션](#)
- [SIRP 템플릿 섹션](#)
- [DAP 템플릿 섹션](#)
- [BH_SOAP 템플릿 섹션](#)
- [PH_SOAP 템플릿 섹션](#)

HISTORY_AND_PHYSICAL 템플릿 섹션

HISTORY_AND_PHYSICAL 인사이트 템플릿에는 다음 섹션이 포함되어 있습니다.

섹션	설명
주요 불편 사항	환자가 임상 의에게 방문한 이유에 대한 간략한 설명.
현재 질병의 이력	중증도, 발병 시기, 현재 치료 방법, 감염 부위에 대한 참조를 포함하여 환자의 질병에 대한 정보를 제공하는 노트.

섹션	설명
신체 증상	환자가 보고한 다양한 신체 전반의 증상 평가입니다.
과거 병력	환자의 이전 건강 상태, 수술 및 치료에 대한 자세한 설명입니다.
평가	환자의 건강에 대한 임상사의 평가 정보를 제공하는 노트.
PLAN	모든 의학적 치료, 생활 습관 조정 및 추가 진료 예약을 언급하는 메모입니다.
PHYSICAL_EXAMINATION	환자의 신체 계통 및 생체 징후에 대한 신체 검사의 임상사의 조사 결과 문서화.
PAST_FAMILY_HISTORY	환자 가족에서 실행되는 건강 상태에 대한 정보입니다.
과거_사회_이력	환자의 사회생활, 습관, 직업 및 건강에 영향을 미치는 환경 요인에 대한 세부 정보입니다.
진단_테스트	실험실 검사, 영상 검사 및 기타 진단 절차의 결과 및 해석.

Summary의 모든 문장에는 요약된 트랜스크립트의 관련 대화 내용에 대한 SegmentId를 제공하는 EvidenceLinks가 함께 제공됩니다. 이를 통해 사용자는 애플리케이션에서 요약의 정확성을 검증할 수 있습니다. 설명 가능성과 같이 AI 생성형 인사이트에 추적성과 투명성을 제공하는 것은 책임있는 AI 원칙과 일치합니다. 이러한 참고 자료를 요약 노트와 함께 임상사나 의료진에게 제공하면 신뢰를 높이고 임상 환경에서 인공지능을 안전하게 사용하도록 장려하는 데 도움이 됩니다.

트랜스크립션 작업의 임상 설명서 파일의 예는 [트랜스크립션 작업 출력 예제](#)의 임상 설명서 파일 예제를 참조하세요. 스트리밍의 임상 설명서 파일의 예는 [스트리밍 트랜스크립션 출력 예제](#)의 임상 설명서 파일 예제를 참조하세요.

GIRPP 템플릿 섹션

GIRPP 인사이트 템플릿에는 다음 섹션이 포함되어 있습니다.

섹션	설명
Goal	처리를 통해 해결해야 하는 식별된 문제, 문제 또는 동작입니다.
개입	환자가 식별된 목표를 해결하는 데 도움이 되도록 임상이가 사용하는 특정 치료, 방법 또는 기법입니다.
응답	참여 수준, 반응 및 피드백을 포함하여 환자가 개입에 반응한 방식입니다.
Progress	환자 개선 또는 장벽에 대한 관찰을 포함하여 치료 목표를 향한 임상이의 움직임 평가.
계획	향후 개입, 숙제 할당 및 추천을 포함한 처리의 다음 단계입니다.

BIRP 템플릿 섹션

BIRP 인사이트 템플릿에는 다음 섹션이 포함되어 있습니다.

섹션	설명
동작	환자가 제기하는 문제와 치료에 대한 환자의 응답입니다.
개입	환자가 식별된 목표를 해결하는 데 도움이 되도록 임상이가 사용하는 특정 치료, 방법 또는 기법입니다.
응답	참여 수준, 반응 및 피드백을 포함하여 환자가 개입에 반응한 방식입니다.
계획	향후 개입, 숙제 할당 및 추천을 포함한 처리의 다음 단계입니다.

SIRP 템플릿 섹션

SIRP 인사이트 템플릿에는 다음 섹션이 포함되어 있습니다.

섹션	설명
상황	환자가 제기하는 문제와 치료 목적.
개입	환자가 식별된 목표를 해결하는 데 도움이 되도록 임상이가 사용하는 특정 치료, 방법 또는 기법입니다.
응답	참여 수준, 반응 및 피드백을 포함하여 환자가 개입에 반응한 방식입니다.
계획	향후 개입, 숙제 할당 및 추천을 포함한 처리의 다음 단계입니다.

DAP 템플릿 섹션

DAP 인사이트 템플릿에는 다음 섹션이 포함되어 있습니다.

섹션	설명
데이터	환자의 치료 요청 이유 및 환자 정보.
평가	환자의 상황에 대한 임상이의 진단.
계획	향후 개입, 숙제 할당 및 추천을 포함한 처리의 다음 단계입니다.

BH_SOAP 템플릿 섹션

BH_SOAP 인사이트 템플릿에는 다음 섹션이 포함되어 있습니다.

섹션	설명
주관적 정보	클라이언트의 처리 목표 및 경험, 기존 및 과거 문제.
목표	클라이언트에 대한 데이터 및 팩트입니다.
평가	환자의 상황에 대한 임상사의 진단.
계획	향후 개입, 숙제 할당 및 추천을 포함한 처리의 다음 단계입니다.

PH_SOAP 템플릿 섹션

PH_SOAP 인사이트 템플릿에는 다음 섹션이 포함되어 있습니다.

섹션	설명
주관적 정보	클라이언트의 처리 목표 및 경험, 기존 및 과거 문제.
목표	클라이언트에 대한 데이터 및 팩트입니다.
평가	환자의 상황에 대한 임상사의 진단.
계획	향후 개입, 숙제 할당 및 추천을 포함한 처리의 다음 단계입니다.

AWS HealthScribe 트랜스크립션 작업

AWS HealthScribe 트랜스크립션 작업은 Amazon S3 버킷의 미디어 파일을 처리합니다. 미디어 파일을 처리할 때 환자-의사 대화를 트랜스크립션하고 의료 상담을 분석하여 [트랜스크립트](#) 파일과 [임상 문서](#) 파일이라는 두 개의 JSON 출력 파일을 생성합니다.

다음은 AWS HealthScribe 트랜스크립션 작업과 관련된 API 작업입니다.

- [StartMedicalScribeJob](#)

- [ListMedicalScribeJobs](#)
- [GetMedicalScribeJob](#)
- [DeleteMedicalScribeJob](#)

AWS HealthScribe 트랜스크립션 작업 시작

AWS CLI 또는 AWS SDK를 사용하여 AWS HealthScribe 작업을 시작할 수 있습니다.

AWS CLI

이 예시에서는 [start-medical-scribe-job](#) 명령을 사용합니다. 자세한 내용은 [StartMedicalScribeJob](#)을 참조하세요.

```
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
```

다음은 [start-medical-scribe-job](#) 명령을 사용하는 또 다른 예 및 추가 설정이 포함된 요청 본문입니다.

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

my-first-medical-scribe-job.json 파일에는 다음과 같은 요청 본문이 포함되어 있습니다.

```
{  
  "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 for Python (Boto3)

다음 예시에서는 AWS SDK for Python (Boto3)를 사용하여 [start_medical_scribe_job](#) 요청을 수행합니다. 자세한 내용은 [StartMedicalScribeJob](#)을 참조하세요.

```

from __future__ import print_function
import time
import boto3

transcribe = boto3.client('transcribe', 'us-west-2')
job_name = "my-first-medical-scribe-job"
job_uri = "s3://amzn-s3-demo-bucket/my-input-files/my-media-file.flac"
transcribe.start_medical_scribe_job(
    MedicalScribeJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'amzn-s3-demo-bucket',
    DataAccessRoleArn = 'arn:aws:iam::111122223333:role/ExampleRole',
    Settings = {
        'ShowSpeakerLabels': false,
        'ChannelIdentification': true
    },
    ChannelDefinitions = [
        {
            'ChannelId': 0,
            'ParticipantRole': 'CLINICIAN'
        }
    ]
)

```

```

    }, {
        'ChannelId': 1,
        'ParticipantRole': 'PATIENT'
    }
]
)
while True:
    status = transcribe.get_medical_scribe_job(MedicalScribeJobName = job_name)
    if status['MedicalScribeJob']['MedicalScribeJobStatus'] in ['COMPLETED', 'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
print(status)

```

Note

AWS 관리 콘솔은 현재 AWS HealthScribe 작업을 지원하지 않습니다.

트랜스크립션 작업 출력 예제

StartMedicalScribeJob 요청은 트랜스크립트 외에도 별도의 임상 문서 파일을 생성합니다. 두 파일 모두 JSON 형식이며 사용자가 요청에서 지정한 출력 위치에 저장됩니다. 다음은 각 출력 유형 예시입니다.

트랜스크립트 출력 예시

StartMedicalScribeJob 요청에서 비롯된 AWS HealthScribe 트랜스크립트 파일의 형식은 다음과 같습니다.

```

{
  "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"
  }
]
}
}

```

다음은 [start-medical-scribe-job](#) 명령을 사용하는 또 다른 예 및 추가 설정이 포함된 요청 본문입니다.

```

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

```

my-first-medical-scribe-job.json 파일에는 다음과 같은 요청 본문이 포함되어 있습니다.

```

{
  "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"
    }
  ]
}

```

임상 문서 출력 예시

StartMedicalScribeJob 요청에서 비릇된 문서 인사이트 파일의 형식은 다음과 같습니다.

```

{
  "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 스트리밍

AWS HealthScribe 스트리밍을 사용하면 의료 대화를 실시간으로 트랜스크립션할 수 있습니다. AWS HealthScribe 스트리밍은 한 채널에서 오디오 스트림을 수락하고 다른 채널에서 오디오 트랜스크립션을 제공하는 실시간 HTTP2 기반 양방향 서비스입니다. 스트리밍이 완료되면 AWS HealthScribe는 스트림 콘텐츠를 분석하고 트랜스크립트 JSON 파일과 임상 노트 JSON 파일을 생성합니다.

스트리밍을 시작하려면 [StartMedicalScribeStream](#) API 작업을 사용합니다. 이 API는 오디오 이벤트를 스트리밍하는 데 사용하는 HTTP2 기반 양방향 채널을 시작합니다.

스트림을 시작할 때 먼저에서 스트림 구성을 지정합니다 `MedicalScribeConfigurationEvent`. 이 이벤트에는 집계된 트랜스크립트 및 임상 노트 생성을 위한 출력 구성과 같은 채널 정의, 암호화 설정 및 스트림 후 분석 설정이 포함됩니다.

오디오 스트리밍을 시작한 후 다음과 같이 스트림을 관리합니다.

- 완료되면 스트림 후 분석으로 결과 처리를 시작하려면 `Type`의 `MedicalScribeSessionControlEvent`로 전송 `END_OF_SESSION`하고 AWS HealthScribe는 분석을 시작합니다.
- 스트리밍을 일시 중지하려면 `MedicalScribeSessionControlEvent`를 전송하지 않고 입력 스트림을 완료합니다.
- 일시 중지된 스트림을 재개하려면 `StartMedicalScribeStream` API 작업을 사용하고 동일한 `SessionId`를 지정합니다. 스트림을 처음 시작할 때 사용한 `SessionId`입니다.

주제

- [지침 및 요구 사항](#)
- [ResourceAccessRoleArn 역할 권한](#)
- [Starting AWS HealthScribe 스트리밍 트랜스크립션](#)

지침 및 요구 사항

다음은 AWS HealthScribe 스트리밍에 대한 지침 및 요구 사항입니다.

- 오디오 이벤트를 전송하기 전에 먼저 [MedicalScribeConfigurationEvent](#)에서 스트림 구성을 지정해야 합니다.

- 스트림 후 분석을 실행하려면 `MedicalScribeConfigurationEvent`의 `ResourceAccessRoleArn`에 올바른 권한이 있어야 합니다. 자세한 내용은 [ResourceAccessRoleArn 역할 권한](#) 단원을 참조하십시오.
- 초기 스트림 생성 후 5시간 이내에 원하는 횟수만큼 세션을 재개할 수 있습니다.
- 모든 스트리밍 요청에서 세션을 통해 최대 2시간의 오디오를 스트리밍할 수 있습니다.
- 기본적으로 AWS HealthScribe는 Amazon S3관리형 키를 사용하여 민감한 고객 데이터를 보호하기 위해 저장 시 암호화를 제공합니다. 스트림을 시작할 때 두 번째 암호화 계층에 대한 AWS KMS 키를 지정할 수 있습니다. 예는 AWS KMS 키를 사용할 수 있는 권한이 `ResourceAccessRoleArn` 있어야 합니다. 자세한 내용은 [AWS HealthScribe의 유휴 데이터 암호화](#) 단원을 참조하십시오.
- SDK for Python(Boto3) 및 SDK for PHP를 AWS SDKs 제외하고 SDK와 함께 AWS HealthScribe 스트리밍을 사용할 수 있습니다.
- 스트림을 종료한 후에도 `LimitExceededException` 예외가 발생하면 세션을 다시 시작하고 스트림 후 분석을 생성할 수 있습니다. 스트림을 다시 시작하려면 [StartMedicalScribeStream](#) API를 사용하고 동일한 `SessionID`를 사용합니다. 그런 다음 `Type END_OF_SESSION` 및 AWS HealthScribe의 `MedicalScribeSessionControlEvent`를 사용하여 전송하면 분석이 시작됩니다.

ResourceAccessRoleArn 역할 권한

스트림 후 분석을 실행하려면 `ResourceAccessRoleArn`의가 Amazon S3 출력 버킷에 액세스할 수 있어야 `MedicalScribeConfigurationEvent` 하며, 제공된 경우 AWS KMS 키에 액세스할 수 있어야 합니다. 또한, 역할을 맡을 `transcribe.streaming.amazonaws.com` 서비스 권한은 역할의 신뢰 정책에서 부여합니다.

다음은 Amazon S3 버킷 권한 및 AWS KMS 키 권한을 부여하는 IAM 정책의 예입니다. 자세한 내용은 [AWS HealthScribe의 유휴 데이터 암호화](#) 단원을 참조하십시오.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket",
```

```

        "arn:aws:s3:::amzn-s3-demo-bucket/*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "kms:DescribeKey",
        "kms:Decrypt",
        "kms:Encrypt",
        "kms:GenerateDataKey*"
      ],
      "Resource": "arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
      "Effect": "Allow"
    }
  ]
}

```

다음은 신뢰 정책 예시입니다.

JSON

```

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

```

Starting AWS HealthScribe 스트리밍 트랜스크립션

다음 코드 예제에서는 AWS SDKs를 사용하여 AWS HealthScribe 스트리밍 트랜스크립션을 설정하는 방법을 보여줍니다.

주제

- [SDK for Java 2.x](#)
- [스트리밍 트랜스크립션 출력 예제](#)

SDK for Java 2.x

다음 예제에서는 Java 2.x용 SDK를 사용하여 스트리밍을 설정하고 [StartMedicalScribeStream](#) 요청을 수행합니다.

```
package org.example;

import io.reactivex.rxjava3.core.BackpressureStrategy;
import io.reactivex.rxjava3.core.Flowable;
import org.reactivestreams.Publisher;
import org.reactivestreams.Subscriber;
import software.amazon.awssdk.auth.credentials.AwsCredentialsProvider;
import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
import software.amazon.awssdk.core.SdkBytes;
import software.amazon.awssdk.regions.Region;
import
    software.amazon.awssdk.services.transcribestreaming.TranscribeStreamingAsyncClient;
import
    software.amazon.awssdk.services.transcribestreaming.model.ClinicalNoteGenerationSettings;
import software.amazon.awssdk.services.transcribestreaming.model.LanguageCode;
import software.amazon.awssdk.services.transcribestreaming.model.MediaEncoding;

import
    software.amazon.awssdk.services.transcribestreaming.model.MedicalScribeInputStream;
import
    software.amazon.awssdk.services.transcribestreaming.model.MedicalScribePostStreamAnalyticsSettings;
import
    software.amazon.awssdk.services.transcribestreaming.model.MedicalScribeSessionControlEventType;
import
    software.amazon.awssdk.services.transcribestreaming.model.MedicalScribeTranscriptEvent;
import
    software.amazon.awssdk.services.transcribestreaming.model.MedicalScribeTranscriptSegment;
```



```

import
    software.amazon.awssdk.services.transcribestreaming.model.StartMedicalScribeStreamRequest;
import
    software.amazon.awssdk.services.transcribestreaming.model.StartMedicalScribeStreamResponseHandler;
import
    software.amazon.awssdk.services.transcribestreaming.model.medicalscribeinputstream.DefaultConfiguration;

import software.amazon.awssdk.http.nio.netty.NettyNioAsyncHttpClient;

import javax.sound.sampled.AudioFormat;
import javax.sound.sampled.AudioInputStream;
import javax.sound.sampled.AudioSystem;
import javax.sound.sampled.DataLine;
import javax.sound.sampled.LineUnavailableException;
import javax.sound.sampled.TargetDataLine;
import java.io.BufferedInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.UncheckedIOException;
import java.util.Arrays;
import java.util.concurrent.CompletableFuture;

public class HealthScribeStreamingDemoApp {
    private static final int CHUNK_SIZE_IN_BYTES = 6400;
    private static final int SAMPLE_RATE = 16000;
    private static final Region REGION = Region.US_EAST_1;
    private static final String sessionId = "1234abcd-12ab-34cd-56ef-123456SAMPLE";
    private static final String bucketName = "amzn-s3-demo-bucket";
    private static final String resourceAccessRoleArn =
        "arn:aws:iam::123456789012:role/resource-access-role";
    private static TranscribeStreamingAsyncClient client;

    public static void main(String args[]) {

        client = TranscribeStreamingAsyncClient.builder()
            .credentialsProvider(getCredentials())
            .httpClientBuilder(NettyNioAsyncHttpClient.builder())
            .region(REGION)
            .build();

        try {
            StartMedicalScribeStreamRequest request =
                StartMedicalScribeStreamRequest.builder()
                    .languageCode(LanguageCode.EN_US.toString())

```

```

        .mediaSampleRateHertz(SAMPLE_RATE)
        .mediaEncoding(MediaEncoding.PCM.toString())
        .sessionId(sessionId)
        .build();

    MedicalScribeInputStream endSessionEvent =
MedicalScribeInputStream.sessionControlEventBuilder()
        .type(MedicalScribeSessionControlEventType.END_OF_SESSION)
        .build();

    CompletableFuture<Void> result = client.startMedicalScribeStream(
        request,
        new AudioStreamPublisher(getStreamFromMic(),
getConfigurationEvent(),endSessionEvent),
        getMedicalScribeResponseHandler());
    result.get();
    client.close();
} catch (Exception e) {
    System.err.println("Error occurred: " + e.getMessage());
    e.printStackTrace();
}
}

private static AudioInputStream getStreamFromMic() throws LineUnavailableException
{
    // Signed PCM AudioFormat with 16kHz, 16 bit sample size, mono
    AudioFormat format = new AudioFormat(SAMPLE_RATE, 16, 1, true, false);
    DataLine.Info info = new DataLine.Info(TargetDataLine.class, format);

    if (!AudioSystem.isLineSupported(info)) {
        System.out.println("Line not supported");
        throw new LineUnavailableException("The audio system microphone line is not
supported.");
    }
    TargetDataLine line = (TargetDataLine) AudioSystem.getLine(info);
    int bufferSize = (CHUNK_SIZE_IN_BYTES / format.getFrameSize()) *
format.getFrameSize();
    line.open(format);
    line.start();

    // Create a wrapper class that can be closed when Enter is pressed
    AudioInputStream audioStream = new AudioInputStream(line);

    // Start a thread to monitor for Enter key

```

```

        System.out.println("Recording... Press Enter to stop");
        Thread monitorThread = new Thread(() -> {
            try {
                System.in.read();
                line.stop();
                line.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        });
        monitorThread.setDaemon(true); // Set as daemon thread so it doesn't prevent
JVM shutdown
        monitorThread.start();

        return new AudioInputStream(
            new BufferedInputStream(new AudioInputStream(line)),
            format,
            AudioSystem.NOT_SPECIFIED
        );
    }

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

    private static StartMedicalScribeStreamResponseHandler
getMedicalScribeResponseHandler() {

        return StartMedicalScribeStreamResponseHandler.builder()
            .onResponse(r -> {
                System.out.println("Received Initial response");
            })
            .onError(Throwable::printStackTrace)
            .onComplete(() -> {
                System.out.println("=== All records streamed successfully ===");
            })
            .subscriber(event -> {
                if (event instanceof MedicalScribeTranscriptEvent) {
                    MedicalScribeTranscriptSegment segment =
((MedicalScribeTranscriptEvent) event).transcriptSegment();
                    if (segment != null && segment.content() != null && !
segment.content().isEmpty()) {
                        System.out.println(segment.content());
                    }
                }
            })
    }

```

```

        }
    })
    .build();
}

private static DefaultConfigurationEvent getConfigurationEvent() {
    MedicalScribePostStreamAnalyticsSettings postStreamSettings =
MedicalScribePostStreamAnalyticsSettings
    .builder()
    .clinicalNoteGenerationSettings(
        ClinicalNoteGenerationSettings.builder()
            .outputBucketName(bucketName)
            .build()
        )
    .build();
    return (DefaultConfigurationEvent)
MedicalScribeInputStream.configurationEventBuilder()
    .resourceAccessRoleArn(resourceAccessRoleArn)
    .postStreamAnalyticsSettings(postStreamSettings)
    .build();
}

private static class AudioStreamPublisher implements
Publisher<MedicalScribeInputStream> {
    private final InputStream audioInputStream;
    private final MedicalScribeInputStream configEvent;
    private final MedicalScribeInputStream endSessionEvent;

    private AudioStreamPublisher(AudioInputStream audioInputStream,
                                MedicalScribeInputStream configEvent,
                                MedicalScribeInputStream endSessionEvent) {
        this.audioInputStream = audioInputStream;
        this.configEvent = configEvent;
        this.endSessionEvent = endSessionEvent;
    }

    @Override
    public void subscribe(Subscriber<? super MedicalScribeInputStream> subscriber)
    {
        createAudioFlowable()
            .doOnComplete(() -> {
                try {
                    audioInputStream.close();
                } catch (IOException e) {

```

```

        throw new UncheckedIOException(e);
    }
    })
    .subscribe(subscriber);
}

private Flowable<MedicalScribeInputStream> createAudioFlowable() {
    // Start with config event
    Flowable<MedicalScribeInputStream> configFlow = Flowable.just(configEvent);

    // Create audio chunk flowable
    Flowable<MedicalScribeInputStream> audioFlow = Flowable.create(emitter -> {
        byte[] buffer = new byte[CHUNK_SIZE_IN_BYTES];
        int bytesRead;

        try {
            while (!emitter.isCancelled() && (bytesRead =
audioInputStream.read(buffer)) > 0) {
                byte[] audioData = bytesRead < buffer.length
                    ? Arrays.copyOfRange(buffer, 0, bytesRead)
                    : buffer;

                MedicalScribeInputStream audioEvent =
MedicalScribeInputStream.audioEventBuilder()
                    .audioChunk(SdkBytes.fromByteArray(audioData))
                    .build();

                emitter.onNext(audioEvent);
            }
            emitter.onComplete();
        } catch (IOException e) {
            emitter.onError(e);
        }
    }, BackpressureStrategy.BUFFER);

    // End with session end event
    Flowable<MedicalScribeInputStream> endFlow =
Flowable.just(endSessionEvent);

    // Concatenate all flows
    return Flowable.concat(configFlow, audioFlow, endFlow);
}
}

```

}

스트리밍 트랜스크립션 출력 예제

스트리밍이 완료되면 AWS HealthScribe는 스트림 콘텐츠를 분석하고 트랜스크립트 JSON 파일과 임상 노트 JSON 파일을 생성합니다. 다음은 각 출력 유형 예시입니다.

트랜스크립트 출력 예시

다음은 스트리밍 세션의 AWS HealthScribe 트랜스크립트 파일의 예입니다.

```
{
  "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"
  }
]
}

```

임상 문서 출력 예시

다음은 스트리밍 세션의 AWS HealthScribe 임상 설명서 인사이트 파일의 예입니다.

```

{
  "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"
}
}

```

AWS HealthScribe의 유휴 데이터 암호화

기본적으로 AWS HealthScribe는 AWS HealthScribe 관리형 AWS Key Management Service (AWS KMS) 키를 사용하여 민감한 고객 데이터를 보호하기 위해 저장 시 암호화를 제공합니다. 기본적으로 저장된 데이터를 암호화하면 민감한 데이터를 보호하는 데 수반되는 운영 오버헤드와 복잡성을 줄이는 데 도움이 됩니다. 동시에 엄격한 암호화 규정 준수 및 규제 요구 사항을 충족하는 안전한 애플리케이션을 구축할 수 있습니다. AWS HealthScribe 전사 작업을 생성하거나 스트림을 시작할 때 고객 관리형 키를 지정할 수 있습니다. 이렇게 하면 두 번째 암호화 계층이 추가됩니다.

- **AWS HealthScribe 관리형 AWS KMS 키** - AWS HealthScribe는 기본적으로 AWS HealthScribe 관리형 AWS Key Management Service (AWS KMS) 키를 사용하여 중간 파일을 자동으로 암호화합니다. 이 암호화 계층을 비활성화하거나 대체 암호화 유형을 선택할 수 없습니다. 사용자는 키를 확인, 관리 또는 사용하거나 키 사용을 감사할 수 없습니다. 하지만 데이터를 암호화하는 키를 보호하기 위해 어떤 작업을 수행하거나 어떤 프로그램을 변경할 필요가 없습니다.
- **고객 관리형 키** - AWS HealthScribe는 사용자가 생성, 소유 및 관리하는 대칭 고객 관리형 키를 사용하여 기존 AWS 소유 암호화에 두 번째 암호화 계층을 추가할 수 있도록 지원합니다. 이 암호화 계층을 완전히 제어할 수 있으므로 다음과 같은 작업을 수행할 수 있습니다.
 - 키 정책 수립 및 유지
 - IAM 정책 및 권한 부여 설정 및 유지

- 키 정책 활성화 및 비활성화
- 키 암호화 자료 교체
- 태그 추가
- 키 별칭 만들기
- 삭제를 위한 스케줄 키

자세한 내용은 AWS Key Management Service 개발자 안내서의 [고객 관리형 키를](#) 참조하세요.

Note

AWS HealthScribe는 AWS소유 키를 사용하여 저장 데이터 암호화를 자동으로 활성화하여 개인 식별 데이터를 무료로 보호합니다. 그러나 고객 관리형 키를 사용하는 경우 AWS KMS 요금이 부과됩니다. 요금에 대한 자세한 내용은 [AWS Key Management Service 요금](#)을 참조하세요.

에 대한 자세한 내용은 [란 무엇입니까? AWS Key Management Service](#)를 AWS KMS참조하십시오.

주제

- [고객 관리형 키 생성](#)
- [AWS HealthScribe에 대한 고객 관리형 키 지정](#)
- [AWS KMS 암호화 컨텍스트](#)
- [AWS HealthScribe에 대한 암호화 키 모니터링](#)

고객 관리형 키 생성

AWS Management Console또는 AWS KMS APIs. 대칭 고객 관리형 키를 생성하려면 AWS Key Management Service 개발자 안내서의 [대칭 고객 관리형 키 생성](#) 단계를 따르세요.

키 정책에서는 고객 관리형 키에 대한 액세스를 제어합니다. 모든 고객 관리형 키에는 키를 사용할 수 있는 사람과 키를 사용하는 방법을 결정하는 문장이 포함된 정확히 하나의 키 정책이 있어야 합니다. 고객 관리형 키를 만들 때 키 정책을 지정할 수 있습니다. 자세한 내용은 AWS Key Management Service 개발자 안내서의 [고객 관리형 키에 대한 액세스 관리](#)를 참조하세요.

AWS KMS AWS HealthScribe에 대한 키 정책

StartMedicalScribeJob 또는 DataAccessRole [StartMedicalScribeStream](#) 요청에서 로 지정한 IAM 역할과 동일한 계정ResourceAccessRole에서 키를 사용하는 경우 키 정책을 업데이트할 필요가 없습니다. [StartMedicalScribeJob](#) 다른 계정의 고객 관리형 키를 DataAccessRole(전사 작업의 경우) 또는 ResourceAccessRole(스트리밍의 경우)로 사용하려면 다음 작업에 대해 키 정책의 각 역할을 신뢰해야 합니다.

- [kms:Encrypt](#) - 고객 관리형 키를 사용한 암호화를 허용합니다.
- [kms:Decrypt](#) - 고객 관리형 키를 사용한 복호화를 허용합니다.
- [kms:DescribeKey](#) - AWS HealthScribe가 키를 검증할 수 있도록 고객 관리형 키 세부 정보를 제공합니다.

다음은 ResourceAccessRole 교차 계정에 AWS HealthScribe 스트리밍에 고객 관리형 키를 사용할 수 있는 권한을 부여하는 데 사용할 수 있는 예제 키 정책입니다. 전사 작업에 이 정책을 사용하려면 DataAccessRole ARN을 사용하도록 Principal를 업데이트하고 암호화 컨텍스트를 제거하거나 수정합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAccessForKeyAdministrators",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:root"
      },
      "Action": [
        "kms:*"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowAccessToResourceAccessRoleForMedicalScribe",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/ResourceAccessRole"
      }
    }
  ]
}
```

```

    },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:GenerateDataKey*"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "kms:EncryptionContext:aws:us-east-1:transcribe:medical-
scribe:session-id": "1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    }
},
{
    "Sid": "AllowAccessToResourceAccessRoleForDescribeKey",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/ResourceAccessRole"
    },
    "Action": "kms:DescribeKey",
    "Resource": "*"
}
]
}

```

액세스 역할에 대한 IAM 정책 권한

DataAccessRole 또는 ResourceAccessRole에 연결된 IAM 정책은 고객 관리형 키와 역할이 동일한 계정에 있는지 아니면 다른 계정에 있는지에 관계없이 필요한 AWS KMS 작업을 수행할 수 있는 권한을 부여해야 합니다. 또한 역할의 신뢰 정책은 역할을 수임할 수 있는 AWS HealthScribe 권한을 부여해야 합니다.

다음 IAM 정책 예제에서는 AWS HealthScribe 스트리밍에 대한 ResourceAccessRole 권한을 부여하는 방법을 보여줍니다. 전사 작업에 이 정책을 사용하려면 `transcribe.streaming.amazonaws.com`를 `transcribe.amazonaws.com`로 바꾸고 암호화 컨텍스트를 제거하거나 수정합니다.

JSON

```
{
```

```

    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Action": [
          "kms:Encrypt",
          "kms:Decrypt",
          "kms:GenerateDataKey*"
        ],
        "Resource": "arn:aws:kms:us-west-2:111122223333:key/KMS-Example-  
KeyId",
        "Condition": {
          "StringEquals": {
            "kms:ViaService": "transcribe.streaming.amazonaws.com",
            "kms:EncryptionContext:aws:us-east-1:transcribe:medical-  
scribe:session-id": "1234abcd-12ab-34cd-56ef-123456SAMPLE"
          }
        }
      },
      {
        "Effect": "Allow",
        "Action": [
          "kms:DescribeKey"
        ],
        "Resource": "arn:aws:kms:us-west-2:111122223333:key/KMS-Example-  
KeyId",
        "Condition": {
          "StringEquals": {
            "kms:ViaService": "transcribe.streaming.amazonaws.com"
          }
        }
      }
    ]
  }

```

다음은 ResourceAccessRole에 대한 신뢰 정책의 예입니다. DataAccessRole의 경우 transcribe.streaming.amazonaws.com를 transcribe.amazonaws.com로 바꿉니다.

JSON

```
{
```

```

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

```

[정책에서 권한을 지정](#)하거나 [키 액세스 문제를 해결하는](#) 방법에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서를 참조하세요.

AWS HealthScribe에 대한 고객 관리형 키 지정

고객 관리형 키를 전사 작업 또는 스트리밍을 위한 두 번째 계층 암호화로 지정할 수 있습니다.

- 전사 작업의 경우 [StartMedicalScribeJob](#) API 작업의 [OutputEncryptionKMSKeyId](#)에 키를 지정합니다.
- 스트리밍의 경우 [MedicalScribeConfigurationEvent](#)의 [MedicalScribeEncryptionSettings](#)에 키를 지정합니다.

AWS KMS 암호화 컨텍스트

AWS KMS 암호화 컨텍스트는 일반 텍스트, 비밀이 아닌 키값 페어의 맵입니다. 이 맵은 데이터에 대한 추가 보안 계층을 제공하는 암호화 컨텍스트 페어라고 하는 추가 인증 데이터를 나타냅니다. AWS HealthScribe는 AWS HealthScribe 출력을 고객 지정 Amazon S3 버킷으로 암호화하려면 대칭 암호화 키가 필요합니다. 자세한 내용은 [AWS KMS의 비대칭 키](#)를 참조하세요.

암호화 컨텍스트 페어를 생성할 때 민감한 정보를 포함해서는 안 됩니다. 암호화 컨텍스트는 비밀이 아닙니다. CloudTrail 로그 내에서 일반 텍스트로 볼 수 있습니다(암호화 작업을 식별하고 분류하는 데 사용할 수 있음). 암호화 컨텍스트 페어는 밑줄(_), 대시(-), 슬래시(/, \), 콜론(:) 같은 특수 문자를 포함할 수 있습니다.

Tip

암호화 컨텍스트 페어의 값을 암호화된 데이터와 연관시키는 것이 유용할 수 있습니다. 필수 사항은 아니지만 파일 이름, 헤더 값 또는 암호화되지 않은 데이터베이스 필드 등 암호화된 콘텐츠와 관련된 민감하지 않은 메타데이터를 사용하는 것이 좋습니다.

API와 함께 출력 암호화를 사용하려면 [StartMedicalScribeJob](#) 작업에서 [KMSEncryptionContext](#) 파라미터를 설정하세요. 출력 암호화 작업에 대한 암호화 컨텍스트를 제공하려면 [OutputEncryptionKMSKeyId](#) 파라미터가 대칭 AWS KMS 키 ID를 참조해야 합니다.

스트리밍의 경우 [MedicalScribeConfigurationEvent](#)의 [MedicalScribeEncryptionSettings](#)에서 [KmsEncryptionContext](#)에 대한 키 값 페어를 지정합니다.

IAM 정책과 함께 [AWS KMS 조건 키](#)를 사용하여 암호화 [작업](#) 요청에 사용된 암호화 컨텍스트를 기반으로 대칭 암호화 AWS KMS 키에 대한 액세스를 제어할 수 있습니다. 암호화 컨텍스트 정책의 예는 [AWS KMS 암호화 컨텍스트 정책](#)을 참조하세요.

암호화 컨텍스트는 선택 사항이지만 권장됩니다. 자세한 정보는 [암호화 컨텍스트](#)를 참조하세요.

AWS HealthScribe 암호화 컨텍스트

AWS HealthScribe는 모든 암호화 작업에서 동일한 AWS Key Management Service 암호화 컨텍스트를 사용합니다. 암호화 컨텍스트는 원하는 모든 것에 맞게 사용자 지정할 수 있는 문자열과 문자열의 지도입니다.

```
"encryptionContext": {
  "ECKey": "ECValue"
  ...
}
```

AWS HealthScribe 스트림의 경우 기본 서비스 생성 암호화 컨텍스트는 다음과 같습니다. 이 컨텍스트는 사용자가 제공하는 모든 암호화 컨텍스트 위에 적용됩니다.

```
"encryptionContext": {
```

```
"aws:<region>:transcribe:medical-scribe:session-id":
"1234abcd-12ab-34cd-56ef-123456SAMPLE"
}
```

AWS HealthScribe 트랜스크립션 작업의 경우 기본 서비스 생성 암호화 컨텍스트는 다음과 같습니다. 이 컨텍스트는 사용자가 제공하는 모든 암호화 컨텍스트 위에 적용됩니다.

```
"encryptionContext": {
  "aws:<region>:transcribe:medical-scribe:job-name": "<job-name>",
  "aws:<region>:transcribe:medical-scribe:start-time-epoch-ms": "<job-start-time>"
}
```

암호화 컨텍스트를 제공하지 않으면 서비스에서 생성한 암호화 컨텍스트만 모든 AWS KMS 암호화 작업에 사용됩니다.

암호화 컨텍스트를 사용한 AWS HealthScribe 모니터링

대칭 고객 관리형 키를 사용하여 AWS HealthScribe에서 저장 데이터를 암호화하는 경우 감사 레코드 및 로그의 암호화 컨텍스트를 사용하여 고객 관리형 키가 사용되는 방식을 식별할 수도 있습니다. 암호화 컨텍스트는 AWS CloudTrail 또는 CloudWatch Logs에서 생성된 로그에도 나타납니다.

암호화 컨텍스트를 사용하여 고객 관리형 키에 대한 액세스 제어

그러나 키 정책 및 IAM 정책에서 암호화 컨텍스트를 조건으로 사용하여 대칭형 고객 관리형 키에 대한 액세스를 제어할 수도 있습니다.

다음은 특정 암호화 컨텍스트에서 고객 관리형 키에 대한 액세스 권한을 부여하는 키 정책 설명의 예입니다. 이 정책 설명의 조건에 따라 KMS 키 사용에는 암호화 컨텍스트를 지정하는 암호화 컨텍스트 제약 조건이 있어야 합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAccessToResourceAccessRoleForMedicalScribe",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/ResourceAccessRole"
      },
      "Action": [
```

```

        "kms:Encrypt",
        "kms:Decrypt",
        "kms:GenerateDataKey*"
    ],
    "Resource": "arn:aws:kms:us-west-2:111122223333:key/KMS-Example-  
KeyId",
    "Condition": {
        "StringEquals": {
            "kms:EncryptionContext:aws:us-east-1:transcribe:medical-  
scribe:session-id": "1234abcd-12ab-34cd-56ef-123456SAMPLE",
            "kms:EncryptionContext:ECKey": "ECValue"
        }
    }
},
{
    "Sid": "AllowAccessToResourceAccessRoleForDescribeKey",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/ResourceAccessRole"
    },
    "Action": "kms:DescribeKey",
    "Resource": "arn:aws:kms:us-west-2:111122223333:key/KMS-Example-  
KeyId"
}
]
}

```

AWS HealthScribe에 대한 암호화 키 모니터링

AWS HealthScribe에서 AWS Key Management Service 고객 관리형 키를 사용하는 경우 AWS CloudTrail 또는 CloudWatch 로그를 사용하여 AWS HealthScribe가 보내는 요청을 추적할 수 있습니다 AWS KMS.

다음 예제는 AWS HealthScribe가 고객 관리형 키를 사용하는 방식을 모니터링할 수 있도록 사용할 수 있는 CloudTrail Encrypt 및 Decrypt 이벤트입니다.

암호화

```

{
    "eventVersion": "1.09",
    "userIdentity": {
        "type": "AssumedRole",

```



```

    "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
    "arn": "arn:aws:sts::123456789012:assumed-role/Admin/Sampleuser01",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::123456789012:assumed-role/Admin/Sampleuser01",
        "accountId": "123456789012",
        "userName": "Admin"
      },
      "attributes": {
        "creationDate": "2024-08-16T01:10:05Z",
        "mfaAuthenticated": "false"
      }
    },
    "invokedBy": "transcribe.streaming.amazonaws.com"
  },
  "eventTime": "2024-08-16T01:10:05Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "Encrypt",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "transcribe.streaming.amazonaws.com",
  "userAgent": "transcribe.streaming.amazonaws.com",
  "requestParameters": {
    "encryptionContext": {
      "aws:us-east-1:transcribe:medical-scribe:session-id": "1234abcd-12ab-34cd-56ef-123456SAMPLE"
    },
    "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
    "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "responseElements": null,
  "requestID": "cbe0ac33-8cca-49e5-9bb5-dc2b8dfcb389",
  "eventID": "1b9fedde-aa96-48cc-9dd9-a2cce2964b3c",
  "readOnly": true,
  "resources": [
    {
      "accountId": "123456789012",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-west-2:123456789012:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    }
  ]
}

```

```

],
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "123456789012",
"eventCategory": "Management"
}

```

Decrypt

```

{
  "eventVersion": "1.09",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
    "arn": "arn:aws:sts::123456789012:assumed-role/Admin/Sampleuser01",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::123456789012:assumed-role/Admin/Sampleuser01",
        "accountId": "123456789012",
        "userName": "Admin"
      },
      "attributes": {
        "creationDate": "2024-08-16T20:47:04Z",
        "mfaAuthenticated": "false"
      }
    },
    "invokedBy": "transcribe.streaming.amazonaws.com"
  },
  "eventTime": "2024-08-16T20:47:04Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "Decrypt",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "transcribe.streaming.amazonaws.com",
  "userAgent": "transcribe.streaming.amazonaws.com",
  "requestParameters": {
    "keyId": "mrk-de27f019178f4fbf86512ab03ba860be",
    "encryptionAlgorithm": "SYMMETRIC_DEFAULT",

```

```
    "encryptionContext":{
      "aws:us-east-1:transcribe:medical-scribe:session-
id":"1234abcd-12ab-34cd-56ef-123456SAMPLE"
    }
  },
  "responseElements":null,
  "requestID":"8b7fb865-48be-4e03-ac3d-e7bee3ba30a1",
  "eventID":"68b7a263-d410-4701-9e2b-20c196628966",
  "readOnly":true,
  "resources":[
    {
      "accountId":"123456789012",
      "type":"AWS::KMS::Key",
      "ARN":"arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    }
  ],
  "eventType":"AwsApiCall",
  "managementEvent":true,
  "recipientAccountId":"123456789012",
  "eventCategory":"Management"
}
```

Amazon Transcribe에 대한 문서 기록

- 최신 문서 업데이트: 2023년 11월 13일

다음 테이블에서는 Amazon Transcribe의 각 릴리스에서 변경된 중요 사항에 대해 설명합니다. 이 설명서에 대한 업데이트 알림을 받으려면 RSS 피드를 구독하세요.

변경 사항	설명	날짜
새로운 기능	AWS HealthScribe는 이제 BIRP, SIRP, DAP, BH_SOAP, PH_SOAP 등 임상 노트 요약을 위한 새 템플릿을 지원합니다. 자세한 내용은 AWS HealthScribe 임상 설명서 파일 을 참조하세요.	2025년 5월 30일
새로운 기능	AWS HealthScribe는 이제 임상 노트 요약을 위한 GIRPP 템플릿을 지원합니다. 자세한 내용은 AWS HealthScribe 임상 설명서 파일 을 참조하세요.	2025년 2월 4일
새로운 기능	AWS HealthScribe는 이제 의료 대화를 실시간으로 트랜스크립션하기 위한 스트리밍을 지원합니다.	2025년 1월 29일
섹션 업데이트	8,000회 빈도로 ar-SA 언어 지원 업데이트.	2025년 1월 16일
섹션 업데이트	8,000회 빈도로 ar-SA 언어 지원 업데이트.	2025년 1월 16일
섹션 업데이트	(미국 서부, us-gov-west-1), AWS GovCloud (US) (AWS GovCloud (US)미국 동부, us-	2025년 1월 14일

gov-east-1) 또는 아프리카(케이프타운, af-south-1) 지역에서 지원되지 않는 언어 코드로 지원되는 언어 섹션을 업데이트합니다.

섹션 업데이트

"audio_segments"라는 새 필드를 사용하여 Transcribe Batch json 출력을 확장합니다.

2024년 7월 15일

기능 업데이트

분할의 최대 화자를 10이 아닌 30으로 업데이트합니다.

2024년 5월 10일

섹션 업데이트

생성형 통화 요약 업데이트 및 오류 출력 세부 정보 추가.

2024년 4월 30일

섹션 업데이트

사용자 지정 어휘 열 업데이트 - IPA 및 SoundsLike.

2024년 4월 30일

기능 업데이트

Amazon Transcribe Call Analytics은 이제 생성형 통화 요약을 지원합니다.

2023년 11월 29일

섹션 업데이트

새 PII 수정 및 언어 식별 출력 형식을 업데이트했습니다.

2023년 11월 13일

기능 업데이트

이제 분할을 채널 식별과 결합할 수 있습니다.

2023년 3월 6일

기능 업데이트

이제 채널 식별을 분할과 결합할 수 있습니다.

2023년 3월 6일

섹션 업데이트

IAM 모범 사례가 업데이트되었습니다.

2023년 2월 13일

새로운 언어

Amazon Transcribe는 이제 베트남어와 스웨덴어를 지원합니다.

2022년 12월 6일

새로운 기능	Amazon Transcribe는 이제 실시간 통화 분석을 지원합니다.	2022년 11월 28일
기능 업데이트	이제 힌디어와 태국어로 스트리밍 편집 및 식별을 사용할 수 있습니다.	2022년 11월 11일
섹션 업데이트	스트리밍 편집 및 식별을 위해 새 PII 범주를 사용할 수 있습니다.	2022년 9월 14일
섹션 업데이트	사용자 지정 언어 모델 섹션이 수정되었습니다.	2022년 6월 18일
섹션 업데이트	이제 배치 언어 식별을 통해 오디오 파일당 여러 언어를 식별할 수 있습니다.	2022년 5월 31일
설명서 업데이트	Amazon Transcribe API 참조는 이제 독립적인 설명서입니다.	2022년 4월 1일
새로운 장	Amazon Transcribe, Amazon Transcribe 의료 및 Amazon Transcribe 통화 분석에 대한 새로운 비교 테이블이 포함되었습니다.	2022년 3월 21일
새로운 장	새 SDK 코드 예시 장이 포함되었습니다.	2022년 3월 21일
기능 업데이트:	이제 통화 분석에서 통화 요약 을 제공합니다.	2022년 3월 21일
장 업데이트	소개 장에서는 이제 Amazon Transcribe 사용 사례를 보여줍니다.	2022년 3월 21일

장 업데이트	시작하기 장이 메서드별로 업데이트되었습니다.	2022년 3월 21일
장 업데이트	스트리밍 장이 업데이트 및 재구성되었습니다.	2022년 3월 21일
기능 업데이트:	이제 언어 식별은 스트리밍 트랜스크립션을 통한 사용자 지정 어휘와 사용자 지정 어휘 필터를 지원합니다.	2022년 3월 11일
새로운 이벤트	새 이벤트 유형인 어휘 이벤트가 있습니다.	2022년 2월 7일
섹션 업데이트	사용자 지정 어휘 섹션이 업데이트되었습니다.	2022년 1월 20일
새로운 기능	이제 스트리밍 트랜스크립션에 언어 식별을 사용할 수 있습니다.	2021년 11월 23일
새로운 기능	이제 사용자 지정 언어 모델, 사용자 지정 어휘, 어휘 필터링 및 콘텐츠 편집과 함께 언어 식별을 사용할 수 있습니다.	2021년 10월 29일
새로운 기능	Amazon Transcribe는 이제 스트리밍 트랜스크립션을 통한 사용자 지정 언어 모델을 지원합니다.	2021년 10월 20일
새로운 기능	Amazon Transcribe는 이제 비디오 파일의 자막을 생성할 수 있습니다.	2021년 9월 16일
새로운 기능	Amazon Transcribe는 이제 스트리밍을 위한 PII 편집 및 식별을 지원합니다.	2021년 9월 14일

새로운 기능	Amazon Transcribe는 이제 AWS 계정 리소스 보안 수준을 높이기 위해 AWS KMS 암호화 컨텍스트를 지원합니다.	2021년 9월 10일
새로운 언어	Amazon Transcribe는 이제 아프리카너, 덴마크어, 중국어 (번체), 태국어, 뉴질랜드어, 남아프리카 영어를 지원합니다.	2021년 8월 26일
새로운 기능	Amazon Transcribe는 이제 리소스 태그 지정을 지원합니다.	2021년 8월 24일
새로운 기능	Amazon Transcribe는 이제 배치 트랜스크립션 작업에 대한 통화 분석을 지원합니다.	2021년 8월 4일
새로운 기능	Amazon Transcribe는 이제 배치 사용자 지정 언어 모델과 함께 사용자 지정 어휘의 사용을 지원합니다.	2021년 5월 12일
새로운 기능	Amazon Transcribe는 이제 스트리밍 트랜스크립션의 부분 결과 안정화를 지원합니다.	2021년 5월 11일
새로운 기능	Amazon Transcribe는 이제 사용자 지정 언어 모델에 호주 영어, 영국 영어, 힌디어 및 미국 스페인어를 지원합니다.	2021년 3월 19일
새로운 기능	Amazon Transcribe는 이제 스트리밍 오디오 트랜스크립션을 위한 OGG/OPUS 및 FLAC 코덱을 지원합니다.	2020년 11월 24일

새로운 언어	Amazon Transcribe에 스트리밍 오디오 트랜스크립션을 위해 이탈리아어 및 독일어에 대한 지원이 추가되었습니다.	2020년 11월 4일
AWS 리전 확장	Amazon Transcribe는 현재 프랑크푸르트(eu-central-1) 및 런던(eu-west-2)에서 사용할 수 있습니다.	2020년 11월 4일
새로운 기능	Amazon Transcribe에 배치 트랜스크립션의 인터페이스 VPC 엔드포인트에 대한 지원이 추가되었습니다.	2020년 10월 9일
새로운 기능	Amazon Transcribe에 스트리밍의 채널 식별에 대한 지원이 추가되었습니다.	2020년 9월 17일
새로운 기능	Amazon Transcribe에 배치 트랜스크립션의 자동 언어 식별에 대한 지원이 추가되었습니다.	2020년 9월 15일
새로운 기능	Amazon Transcribe에 스트리밍의 화자 파티셔닝에 대한 지원이 추가되었습니다.	2020년 8월 19일
새로운 기능	Amazon Transcribe에 사용자 지정 언어 모델에 대한 지원이 추가되었습니다.	2020년 8월 5일
새로운 기능	Amazon Transcribe에 스트리밍의 인터페이스 VPC 엔드포인트에 대한 지원이 추가되었습니다.	2020년 6월 26일

새로운 기능	Amazon Transcribe에 스트리밍의 어휘 필터링 지원이 추가되었습니다.	2020년 5월 20일
새로운 기능	Amazon Transcribe에 개인 식별 정보를 자동으로 편집하기 위한 지원이 추가되었습니다.	2020년 2월 26일
새로운 기능	Amazon Transcribe에 트랜스크립션에서 필터링할 단어의 사용자 지정 어휘를 만드는 기능에 대한 지원이 추가되었습니다.	2019년 12월 20일
새로운 기능	Amazon Transcribe에 트랜스크립션 작업을 대기열에 추가하는 기능에 대한 지원이 추가되었습니다.	2019년 12월 19일
새로운 언어	Amazon Transcribe에 걸프 아랍어, 히브리어, 일본어, 말레이어, 스위스 독일어, 텔루구어 및 터키어에 대한 지원이 추가되었습니다.	2019년 11월 21일
AWS 리전 확장	Amazon Transcribe는 이제 아시아 태평양(도쿄)(ap-northeast-1)에서 사용할 수 있습니다.	2019년 11월 21일
새로운 기능	Amazon Transcribe에 대체 트랜스크립션에 대한 지원이 추가되었습니다.	2019년 11월 20일

새로운 언어	Amazon Transcribe에 네덜란드어, 페르시아어, 인도네시아어, 아일랜드 영어, 포르투갈어, 스코틀랜드 영어, 타밀어 및 웨일스 영어에 대한 지원이 추가되었습니다.	2019년 11월 12일
새로운 언어	Amazon Transcribe는 이제 호주 영어(en-AU)에 대한 스트리밍 트랜스크립션을 지원합니다.	2019년 10월 25일
AWS 리전 확장	Amazon Transcribe는 이제 중국(베이징)(cn-north-1) 및 중국(닝샤)(cn-northwest-1)에서 사용할 수 있습니다.	2019년 10월 9일
새로운 기능	Amazon Transcribe를 사용하면 트랜스크립션 출력 파일을 암호화할 KMS key를 제공할 수 있습니다. 자세한 내용은 StartStreamTranscription API 의 OutputEncryptionKMSSKeyId 파라미터를 참조하세요.	2019년 9월 24일
새로운 언어	Amazon Transcribe에 표준 중국어(만다린), 간체, 본토 중국어 및 러시아어 지원이 추가되었습니다.	2019년 8월 23일
새로운 기능	Amazon Transcribe에 WebSocket 프로토콜을 사용한 스트리밍 오디오 트랜스크립션에 대한 지원이 추가되었습니다.	2019년 7월 19일

새로운 기능	AWS CloudTrail는 이제 StartStreamscript API에 대한 이벤트를 기록합니다.	2019년 7월 19일
AWS 리전 확장	Amazon Transcribe는 이제 미국 서부(캘리포니아 북부)(us-west-1)에서 사용할 수 있습니다.	2019년 6월 27일
새로운 언어	Amazon Transcribe에 현대 표준 아랍어에 대한 지원이 추가되었습니다.	2019년 5월 28일
새로운 기능	Amazon Transcribe는 이제 미국 영어의 숫자 단어를 숫자로 트랜스크립션합니다. 예를 들어 "마흔 둘"은 "42"로 트랜스크립션됩니다.	2019년 5월 23일
새로운 언어	Amazon Transcribe에 힌디어와 인도 영어에 대한 지원이 추가되었습니다.	2019년 5월 15일
새 SDK	C++용 AWS SDK는 이제 Amazon Transcribe를 지원합니다.	2019년 5월 8일
새로운 언어	Amazon Transcribe에 스페인어에 대한 지원이 추가되었습니다.	2019년 4월 19일
AWS 리전 확장	Amazon Transcribe는 이제 EU(프랑크푸르트)(eu-central-1) 및 아시아 태평양(서울)(ap-northeast-2)에서 사용 가능합니다.	2019년 4월 18일

새로운 언어	Amazon Transcribe에 영국 영어, 프랑스어 및 캐나다 프랑스어에 대한 스트리밍 트랜스크립션 지원이 추가되었습니다.	2019년 4월 5일
새로운 기능	Ruby V3용 AWS SDK는 이제 Amazon Transcribe를 지원합니다.	2019년 3월 25일
새로운 기능	Amazon Transcribe는 오디오 입력에서 Amazon Transcribe가 인식할 특정 단어 목록인 사용자 지정 어휘를 허용합니다.	2019년 3월 25일
새로운 언어	Amazon Transcribe에 독일어 및 한국어에 대한 지원이 추가되었습니다.	2019년 3월 22일
새로운 언어	Amazon Transcribe는 이제 미국 스페인어(es-US)에 대한 스트리밍 트랜스크립션을 지원합니다.	2019년 2월 7일
AWS 리전 확장	Amazon Transcribe는 이제 남아메리카(상파울루)(sa-east-1)에서 사용 가능합니다.	2019년 2월 7일
AWS 리전 확장	Amazon Transcribe는 이제 아시아 태평양(뭄바이)(ap-south-1), 아시아 태평양(싱가포르)(ap-southeast-1), EU(런던)(eu-west-2) 및 EU(파리)(eu-west3)에서 사용할 수 있습니다.	2019년 1월 24일
새로운 언어	Amazon Transcribe에 프랑스어, 이탈리아어 및 포르투갈어(브라질)에 대한 지원이 추가되었습니다.	2018년 12월 20일

새로운 기능	Amazon Transcribe는 이제 오디오 스트림의 트랜스크립션을 지원합니다.	2018년 11월 19일
새로운 언어	Amazon Transcribe에 호주 영어, 영국 영어 및 캐나다 영어에 대한 지원이 추가되었습니다.	2018년 11월 15일
AWS 리전 확장	Amazon Transcribe는 이제 캐나다(중부)(ca-central-1) 및 아시아 태평양(시드니)(ap-southeast-2)에서 사용할 수 있습니다.	2018년 7월 17일
새로운 기능	이제 트랜스크립션 작업의 출력을 저장할 위치를 지정할 수 있습니다.	2018년 7월 11일
새로운 기능	AWS CloudTrail 및 Amazon CloudWatch Events 통합이 추가되었습니다.	2018년 6월 28일
새로운 기능	Amazon Transcribe에 사용자 지정 어휘에 대한 지원이 추가되었습니다.	2018년 4월 4일
새 안내서	이 문서는 첫 번째 Amazon Transcribe 개발자 안내서 릴리스입니다.	2017년 11월 29일

AWS 용어집

최신 AWS 용어는 AWS 용어집 참조서의 [AWS 용어집](#)을 참조하십시오.

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.