



개발자 안내서

# Amazon Transcribe



# Amazon Transcribe: 개발자 안내서

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

Amazon의 상표 및 브랜드 디자인은 Amazon 외 제품 또는 서비스와 함께, 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
요금 .....	2
지역 가용성 및 할당량 .....	2
사용 가능한 기능 .....	4
지원되는 언어 .....	6
지원되는 프로그래밍 언어 .....	14
문자 집합 .....	15
압하스어 .....	19
아프리카스어 .....	21
아랍어 .....	22
아스투리아스어 .....	23
아제르바이잔어 .....	23
아르메니아어 .....	24
바쉬르어 .....	25
바스크어 .....	28
벨라루스어 .....	28
벵골어 .....	29
보스니아어 .....	31
불가리아어 .....	32
카탈루냐어 .....	33
소라니어 .....	34
중국어, 간체 .....	35
중국어, 번체 .....	36
크로아티아어 .....	37
체코어 .....	38
덴마크어 .....	38
네덜란드어 .....	39
영어 .....	40
에스토니아어 .....	40
페르시아어 .....	41
핀란드어 .....	42
프랑스어 .....	42
갈리시아어 .....	44

조지아어 .....	44
독일어 .....	45
그리스어 .....	46
구자라트어 .....	47
하우사어 .....	49
히브리어 .....	50
힌디어 .....	50
헝가리어 .....	53
아이슬란드어 .....	53
인도네시아어 .....	54
이탈리아어 .....	54
일본어 .....	55
커바일어 .....	55
칸나다어 .....	56
카자흐어 .....	58
키냐르완다어 .....	59
한국어 .....	60
키르기스어 .....	60
라트비아어 .....	63
리투아니아어 .....	63
간다어 .....	64
마케도니아어 .....	64
말레이어 .....	67
말라얄람어 .....	67
몰타어 .....	69
마라티어 .....	70
메도우 마리 .....	72
몽골어 .....	74
노르웨이어(보크몰) .....	77
오디아/오리야어 .....	77
파슈토어 .....	79
폴란드어 .....	81
포르투갈어 .....	81
펀자브어 .....	83
루마니아어 .....	84
러시아어 .....	85

세르비아어 .....	86
싱할라어 .....	89
슬로바키아어 .....	91
슬로베니아어 .....	92
소말리아어 .....	92
스페인어 .....	93
순다어 .....	93
스와힐리어 .....	94
스웨덴어 .....	94
타갈로그어/필리핀어 .....	95
타밀어 .....	95
타타르어 .....	97
텔루구어 .....	99
태국어 .....	101
터키어 .....	103
우크라이나어 .....	104
위구르어 .....	105
우즈베크어 .....	108
베트남어 .....	109
웨일스어 .....	113
월로프어 .....	114
줄루어 .....	115
작동 방식 .....	116
데이터 입력 및 출력 .....	116
미디어 형식 .....	117
오디오 채널 .....	118
샘플 속도 .....	119
출력 .....	119
숫자 표기하기 .....	122
시작하기 .....	128
에 가입AWS 계정 .....	128
AWS CLI및 SDK 설치 .....	129
IAM자격 증명 구성 .....	129
Amazon S3버킷 생성 .....	130
IAM 정책 생성 .....	130
를 사용하여 필사하기AWS Management Console .....	132

를 사용하여 필사하기AWS CLI .....	142
새 트랜스크립션 작업 시작 .....	142
트랜스크립션 작업의 상태 가져오기 .....	143
트랜스크립션 작업 목록 작성 .....	145
트랜스크립션 작업 삭제하기 .....	145
SDK를 AWS 사용한 트랜스크립션 .....	146
SDK로 작업하기 AWS .....	157
HTTP를 이용한 트랜스크립팅 또는 WebSockets .....	159
스트리밍 트랜스크립션 .....	161
모범 사례 .....	162
스트리밍 및 부분 결과 .....	163
부분 결과 안정화 .....	164
스트리밍 트랜스크립션 설정 .....	168
이벤트 스트림 인코딩 .....	181
데이터 프레임 .....	183
Job 대기열 .....	185
작업 .....	185
리소스에 태그 지정 .....	189
태그 기반 액세스 제어 .....	190
Amazon Transcribe리소스에 태그 추가 .....	190
화자 파티셔닝(분할) .....	195
배치 트랜스크립션의 화자 파티셔닝 .....	196
스트리밍 트랜스크립션의 화자 파티셔닝 .....	199
출력 예 .....	202
멀티 채널 오디오 트랜스크립팅 .....	208
일괄 트랜스크립션에서 채널 식별 사용 .....	209
스트리밍 트랜스크립션에서 채널 식별 사용 .....	213
출력 예 .....	214
식별 언어 .....	222
배치 언어 식별 .....	222
다국어 오디오의 언어 식별 .....	223
언어 식별 정확도 향상 .....	224
언어 식별과 다른 Amazon Transcribe 기능 결합 .....	225
배치 트랜스크립션 작업을 통한 언어 식별 사용 .....	225
스트리밍 언어 식별 .....	232
다국어 오디오의 언어 식별 .....	233

스트리밍 미디어를 통한 언어 식별 사용 .....	233
대체 트랜스크립션 .....	240
대체 트랜스크립션 요청 .....	242
트랜스크립션 .....	247
사용자 지정 어휘 .....	248
사용자 지정 어휘 테이블과 목록 비교 .....	249
테이블을 사용하여 사용자 지정 어휘 생성 .....	250
목록을 사용하여 사용자 지정 어휘 만들기 .....	260
사용자 지정 어휘 사용 .....	263
사용자 지정 언어 모델 .....	270
데이터 원본 .....	270
트레이닝 vs. 튜닝 데이터 .....	271
사용자 지정 언어 모델 만들기 .....	272
사용자 지정 언어 모델 사용 .....	277
단어 필터링 .....	285
어휘 필터 만들기 .....	286
사용자 지정 어휘 필터 만들기 .....	287
사용자 지정 단어 필터 사용 .....	291
일괄 트랜스크립션에 사용자 지정 어휘 필터 사용 .....	263
스트리밍 트랜스크립션에서 사용자 지정 어휘 필터 사용 .....	267
독성 음성 감지 .....	300
독성 음성 감지 사용 .....	301
일괄 트랜스크립션에 독성 음성 감지 기능 사용 .....	301
출력 예 .....	306
성적 증명서 수정 .....	308
배치 작업의 PII 편집 .....	308
실시간 스트림에서 PII 편집 또는 식별 .....	315
출력 예 .....	320
편집된 출력 예제 (일괄 처리) .....	320
편집된 스트리밍 출력 예시 .....	323
PII 식별 출력 예시 .....	324
자막 생성 .....	326
자막 파일 생성 .....	327
콜센터 오디오 분석 .....	330
일반 사용 사례 .....	330
고려 사항 및 추가 정보 .....	332

리전 가용성 및 할당량 .....	332
통화 후 분석 .....	333
통화 후 인사이트 .....	334
범주 생성 .....	336
트랜스크립션 시작 .....	348
통화 후 분석 출력 .....	356
생성형 통화 요약 활성화 .....	369
실시간 Call Analytics .....	373
실시간 인사이트 .....	374
범주 생성 .....	376
실시간 트랜스크립션을 통한 통화 후 분석 .....	383
트랜스크립션 시작 .....	390
실시간 Call Analytics 출력 .....	398
Amazon Chime 통화 내용 받아쓰기 .....	403
코드 예시 .....	405
작업 .....	406
CreateVocabulary .....	407
DeleteMedicalTranscriptionJob .....	410
DeleteTranscriptionJob .....	413
DeleteVocabulary .....	416
GetTranscriptionJob .....	418
GetVocabulary .....	422
ListMedicalTranscriptionJobs .....	424
ListTranscriptionJobs .....	429
ListVocabularies .....	435
StartMedicalTranscriptionJob .....	439
StartStreamTranscriptionAsync .....	450
StartTranscriptionJob .....	454
UpdateVocabulary .....	473
시나리오 .....	476
사용자 지정 어휘 생성 및 세부 조정 .....	476
오디오 트랜스크립션 및 작업 데이터 가져오기 .....	486
교차 서비스 예시 .....	497
Amazon Transcribe 앱 구축 .....	498
Amazon Transcribe 스트리밍 앱 구축 .....	498
텍스트를 스피치로, 다시 스피치에서 텍스트로 변환 .....	499



보안 .....	501
ID 및 액세스 관리 .....	501
고객 .....	502
ID를 통한 인증 .....	503
정책을 사용한 액세스 관리 .....	506
IAM의 Amazon Transcribe 작동 방식 .....	508
혼동된 대리자 방지 .....	514
자격 증명 기반 정책 예제 .....	515
문제 해결 .....	524
데이터 보호 .....	526
인터넷워크 트래픽 개인 정보 보호 .....	527
데이터 암호화 .....	527
서비스 개선을 위한 데이터 사용 거부 .....	530
모니터링 Amazon Transcribe .....	530
를 통한 모니터링 CloudWatch .....	531
Amazon Transcribe 를 통한 모니터링 CloudTrail .....	531
Amazon EventBridge 와 함께 사용 Amazon Transcribe .....	535
규정 준수 확인 .....	542
복원성 .....	543
인프라 보안 .....	544
취약성 분석 및 관리 .....	544
VPC 엔드포인트(AWS PrivateLink) .....	544
공유 서브넷 .....	547
보안 모범 사례 .....	547
Amazon Transcribe 의료 .....	549
지역 가용성 및 할당량 .....	550
Medical .....	551
의료 용어 및 측정값 필사 .....	552
숫자 표기하기 .....	555
의료 대화 내용 필사하기 .....	557
오디오 파일 스크립트 작성 .....	558
실시간 스트림 트랜스크립팅 하기 .....	563
스피커 분할 사용 설정 설정 설정 설정 설정 .....	566
멀티 채널 오디오 트랜스크립팅 .....	575
의료 받아쓰기 받아쓰기 .....	583
오디오 파일 스크립트 작성 .....	584

스트리밍 중인 의료 받아쓰기 .....	588
맞춤형 의료 어휘 생성 및 사용 .....	591
의료 맞춤 어휘를 위한 텍스트 파일 만들기 .....	591
텍스트 파일을 사용하여 맞춤형 의료 어휘 만들기 .....	596
의료 맞춤 어휘를 사용하여 오디오 파일 녹음하기 .....	598
의료 맞춤 어휘를 사용하여 실시간 스트림 자막 작성하기 .....	600
Amazon Transcribe의료용 문자 세트 .....	602
트랜스크립트에서 PHI 식별하기 .....	604
오디오 파일에서 PHI 식별하기 .....	605
실시간 스트림에서 PHI 식별 .....	609
대체 트랜스크립션 생성 .....	611
VPC 엔드포인트(AWS PrivateLink) .....	614
에 대한 고려 사항Amazon Transcribe의료용 VPC 엔드포인트 .....	614
에 대한 인터페이스 VPC 엔드포인트 생성Amazon Transcribe의료 .....	614
다음에 대한 VPC 엔드포인트 정책 생성Amazon Transcribe의료 스트리밍 .....	615
공유 서브넷 .....	616
AWS HealthScribe .....	617
트랜스크립트 파일 .....	618
임상 문서 파일 .....	619
AWS HealthScribe 작업 시작 .....	620
출력 예시 .....	622
AWS HealthScribe의 저장 데이터 암호화 .....	635
고객 관리형 키를 생성하려면 .....	636
AWS HealthScribe에 대한 고객 관리형 키 지정 .....	638
AWS KMS 암호화 컨텍스트 .....	638
사용 설명서 기록 .....	639
AWS 용어집 .....	649
.....	dcl

# Amazon Transcribe란 무엇인가요?

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

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

미디어를 실시간으로 트랜스크립션 (스트리밍) 하거나 Amazon S3 버킷에 있는 미디어 파일을 트랜스크립션 (배치) 할 수 있습니다. 각 트랜스크립션 유형에 어떤 언어가 지원되는지 알아보려면 [지원되는 언어 및 언어별 기능](#) 표를 참조하십시오.

## 주제

- [Amazon Transcribe 및 HIPAA 자격](#)
- [요금](#)
- [지역 가용성 및 할당량](#)

[이란 무엇입니까 Amazon Transcribe?](#) 를 참조하십시오. 이 서비스에 대한 짧은 비디오 투어를 위해

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

### Tip

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

## Amazon Transcribe 및 HIPAA 자격

Amazon Transcribe HIPAA 자격 및 BAA가 적용되므로 BAA 고객은 사용 중 및 전송 중인 AWS 모든 PHI를 암호화해야 합니다. 자동 PHI 식별은 Amazon Transcribe 운영되는 모든 지역에서 추가 비용 없이 사용할 수 있습니다. 자세한 내용은 [HIPAA 자격 및 BAA](#)를 참조하십시오.

## 요금

Amazon Transcribe pay-as-you-go 서비스입니다. 요금은 몇 초 분량의 녹음된 오디오를 기준으로 하며 월 단위로 청구됩니다.

사용량은 1초 단위로 청구되며 최소 청구 시간은 10분입니다. PII 콘텐츠 편집 및 사용자 지정 언어 모델과 같은 기능에는 추가 요금이 부과됩니다.

각 AWS 리전 비용에 대한 자세한 내용은 [Amazon Transcribe 요금](#)을 참조하십시오.

## 지역 가용성 및 할당량

Amazon Transcribe 다음에서 지원됩니다 AWS 리전.

리전(Region)	트랜스크립션 유형
af-south	배치, 스트림
ap-east-	배치
ap-northeast-1(도쿄)	배치, 스트림
ap-northeast-2(서울)	배치, 스트림
ap-south	배치, 스트림
ap-southeast-1(싱가포르)	배치, 스트림
ap-southeast-2(시드니)	배치, 스트림
ca-central-	배치, 스트림
eu-central-1(프랑크푸르트)	배치, 스트림
eust-	배치
eu-west-1(아일랜드)	배치, 스트림
eust-2	배치, 스트림
eust-3	배치

리전(Region)	트랜스크립션 유형
me-south	배치
sa-east-	배치, 스트림
us-east-	배치, 스트림
us-east-2	배치, 스트림
us-gov-east-1 (GovCloud, 미국 동부)	배치, 스트림
us-gov-west-1 (GovCloud, 미국 서부)	배치, 스트림
us-west-	배치
us-west-2(오리건)	배치, 스트림

**⚠ Important**

지역 지원은 Amazon Transcribe [Amazon Transcribe Medical](#), 및 [통화 분석](#)에 따라 다릅니다.

지원되는 각 지역의 엔드포인트를 얻으려면 AWS일반 참조의 [서비스 엔드포인트](#)를 참조하십시오.

트랜스크립트와 관련된 할당량 목록은 일반 참조서의 [서비스 할당량](#)을 참조하십시오. AWS 요청 시 일부 할당량을 변경할 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 표에는 기능 비교가 나와 있습니다.

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	<a href="#">Amazon Transcribe Medical</a> <sup>1</sup>	<a href="#">Amazon Transcribe 통화 분석</a>
<b>구성 옵션</b>			
<a href="#">대체 트랜스크립션</a>	배치, 스트리밍	배치, 스트리밍	아니요
<a href="#">채널 식별</a>	배치, 스트리밍	배치, 스트리밍	post-call, real-time
<a href="#">작업 대기열</a>	일괄	아니요	post-call
<a href="#">언어 식별</a>	배치, 스트리밍	아니요	post-call
<a href="#">다국어 식별</a>	배치, 스트리밍	아니요	아니요
<a href="#">화자 분할</a>	배치, 스트리밍	배치, 스트리밍	post-call
<a href="#">숫자 트랜스크립션</a> <sup>2</sup>	배치, 스트리밍	배치, 스트리밍	post-call, real-time
<b>대화 분석</b>			
<a href="#">통화 특성</a>	아니요	아니요	post-call
<a href="#">통화 요약</a> <sup>2</sup>	아니요	아니요	post-call
<a href="#">사용자 지정 분류</a>	아니요	아니요	post-call
<a href="#">실시간 범주 이벤트</a>	아니요	아니요	real-time
<a href="#">실시간 문제 감지</a> <sup>2</sup>	아니요	아니요	real-time
<a href="#">실시간 화자 감정</a>	아니요	아니요	real-time

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

 <sup>1</sup> Amazon Transcribe Medical은 미국 영어로만 제공됩니다.

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

## 지원되는 언어 및 언어별 기능

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

[전체 Amazon Transcribe 기능 목록을 보려면 기능 요약을 참조하십시오.](#)

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

언어	언어 코드	<a href="#">데이터 입력</a>	<a href="#">숫자 표기하기</a>	<a href="#">두문자어</a>	<a href="#">사용자 지정 언어 모델</a>	<a href="#">수정</a>	<a href="#">Call Analytics</a> * -
<a href="#">압하스어</a>	ab-GE	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">아프리카나스어</a>	af-ZA	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">아랍어, 걸프</a>	ar-AE	일괄	아니요	아니요	아니요	아니요	post-call
<a href="#">아랍어, 현대 표준어</a>	ar-SA	일괄	아니요	아니요	아니요	아니요	아니요
<a href="#">아르메니아어</a>	hy-AM	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">아스투리아스어</a>	ast-ES	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">아제르바이잔어</a>	az-AZ	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">바쉬르어</a>	ba-RU	일괄	아니요	일괄	아니요	아니요	아니요



언어	언어 코드	<a href="#">데이터 입력</a>	<a href="#">숫자 표기하기</a>	<a href="#">두문자어</a>	<a href="#">사용자 지정 언어 모델</a>	<a href="#">수정</a>	<a href="#">Call Analytics</a> * -
<a href="#">바스크어</a>	eu-ES	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">벨라루스어</a>	be-BY	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">벥골어</a>	bn-IN	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">보스니아어</a>	bs-BA	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">불가리아어</a>	bg-BG	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">카탈루냐어</a>	ca-ES	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">소라니어, 이란</a>	ckb-IR	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">소라니어, 이라크</a>	ckb-IQ	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">중국어, 간체</a>	zh-CN	배치, 스트리밍	아니요	아니요	아니요	아니요	post-call
<a href="#">중국어, 번체</a>	zh-TW	일괄	아니요	아니요	아니요	아니요	아니요
<a href="#">크로아티아어</a>	hr-HR	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">체코어</a>	cs-CZ	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">덴마크어</a>	da-DK	일괄	아니요	일괄	아니요	아니요	아니요

언어	언어 코드	<a href="#">데이터 입력</a>	<a href="#">숫자 표기하기</a>	<a href="#">두문자어</a>	<a href="#">사용자 지정 언어 모델</a>	<a href="#">수정</a>	<a href="#">Call Analytics</a> * -
<a href="#">네덜란드어</a>	nl-NL	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">영어, 호주</a>	en-AU	배치, 스트리밍	배치, 스트리밍	배치, 스트리밍	배치, 스트리밍	스트리밍	post-call, real-time
<a href="#">영어, 영국</a>	en-GB	배치, 스트리밍	배치, 스트리밍	배치, 스트리밍	배치, 스트리밍	스트리밍	post-call, real-time
<a href="#">영어, 인도</a>	en-IN	일괄	일괄	일괄	아니요	아니요	post-call
<a href="#">영어, 아일랜드</a>	en-IE	일괄	일괄	일괄	아니요	아니요	post-call
<a href="#">영어, 뉴질랜드</a>	en-NZ	일괄	일괄	일괄	아니요	아니요	아니요
<a href="#">영어, 스코틀랜드</a>	en-AB	일괄	일괄	일괄	아니요	아니요	post-call
<a href="#">영어, 남아프리카 공화국</a>	en-ZA	일괄	일괄	일괄	아니요	아니요	아니요
<a href="#">영어, 미국</a>	en-US	배치, 스트리밍	배치, 스트리밍	배치, 스트리밍	배치, 스트리밍	배치, 스트리밍	post-call, real-time
<a href="#">영어, 웨일스</a>	en-WL	일괄	일괄	일괄	아니요	아니요	post-call
<a href="#">에스토니아어</a>	et-ET	일괄	아니요	일괄	아니요	아니요	아니요

언어	언어 코드	<u>데이터 입력</u>	<u>숫자 표기하기</u>	<u>두문자어</u>	<u>사용자 지정 언어 모델</u>	<u>수정</u>	<u>Call Analytics</u> * -
<u>페르시아어</u>	fa-IR	일괄	아니요	아니요	아니요	아니요	아니요
<u>핀란드어</u>	fi-FI	일괄	아니요	일괄	아니요	아니요	아니요
<u>프랑스어</u>	fr-FR	배치, 스트리밍	아니요	배치, 스트리밍	아니요	아니요	post-call, real-time
<u>프랑스어, 캐나다</u>	fr-CA	배치, 스트리밍	아니요	배치, 스트리밍	아니요	아니요	post-call, real-time
<u>갈리시아어</u>	gl-ES	일괄	아니요	일괄	아니요	아니요	아니요
<u>조지아어</u>	ka-GE	일괄	아니요	일괄	아니요	아니요	아니요
<u>독일어</u>	de-DE	배치, 스트리밍	배치, 스트리밍	배치, 스트리밍	배치, 스트리밍	아니요	post-call, real-time
<u>독일어, 스위스</u>	de-CH	일괄	일괄	일괄	아니요	아니요	post-call
<u>그리스어</u>	el-GR	일괄	아니요	일괄	아니요	아니요	아니요
<u>구자라트어</u>	gu-IN	일괄	아니요	일괄	아니요	아니요	아니요
<u>하우사어</u>	ha-NG	일괄	아니요	일괄	아니요	아니요	아니요
<u>히브리어</u>	he-IL	일괄	아니요	아니요	아니요	아니요	아니요
<u>힌디어, 인도</u>	hi-IN	배치, 스트리밍	아니요	배치, 스트리밍	일괄	아니요	post-call
<u>헝가리어</u>	hu-HU	일괄	아니요	일괄	아니요	아니요	아니요

언어	언어 코드	<a href="#">데이터 입력</a>	<a href="#">숫자 표기하기</a>	<a href="#">두문자어</a>	<a href="#">사용자 지정 언어 모델</a>	<a href="#">수정</a>	<a href="#">Call Analytics</a> * -
<a href="#">아이슬란드어</a>	is-IS	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">인도네시아어</a>	id-ID	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">이탈리아어</a>	it-IT	배치, 스트리밍	아니요	배치, 스트리밍	아니요	아니요	post-call, real-time
<a href="#">일본어</a>	ja-JP	배치, 스트리밍	아니요	아니요	배치, 스트리밍	아니요	post-call
<a href="#">커바일어</a>	kab-DZ	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">칸나다어</a>	kn-IN	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">카자흐어</a>	kk-KZ	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">키냐르완다어</a>	rw-RW	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">한국어</a>	ko-KR	배치, 스트리밍	아니요	아니요	아니요	아니요	post-call
<a href="#">키르기스어</a>	ky-KG	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">라트비아어</a>	lv-LV	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">리투아니아어</a>	lt-LT	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">간다어</a>	lg-IN	일괄	아니요	일괄	아니요	아니요	아니요

언어	언어 코드	<a href="#">데이터 입력</a>	<a href="#">숫자 표기하기</a>	<a href="#">두문자어</a>	<a href="#">사용자 지정 언어 모델</a>	<a href="#">수정</a>	<a href="#">Call Analytics</a> * -
<a href="#">마케도니아어</a>	mk-MK	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">말레이어</a>	ms-MY	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">말라얄람어</a>	ml-IN	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">몰타어</a>	mt-MT	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">마라티어</a>	mr-IN	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">메도우 마리</a>	mhr-RU	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">몽골어</a>	mn-MN	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">노르웨이어(보크몰)</a>	no-NO	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">오디아/오리아어</a>	or-IN	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">파슈토어</a>	ps-AF	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">폴란드어</a>	pl-PL	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">포르투갈어</a>	pt-PT	일괄	아니요	일괄	아니요	아니요	post-call
<a href="#">포르투갈어, 브라질</a>	pt-BR	배치, 스트리밍	아니요	배치, 스트리밍	아니요	아니요	post-call, real-time
<a href="#">펀자브어</a>	pa-IN	일괄	아니요	일괄	아니요	아니요	아니요

언어	언어 코드	<u>데이터 입력</u>	<u>숫자 표기하기</u>	<u>두문자어</u>	<u>사용자 지정 언어 모델</u>	<u>수정</u>	<u>Call Analytics</u> * -
<u>루마니아어</u>	ro-RO	일괄	아니요	일괄	아니요	아니요	아니요
<u>러시아어</u>	ru-RU	일괄	아니요	아니요	아니요	아니요	아니요
<u>세르비아어</u>	sr-RS	일괄	아니요	일괄	아니요	아니요	아니요
<u>싱할라어</u>	si-LK	일괄	아니요	일괄	아니요	아니요	아니요
<u>슬로바키아어</u>	sk-SK	일괄	아니요	일괄	아니요	아니요	아니요
<u>슬로베니아어</u>	sl-SI	일괄	아니요	일괄	아니요	아니요	아니요
<u>소말리아어</u>	so-SO	일괄	아니요	일괄	아니요	아니요	아니요
<u>스페인어</u>	es-ES	일괄	아니요	일괄	아니요	아니요	post-call
<u>스페인어, 미국</u>	es-US	배치, 스트리밍	아니요	배치, 스트리밍	배치, 스트리밍	배치, 스트리밍	post-call, real-time
<u>순다어</u>	su-ID	일괄	아니요	일괄	아니요	아니요	아니요
<u>스와힐리어, 케냐</u>	sw-KE	일괄	아니요	일괄	아니요	아니요	아니요
<u>스와힐리어, 부룬디</u>	sw-BI	일괄	아니요	일괄	아니요	아니요	아니요
<u>스와힐리어, 르완다</u>	sw-RW	일괄	아니요	일괄	아니요	아니요	아니요

언어	언어 코드	<a href="#">데이터 입력</a>	<a href="#">숫자 표기하기</a>	<a href="#">두문자어</a>	<a href="#">사용자 지정 언어 모델</a>	<a href="#">수정</a>	<a href="#">Call Analytics</a> * -
<a href="#">스와힐리어</a> , 탄자니아	sw-TZ	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">스와힐리어</a> , 우간다	sw-UG	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">스웨덴어</a>	sv-SE	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">타갈로그어/필리핀어</a>	tl-PH	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">타밀어</a>	ta-IN	일괄	아니요	아니요	아니요	아니요	아니요
<a href="#">타타르어</a>	tt-RU	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">텔루구어</a>	te-IN	일괄	아니요	아니요	아니요	아니요	아니요
<a href="#">태국어</a>	th-TH	배치, 스트리밍	아니요	배치, 스트리밍	아니요	아니요	아니요
<a href="#">터키어</a>	tr-TR	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">우크라이나어</a>	uk-UA	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">위구르어</a>	ug-CN	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">우즈베크어</a>	uz-UZ	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">베트남어</a>	vi-VN	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">웨일스어</a>	cy-WL	일괄	아니요	일괄	아니요	아니요	아니요

언어	언어 코드	<a href="#">데이터 입력</a>	<a href="#">숫자 표기하기</a>	<a href="#">두문자어</a>	<a href="#">사용자 지정 언어 모델</a>	<a href="#">수정</a>	<a href="#">Call Analytics</a> * -
<a href="#">월로프어</a>	wo-SN	일괄	아니요	일괄	아니요	아니요	아니요
<a href="#">줄루어</a>	zu-ZA	일괄	아니요	일괄	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 다음 SDK를 지원합니다. AWS

배치 트랜스크립션	스트리밍 트랜스크립션
<a href="#">.NET</a>	스트리밍에는 .NET이 지원하지 않습니다.
<a href="#">AWS 명령줄 인터페이스(CLI)</a>	스트리밍에는 CLI가 지원되지 않습니다.
<a href="#">C++</a>	<a href="#">C++</a>
<a href="#">Go</a>	<a href="#">Go</a>
<a href="#">Java V2</a>	<a href="#">Java V2</a>
<a href="#">JavaScript</a>	<a href="#">JavaScript V3</a>
<a href="#">PHP V3</a>	<a href="#">PHP V3</a>
<a href="#">Python Boto3</a>	<a href="#">Python 스트리밍 SDK를 위한 Amazon Transcribe</a>
<a href="#">Ruby V3</a>	<a href="#">Ruby V3</a>



배치 트랜스크립션	스트리밍 트랜스크립션
<a href="#">Rust</a>	<a href="#">Rust</a>

에서 SDK를 사용하는 방법에 대한 자세한 내용은 [을 Amazon Transcribe참조하십시오. SDK를 AWS 사용한 트랜스크립션](#)

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

### Tip

다음 GitHub 리포지토리에서 SDK 코드 샘플을 찾을 수 있습니다.

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

## 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](#)
- [아프리카스어 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\(대만\), 번체 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/오리야어 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
а	0430	Ӏ	0459
б	0431	ӑ	045A
в	0432	ӓ	045B
г	0433	ӕ	045C
д	0434	#	045D
е	0435	ӗ	045E
ж	0436	ӝ	045F
з	0437	г	0491
и	0438	ƒ	0493
й	0439	ж	0497
к	043A	ӡ	0499
л	043B	қ	049B
м	043C	к	049F

문자	코드	문자	코드
н	043D	κ	04A1
о	043E	ჟ	04A3
п	043F	н	04A5
р	0440	Ꞥ	04A9
с	0441	ç	04AB
т	0442	т	04AD
у	0443	γ	04AF
ф	0444	ϥ	04B1
х	0445	χ	04B3
ц	0446	ц	04B5
ч	0447	ч	04B7
ш	0448	h	04BB
щ	0449	є	04BD
ъ	044A	є	04BF
ы	044B	#	04CA
ь	044C	ă	04D1
э	044D	ä	04D3
ю	044E	ě	04D7
я	044F	ə	04D9
#	0450	з	04E1

문자	코드	문자	코드
ë	0451	й	04E3
ђ	0452	ö	04E7
í	0453	ө	04E9
є	0454	ÿ	04EF
s	0455	ÿ	04F1
i	0456	ÿ	04F3
ï	0457	#	04F7
j	0458	ÿ	04F9
#	0525		

## 아프리카스어 문자 집합

아프리카스어 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

문자	코드	문자	코드
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
أ	0623	ص	0635
ؤ	0624	ض	0636
إ	0625	ط	0637
ئ	0626	ظ	0638
ا	0627	ع	0639
ب	0628	غ	063A
ة	0629	ف	0641
ت	062A	ق	0642
ث	062B	ك	0643
ج	062C	ل	0644
ح	062D	م	0645



문자	코드	문자	코드
ġ	062E	ı	0646
đ	062F	o	0647
đ	0630	o	0648
ر	0631	س	0649
ز	0632	س	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
մ	0561	ւ	0574
բ	0562	յ	0575
գ	0563	փ	0576
դ	0564	ց	0577

문자	코드	문자	코드
ɛ	0565	ɪ	0578
ɶ	0566	ɹ	0579
ɸ	0567	ʉ	057A
ɹ	0568	ɻ	057B
ɻ	0569	ɹ	057C
ɸ	056A	u	057D
ɸ	056B	ɹ	057E
ɹ	056C	ɹ	057F
ɸ	056D	ɸ	0580
ɸ	056E	g	0581
ɸ	056F	ɹ	0582
h	0570	ɸ	0583
ɸ	0571	f	0584
ɸ	0572	o	0585
ɸ	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
а	0430	љ	0459
б	0431	њ	045A
в	0432	ћ	045B
г	0433	ќ	045C
д	0434	#	045D
е	0435	ђ	045E
ж	0436	џ	045F
з	0437	ѓ	0491
и	0438	ѣ	0493
й	0439	ж	0497
к	043A	ѕ	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

문자	코드	문자	코드
y	0443	γ	04AF
φ	0444	ϕ	04B1
x	0445	χ	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	ө	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

문자	코드	문자	코드
ï	0457	#	04F7
j	0458	ÿ	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
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
н	043D	я	044F
о	043E	ë	0451
п	043F	i	0456
р	0440	ÿ	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
ঈ	0987	ব	09AC
ঐ	0988	ভ	09AD
ঊ	0989	ম	09AE
ঋ	098A	য	09AF
ঌ	098B	র	09B0
এ	098F	ল	09B2
ঐ	0990	শ	09B6
ও	0993	ষ	09B7
ঔ	0994	স	09B8
ক	0995	হ	09B9
খ	0996	.	09BC
গ	0997	#	09BD
ঘ	0998	†	09BE



문자	코드	문자	코드
ঔ	0999	ঐ	09BF
চ	099A	ঋ	09C0
ছ	099B	ূ	09C1
জ	099C	ৃ	09C2
ঝ	099D	ৄ	09C3
ঞ	099E	৅	09C4
ট	099F	৆	09C7
ঠ	09A0	ে	09C8
ড	09A1	ৈ	09CB
ঢ	09A2	৉	09CC
ণ	09A3	৊	09CD
ত	09A4	#	09CE
থ	09A5	ো	09D7

## 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

문자	코드	문자	코드
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
ئ	0626	م	0645
ا	0627	ن	0646
ب	0628	و	0648
ت	062A	پ	067E
ج	062C	چ	0686
ح	062D	ر	0695
خ	062E	ژ	0698
د	062F	ف	06A4
ر	0631	ک	06A9
ز	0632	گ	06AF
س	0633	ل	06B5
ش	0634	ھ	06BE
ع	0639	ؤ	06C6
غ	063A	ۇ	06C7

문자	코드	문자	코드
ف	0641	س	06CC
ق	0642	ئ	06CE
ج	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), 번체 문자 집합

중국어(번체) 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(번체)의 4개 톤을 나타냅니다. 다음 테이블은 톤 표시가 ㄉㄚ 단어에 매핑되는 방법을 보여줍니다.

톤	톤 표시
톤 1	ㄉㄚ

톤	톤 표시
톤 2	ㅍㅑ'
톤 3	ㅍㅑ~
톤 4	ㅍㅑ`

중국어(번체) 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
##	###`-##		
##	###~-##'		
####	###'-##'-##`-##~		
#####	#`-###~-##'-##~-###`	###	

## 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	đ	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
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

문자	코드	문자	코드
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

문자	코드	문자	코드
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	ظ	0638
آ	0622	ع	0639
أ	0623	غ	063A
ؤ	0624	ف	0641
ئ	0626	ق	0642
ا	0627	ج	0644
ب	0628	م	0645
ت	062A	ن	0646
ث	062B	ه	0647
ج	062C	و	0648
ح	062D	ـ	064E
خ	062E	ـ	064F
د	062F	ـ	0650
ذ	0630	ـ	0651

문자	코드	문자	코드
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
Ï	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
ႀ	10D0	႐	10E0
ႁ	10D1	႑	10E1
ႂ	10D2	႒	10E2

문자	코드	문자	코드
Œ	10D3	Ɔ	10E3
Ɔ	10D4	Ɔ	10E4
Ɔ	10D5	Ɔ	10E5
Ɔ	10D6	Ɔ	10E6
Ɔ	10D7	Ɔ	10E7
Ɔ	10D8	Ɔ	10E8
Ɔ	10D9	Ɔ	10E9
Ɔ	10DA	Ɔ	10EA
Ɔ	10DB	Ɔ	10EB
Ɔ	10C	Ɔ	10EC
Ɔ	10DD	Ɔ	10ED
Ɔ	10DE	Ɔ	10EE
Ɔ	10DF	Ɔ	10EF
Ɔ	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
á	03AC	ν	03BD
έ	03AD	ξ	03BE
ή	03AE	ο	03BF
í	03AF	π	03C0
ü	03B0	ρ	03C1
α	03B1	ς	03C2
β	03B2	σ	03C3
γ	03B3	τ	03C4



문자	코드	문자	코드
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	ε	0AA6
˙	0A82	૬	0AA7
:	0A83	૭	0AA8

문자	코드	문자	코드
अ	0A85	५	0AAA
आ	0A86	६	0AAB
इ	0A87	७	0AAC
ई	0A88	८	0AAD
उ	0A89	९	0AAE
ऊ	0A8A	५	0AAF
ऋ	0A8B	२	0AB0
ॠ	0A8D	६	0AB2
अे	0A8F	७	0AB3
अै	0A90	९	0AB5
ओ	0A91	श	0AB6
ओ	0A93	५	0AB7
औ	0A94	स	0AB8
ऋ	0A95	५	0AB9
ॠ	0A96	.	0ABC
ॠ	0A97	।	0ABE
ॠ	0A98	।	0ABF
ॠ	0A99	।	0AC0
ॠ	0A9A	ॠ	0AC1
ॠ	0A9B	ॠ	0AC2

문자	코드	문자	코드
ഔ	0A9C	ഓ	0AC3
ഔ	0A9D	ഓ	0AC5
ഔ	0A9E	ഓ	0AC7
ഔ	0A9F	ഓ	0AC8
ഔ	0AA0	ഓ	0AC9
ഔ	0AA1	ഓ	0ACB
ഔ	0AA2	ഓ	0ACC
ഔ	0AA3	ഓ	0ACD
ഔ	0AA4	ഓ	0AD0
ഔ	0AA5	ഓ	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	ḃ	0253
y	01B4	ḋ	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
א	05D0	ב	05DE
ב	05D1	ג	05DF
ג	05D2	ד	05E0
ד	05D3	ה	05E1
ה	05D4	ו	05E2
ו	05D5	ז	05E3
ז	05D6	ח	05E4
ח	05D7	ט	05E5
ט	05D8	י	05E6
י	05D9	כ	05E7
ך	05DA	ל	05E8
ל	05DB	מ	05E9
מ	05DC	נ	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
ॆ	0901	ध	0927
.	0902	न	0928
:	0903	प	092A
अ	0905	फ	092B
आ	0906	ब	092C
इ	0907	भ	092D
ई	0908	म	092E
उ	0909	य	092F
ऊ	090A	र	0930
ऋ	090B	ल	0932
ए	090F	व	0935
ऐ	0910	श	0936
ऑ	0911	ष	0937
ओ	0913	स	0938
औ	0914	ह	0939
क	0915	ट	093E
ख	0916	डि	093F
ग	0917	डी	0940

문자	코드	문자	코드
घ	0918	ॆ	0941
ङ	0919	ॆ	0942
च	091A	ॆ	0943
छ	091B	ॆ	0945
ज	091C	ॆ	0947
झ	091D	ॆ	0948
ञ	091E	ॆ	0949
ट	091F	ॆ	094B
ठ	0920	ॆ	094C
ड	0921	ॆ	094D
ढ	0922	ज़	095B
ण	0923	ड	095C
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)

문자	매핑 대상
ḡ(095A)	ḡ(0917)
ḥ(095E)	ḥ(092B)
ḣ(095F)	ḣ(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
í	00ED	ü	00FC
ó	00F3	ő	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
೦	0C82	೧	0CA7
ಃ	0C83	೨	0CA8
಄	0C85	೩	0CAA
ಅ	0C86	೪	0CAB

문자	코드	문자	코드
ಇ	0C87	ಬ	0CAC
ಈ	0C88	ಫ	0CAD
ಉ	0C89	ಋ	0CAE
ಊ	0C8A	ಯ	0CAF
ಋ	0C8B	ರ	0CB0
ಎ	0C8E	ಲ	0CB2
ಏ	0C8F	ಳ	0CB3
ಐ	0C90	ಳ	0CB5
ಒ	0C92	಴	0CB6
ಓ	0C93	಴	0CB7
ಔ	0C94	ಽ	0CB8
ಕ	0C95	ಱ	0CB9
ಖ	0C96	#	0CBC
ಗ	0C97	#	0CBD
ಘ	0C98	ಃ	0CBE
ಙ	0C99	ಌ	0CBF
ಚ	0C9A	಍	0CC0
ಛ	0C9B	ಎ	0CC1
ಜ	0C9C	ಏ	0CC2
ಝ	0C9D	ಉ	0CC3

문자	코드	문자	코드
ᄀ	0C9E	ᄁ	0CC6
ᄂ	0C9F	ᄃ	0CC7
ᄄ	0CA0	ᄅ	0CC8
ᄆ	0CA1	ᄇ	0CCA
ᄈ	0CA2	ᄉ	0CCB
ᄊ	0CA3	ᄋ	0CCC
ᄌ	0CA4	ᄍ	0CCD
ᄎ	0CA5	ᄏ	0CD5
ᄐ	0CA6	ᄑ	0CD6
ᄒ	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
Ғ	0442	Ы	044B
Ғ	0431	Я	044F
О	043E	С	0441

문자	코드	문자	코드
п	043F	h	04BB
ш	0448	д	0434
и	0438	р	0440
ч	0447	г	0433
н	043D	ё	0451
ќ	049B	й	0439
і	0456	ө	04E9
щ	0449	в	0432
е	0435	э	044D
ө	04D9	ң	04A3
ю	044E	л	043B
з	0437	ф	0444
х	0445	к	043A
ц	0446	у	0443
ү	04AF	ж	0436
м	043C	ғ	0493
ь	044C	а	0430
ъ	044A	ұ	04B1

## 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
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	љ	0459
б	0431	њ	045A
в	0432	ћ	045B
г	0433	ќ	045C
д	0434	#	045D
е	0435	ђ	045E
ж	0436	џ	045F
з	0437	ѓ	0491
и	0438	ѣ	0493
й	0439	ж	0497
к	043A	ѕ	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

문자	코드	문자	코드
т	0442	ṭ	04AD
у	0443	Ƴ	04AF
ф	0444	ƴ	04B1
х	0445	ɣ	04B3
ц	0446	ɥ	04B5
ч	0447	ɥ	04B7
ш	0448	h	04BB
щ	0449	ɛ	04BD
ъ	044A	ɛ̃	04BF
ы	044B	#	04CA
ь	044C	ǎ	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



문자	코드	문자	코드
i	0456	ÿ	04F3
ï	0457	#	04F7
j	0458	ÿ	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
č	010D	ļ	013C
-	0113	ņ	0146
ģ	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	ı	012F
č	010D	š	0161
ę	0119	ı	0173
è	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
а	0430	љ	0459
б	0431	њ	045A
в	0432	ћ	045B
г	0433	ќ	045C
д	0434	#	045D
е	0435	ђ	045E
ж	0436	џ	045F
з	0437	ѓ	0491
и	0438	ѣ	0493
й	0439	ж	0497
к	043A	ѕ	0499
л	043B	ќ	049B
м	043C	к	049F
н	043D	т	04A1
о	043E	ћ	04A3
п	043F	н	04A5
р	0440	ѝ	04A9

문자	코드	문자	코드
с	0441	ç	04AB
т	0442	τ	04AD
у	0443	γ	04AF
φ	0444	ϕ	04B1
х	0445	χ	04B3
ц	0446	ц	04B5
ч	0447	ч	04B7
ш	0448	h	04BB
щ	0449	є	04BD
ъ	044A	ѐ	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

문자	코드	문자	코드
s	0455	ŷ	04F1
i	0456	ÿ	04F3
ï	0457	#	04F7
j	0458	ÿ	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
◌	0D02	ᵠ	0D28
ᵠ	0D03	ᵠ	0D2A

문자	코드	문자	코드
ආ	0D05	ඬ	0D2B
ඈ	0D06	ඹ	0D2C
ඉ	0D07	භ	0D2D
ඊ	0D08	ඪ	0D2E
උ	0D09	ඹ	0D2F
ඌ	0D0A	ඹ	0D30
ඍ	0D0B	ඹ	0D31
ඎ	0D0E	ඹ	0D32
ඏ	0D0F	ඹ	0D33
ඐ	0D10	ඹ	0D34
එ	0D12	ඹ	0D35
ඒ	0D13	ඹ	0D36
උ	0D14	ඹ	0D37
ඌ	0D15	ඹ	0D38
ඍ	0D16	ඹ	0D39
ඎ	0D17	ඹ	0D3E
ඏ	0D18	ඹ	0D3F
ඐ	0D19	ඹ	0D40
එ	0D1A	ඹ	0D41
ඒ	0D1B	ඹ	0D42

문자	코드	문자	코드
꺆	0D1C	꺇	0D43
꺈	0D1D	꺉	0D46
꺊	0D1E	꺋	0D47
꺌	0D1F	꺍	0D48
꺎	0D20	꺏	0D4A
꺑	0D21	꺒	0D4B
꺓	0D22	꺔	0D4C
꺕	0D23	꺖	0D4D
꺗	0D24	#	0D7A
꺘	0D25	#	0D7B
꺙	0D26	#	0D7C
꺚	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

문자	코드	문자	코드
è	00E8	ć	010B
ì	00EC	ġ	0121
ò	00F2	ħ	0127
ž	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
ॆ	0901	थ	0925
ॆ	0902	द	0926
:	0903	ध	0927
अ	0905	न	0928
आ	0906	प	092A
इ	0907	फ	092B
ई	0908	ब	092C
उ	0909	भ	092D
ऊ	090A	म	092E



문자	코드	문자	코드
ऋ	090B	य	092F
ँ	090D	र	0930
ए	090F	ल	0932
ऐ	0910	ळ	0933
ऑ	0911	व	0935
ओ	0913	श	0936
औ	0914	ष	0937
क	0915	स	0938
ख	0916	ह	0939
ग	0917	.	093C
घ	0918	।	093E
ङ	0919	ि	093F
च	091A	ी	0940
छ	091B	ु	0941
ज	091C	ू	0942
झ	091D	े	0943
ञ	091E	ै	0945
ट	091F	ो	0947
ठ	0920	ौ	0948
ड	0921	ँ	0949

문자	코드	문자	코드
ढ	0922	ी	094B
ण	0923	ु	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
в	0432	ћ	045B
г	0433	ќ	045C
д	0434	#	045D
е	0435	ђ	045E
ж	0436	џ	045F
з	0437	ѓ	0491
и	0438	џ	0493

문자	코드	문자	코드
й	0439	ж	0497
к	043A	з	0499
л	043B	к	049B
м	043C	к	049F
н	043D	к	04A1
о	043E	ң	04A3
п	043F	н	04A5
р	0440	œ	04A9
с	0441	ç	04AB
т	0442	т	04AD
у	0443	ү	04AF
ф	0444	ұ	04B1
х	0445	х	04B3
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

## 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

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	џ	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	ђ	04AD
у	0443	у	04AF
ф	0444	у	04B1

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

## 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
ୱ	0B01	ଠ	0B26
୲	0B02	ଢ	0B27
୳	0B03	ଢ	0B28
୴	0B05	ଢ	0B2A
୵	0B06	ଢ	0B2B

문자	코드	문자	코드
ඹ	0B07	ඹ	0B2C
ඹ	0B08	ඹ	0B2D
ඹ	0B09	ඹ	0B2E
ඹ	0B0A	ඹ	0B2F
ඹ	0B0B	ඹ	0B30
ඹ	0B0F	ඹ	0B32
ඹ	0B10	ඹ	0B33
ඹ	0B13	ඹ	0B36
ඹ	0B14	ඹ	0B37
ඹ	0B15	ඹ	0B38
ඹ	0B16	ඹ	0B39
ඹ	0B17	.	0B3C
ඹ	0B18		0B3E
ඹ	0B19	ˆ	0B3F
ඹ	0B1A	1	0B40
ඹ	0B1B	2	0B41
ඹ	0B1C	3	0B42
ඹ	0B1D	4	0B43
ඹ	0B1E	5	0B47
ඹ	0B1F	6	0B48



문자	코드	문자	코드
0	0B20	6#	0B4B
0	0B21	6	0B4C
0	0B22	~	0B4D
6	0B23	~	0B56
0	0B24	0	0B5F
2	0B25	0	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	ﻻ	064A
ؤ	0624	=	064B
ئ	0626	ء	064C
ل	0627	=	064D
ب	0628	ـ	064E

문자	코드	문자	코드
ت	062A	ٲ	064F
ث	062B	ٲ	0650
ج	062C	ٳ	0651
ح	062D	ٴ	0652
خ	062E	#	0654
د	062F	ٲ	0670
ذ	0630	ٲ	067C
ر	0631	ٲ	067E
ز	0632	ٲ	0681
س	0633	ٲ	0685
ش	0634	ج	0686
ص	0635	ٲ	0689
ض	0636	ٲ	0693
ط	0637	ٲ	0696
ظ	0638	ژ	0698
ع	0639	بن	069A
غ	063A	ک	06A9
ف	0641	گ	06AB
ق	0642	گ	06AF
ل	0644	ن	06BC

문자	코드	문자	코드
ρ	0645	ς	06CC
ϋ	0646	Ϙ	06CD
ο	0647	ϙ	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
ę	0119	ź	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
Ë	00CB	ë	00EB
Í	00CD	í	00ED
Ñ	00D1	ñ	00F1
Ó	00D3	ó	00F3
Ô	00D4	ô	00F4
Õ	00D5	õ	00F5
Ö	00D6	ö	00F6
Ú	00DA	ú	00FA

문자	코드	문자	코드
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
ਅ	0A05	ਧ	0A27
ਆ	0A06	ਠ	0A28
ਇ	0A07	ਪ	0A2A
ਈ	0A08	ਫ	0A2B
ਉ	0A09	ਬ	0A2C
ਊ	0A0A	ਭ	0A2D
ਏ	0A0F	ਮ	0A2E
ਐ	0A10	ਯ	0A2F
ੳ	0A13	ਰ	0A30
ੴ	0A14	ਲ	0A32
ਕ	0A15	ਵ	0A35
ਖ	0A16	ਸ	0A38

문자	코드	문자	코드
ᄀ	0A17	ᄁ	0A39
ᄂ	0A18	.	0A3C
ᄃ	0A19	ᄄ	0A3E
ᄅ	0A1A	ᄆ	0A3F
ᄇ	0A1B	ᄈ	0A40
ᄉ	0A1C	ᄊ	0A41
ᄋ	0A1D	ᄌ	0A42
ᄍ	0A1E	ᄎ	0A47
ᄏ	0A1F	ᄐ	0A48
ᄑ	0A20	ᄒ	0A4B
ᄓ	0A21	ᄔ	0A4C
ᄕ	0A22	ᄖ	0A4D
ᄗ	0A23	ᄘ	0A5C
ᄙ	0A24	ᄚ	0A70
ᄛ	0A25	ᄜ	0A71
ᄝ	0A26	ᄞ	0A72
ᄟ	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	ş	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

문자	코드	문자	코드
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



문자	코드	문자	코드
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
л	043B	т	04A1
м	043C	ћ	04A3
н	043D	н	04A5
о	043E	џ	04A9
п	043F	џ	04AB
р	0440	џ	04AD
с	0441	џ	04AF
т	0442	џ	04B1

문자	코드	문자	코드
y	0443	ɣ	04B3
φ	0444	ц	04B5
x	0445	ყ	04B7
ц	0446	h	04BB
ч	0447	є	04BD
ш	0448	ę	04BF
щ	0449	#	04CA
ъ	044A	ă	04D1
ы	044B	ä	04D3
ь	044C	ě	04D7
э	044D	ə	04D9
ю	044E	з	04E1
я	044F	й	04E3
#	0450	ö	04E7
ë	0451	ө	04E9
ђ	0452	ȳ	04EF
í	0453	ÿ	04F1
є	0454	ÿ	04F3
s	0455	#	04F7
ÿ	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
#	0D88	#	0DB5
#	0D89	#	0DB6
#	0D8A	#	0DB7
#	0D8B	#	0DB8
#	0D8C	#	0DB9
#	0D8D	#	0DBA
#	0D91	#	0DB
#	0D92	#	0DBD
#	0D93	#	0DC0

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

문자	코드	문자	코드
#	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
ä	00E4	ó	00F3
č	010D	ô	00F4
d'	010F	í	0155
é	00E9	š	0161
í	00ED	td	0165
í	013A	ú	00FA
ř	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
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
ஏ	0B85	ஈ	0BB0

문자	코드	문자	코드
ஆ	0B86	ல	0BB2
இ	0B87	வ	0BB5
ஈ	0B88	ழ	0BB4
உ	0B89	ள	0BB3
ஊ	0B8A	ற	0BB1
எ	0B8E	ன	0BA9
ஏ	0B8F	ஐ	0B9C
ஐ	0B90	#	0BB6
ஒ	0B92	ஷ	0BB7
ஓ	0B93	ஸ	0BB8
ஔ	0B94	ஹ	0BB9
ஐ	0B83	.	0BCD
க	0B95	ஈ	0BBE
ங	0B99	ற	0BBF
ச	0B9A	ஶ	0BC0
ஞ	0B9E	ழ	0BC1
ட	0B9F	ஶ	0BC2
ண	0BA3	ஶ	0BC6
த	0BA4	ஶ	0BC7
ந	0BA8	ஶ	0BC8

문자	코드	문자	코드
Ɑ	0BAA	ᄀᆞ	0BCA
ᄁ	0BAE	ᄁᆞ	0BCB
ᄂ	0BAF	ᄂᆞ	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	Ӏ	0459
б	0431	ӑ	045A
в	0432	ӕ	045B
г	0433	ӗ	045C
д	0434	ӝ	045D
е	0435	ӧ	045E
ж	0436	ӡ	045F
з	0437	Ӣ	0491
и	0438	Ӥ	0493
й	0439	Ӧ	0497

문자	코드	문자	코드
к	043A	ꞑ	0499
л	043B	ќ	049B
м	043C	κ	049F
н	043D	κ	04A1
о	043E	ҥ	04A3
п	043F	н	04A5
р	0440	œ	04A9
с	0441	ç	04AB
т	0442	т	04AD
у	0443	γ	04AF
ф	0444	ƴ	04B1
х	0445	χ	04B3
ц	0446	ц	04B5
ч	0447	ч	04B7
ш	0448	h	04BB
щ	0449	є	04BD
ъ	044A	є	04BF
ы	044B	#	04CA
ь	044C	ă	04D1
э	044D	ä	04D3

문자	코드	문자	코드
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

## 텔루구어 문자 집합

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

문자	코드	문자	코드
-	002D	ఱ	0C24
ఁ	0C01	ఢ	0C25
ఌ	0C02	ఘ	0C26
఍	0C03	ఞ	0C27
ఐ	0C05	ఠ	0C28

문자	코드	문자	코드
ఆ	0C06	ప	0C2A
ఇ	0C07	భ	0C2B
ఈ	0C08	బ	0C2C
ఉ	0C09	భ	0C2D
ఊ	0C0A	మ	0C2E
ఋ	0C0B	య	0C2F
ఠ	0C30	ఎ	0C0E
ఐ	0C31	ఏ	0C0F
ఊ	0C32	ఐ	0C10
ఋ	0C33	ఒ	0C12
ౠ	0C35	ఓ	0C13
ౡ	0C36	ఔ	0C14
ౢ	0C37	క	0C15
ౣ	0C38	ఖ	0C16
౤	0C39	గ	0C17
౥	0C3E	ఘ	0C18
౦	0C3F	ఙ	0C19
౧	0C40	చ	0C1A
౨	0C41	ఛ	0C1B
౩	0C42	జ	0C1C

문자	코드	문자	코드
๒	0C43	๓	0C1D
๔	0C44	๕	0C1E
๖	0C47	๗	0C1F
๘	0C48	๙	0C20
๐	0C4A	๑	0C21
๒	0C4B	๓	0C22
๔	0C4C	๕	0C23
๖	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
๐	0E01	๑	0E25
๒	0E02	๓	0E26
๔	0E03	๕	0E27
๖	0E04	๗	0E28
๘	0E05	๙	0E29

문자	코드	문자	코드
ฃ	0E06	ส	0E2A
ง	0E07	ห	0E2B
จ	0E08	ฬ	0E2C
ฉ	0E09	อ	0E2D
ช	0E0A	ษ	0E2E
ซ	0E0B	ะ	0E2F
ฌ	0E0C	เ	0E30
ญ	0E0D	อ	0E31
ฎ	0E0E	า	0E32
ฏ	0E0F	า	0E34
ฐ	0E10	า	0E35
ฑ	0E11	า	0E36
ฒ	0E12	า	0E37
ณ	0E13	,	0E38
ด	0E14	ุ	0E39
ต	0E15	.	0E3A
ถ	0E16	เ	0E40
ท	0E17	แ	0E41
ธ	0E18	โ	0E42
น	0E19	ใ	0E43



문자	코드	문자	코드
ı	0E1A	ı	0E44
ı	0E1B	ı	0E45
ı	0E1C	ı	0E46
ı	0E1D	ı	0E47
ı	0E1E	'	0E48
ı	0E1F	ı	0E49
ı	0E20	ı	0E4A
ı	0E21	ı	0E4B
ı	0E22	ı	0E4C
ı	0E23	ı	0E4D
ı	0E24		

## 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	ı	0130
è	00E8	İ	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
a	0430	p	0440
б	0431	c	0441
в	0432	т	0442
г	0433	y	0443
д	0434	ф	0444
е	0435	x	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	є	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	ء	064C
أ	0623	=	064D
ؤ	0624	ـ	064E
إ	0625	ء	064F
ئ	0626	ـ	0650
ا	0627	ء	0651
ب	0628	ء	0652
ة	0629	#	0653
ت	062A	#	0654
ث	062B	#	0657
ج	062C	ـ	0670
ح	062D	ط	0679
خ	062E	ن	067A

문자	코드	문자	코드
د	062F	ب	067B
ذ	0630	ت	067C
ر	0631	ث	067D
ز	0632	پ	067E
س	0633	ش	067F
ش	0634	پ	0680
ص	0635	ح	0681
ض	0636	ج	0683
ط	0637	ج	0684
ظ	0638	خ	0685
ع	0639	ج	0686
غ	063A	چ	0687
-	0640	ڈ	0688
ف	0641	د	0689
ق	0642	د	068A
ك	0643	ذ	068C
ل	0644	د	068D
م	0645	ذ	068F
ن	0646	ژ	0691
ه	0647	ر	0693

문자	코드	문자	코드
گ	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
з	0437	л	043B

문자	코드	문자	코드
x	0445	ɸ	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
ú	00FA	Ú	00DA



문자	코드	문자	코드
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
ã	1EA5	Ã	1EA4
ǎ	1EA7	Ǎ	1EA6
ǻ	1EA9	Ǻ	1EA8
ǿ	1EAB	Ǿ	1EAA
ậ	1EAD	Ậ	1EAC
ắ	1EAF	Ắ	1EAE
ǎ	1EB1	Ǎ	1EB0
ǻ	1EB3	Ǻ	1EB2
ǿ	1EB5	Ǿ	1EB4
ắ	1EB7	Ắ	1EB6
ẹ	1EB9	Ẹ	1EB8

문자	코드	문자	코드
ẻ	1EBB	Ễ	1EBA
ẽ	1EBD	Ễ	1EBC
ế	1EBF	Ễ	1EBE
ề	1EC1	Ề	1EC0
ễ	1EC3	Ễ	1EC2
ễ	1EC5	Ễ	1EC4
ệ	1EC7	Ệ	1EC6
ỉ	1EC9	Ỉ	1EC8
ị	1ECB	Ị	1ECA
ọ	1ECD	Ọ	1ECC
ỏ	1ECF	Ỏ	1ECE
ố	1ED1	Ổ	1ED0
ồ	1ED3	Ồ	1ED2
ỗ	1ED5	Ổ	1ED4
ỗ	1ED7	Ổ	1ED6
ộ	1ED9	Ộ	1ED8
ớ	1EDB	Ớ	1EDA
ờ	1EDD	Ờ	1EDC
ở	1EDF	Ở	1EDE
ỡ	1EE1	Ở	1EE0

문자	코드	문자	코드
ø	1EE3	Ø	1EE2
ұ	1EE5	Ҫ	1EE4
ǔ	1EE7	Ů	1EE6
ú	1EE9	Ú	1EE8
ù	1EEB	Ù	1EEA
ű	1EED	Ű	1EEC
ű	1EEF	Ű	1EEE
ұ	1EF1	Ҫ	1EF0
ỳ	1EF3	Ỳ	1EF2
ʏ	1EF5	Ỳ	1EF4
ÿ	1EF7	Ỳ	1EF6
ÿ	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

문자	코드	문자	코드
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
ỳ	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

문자	코드	문자	코드
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 방식

Amazon Transcribe 머신 러닝 모델을 사용하여 음성을 텍스트로 변환합니다.

트랜스크립트에는 트랜스크립트된 텍스트 외에도 각 단어 또는 구두점에 대한 신뢰도 점수 및 타임스탬프를 포함하여 필수된 내용에 대한 데이터가 포함됩니다. 출력 예제를 보려면 [데이터 입력 및 출력](#) 섹션을 참조하십시오. 트랜스크립션에 적용할 수 있는 전체 기능 목록은 [기능 요약](#)을 참조하십시오.


전사 방법은 두 가지 주요 범주로 나뉩니다.

- Batch 트랜스크립션: Amazon S3 버킷에 업로드된 미디어 파일을 트랜스크립션합니다. [AWS CLI](#) [AWS Management Console](#), 및 다양한 [AWS SDK](#)를 사용하여 일괄 트랜스크립션을 작성할 수 있습니다.
- 스트리밍 트랜스크립션: 미디어 스트림을 실시간으로 트랜스크립션합니다. [HTTP/2](#) 및 다양한 [AWS SDK](#)를 사용하여 트랜스크립션을 스트리밍할 수 있습니다. [AWS Management Console](#) [WebSockets](#)

일괄 처리 및 스트리밍 트랜스크립션의 기능 및 언어 지원은 다릅니다. 자세한 내용은 [Amazon Transcribe features](#) 및 [지원되는 언어](#)를 참조하십시오.

주제

- [데이터 입력 및 출력](#)
- [숫자 표기 및 문장 부호](#)

 시작하는 데 도움이 될 API 작업

Batch: [StartTranscriptionJob](#)

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

## 데이터 입력 및 출력

Amazon Transcribe 오디오 데이터를 Amazon S3 버킷 또는 미디어 스트림의 미디어 파일로 가져와서 텍스트 데이터로 변환합니다.

Amazon S3버킷에 저장된 미디어 파일을 트랜스크립션하는 경우 일괄 트랜스크립션을 수행하는 것입니다. 미디어 스트림을 트랜스크립션하는 경우 스트리밍 트랜스크립션을 수행하는 것입니다. 이 두 프로세스에는 서로 다른 규칙과 요구 사항이 있습니다.

일괄 트랜스크립션을 사용하면 모든 트랜스크립션 작업을 동시에 처리할 필요가 없는 [Job 대기열](#) 경우 사용할 수 있습니다. 이렇게 하면 Amazon Transcribe 트랜스크립션 작업을 추적하고 슬롯이 있을 때 처리할 수 있습니다.

### Note

Amazon Transcribe 분석 모델의 품질을 지속적으로 개선하기 위해 콘텐츠를 임시로 저장할 수 있습니다. 자세한 정보는 [Amazon Transcribe FAQ](#)를 참조하십시오. 에 의해 Amazon Transcribe 저장되었을 수 있는 콘텐츠의 삭제를 요청하려면 로 케이스를 [AWS Support](#) 여십시오.

## 주제

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

## 미디어 형식

지원되는 미디어 유형은 일괄 트랜스크립션과 스트리밍 트랜스크립션 간에 다르지만 무손실 형식이 모두 권장됩니다. 세부 정보는 다음 표를 참조하세요.

	Batch	스트리밍
지원되는 형식	<ul style="list-style-type: none"> <li>• AMR</li> <li>• FLAC</li> <li>• M4A</li> <li>• MP3</li> <li>• MP4</li> <li>• 오그</li> </ul>	<ul style="list-style-type: none"> <li>• FLAC</li> <li>• 오그 오픈스</li> <li>• PCM 인코딩</li> </ul>

	Batch	스트리밍
	<ul style="list-style-type: none"> <li>• WebM</li> <li>• WAV</li> </ul>	
권장 포맷	<ul style="list-style-type: none"> <li>• FLAC</li> <li>• PCM 16비트 인코딩을 지원하는 WAV</li> </ul>	<ul style="list-style-type: none"> <li>• FLAC</li> <li>• PCM 서명 16비트 리틀 엔디안 오디오 (WAV는 포함되지 않음)</li> </ul>

최상의 결과를 얻으려면 PCM 16비트 인코딩이 포함된 FLAC 또는 WAV와 같은 무손실 형식을 사용하십시오.

#### Note

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

## 오디오 채널

Amazon Transcribe 단일 채널 및 이중 채널 미디어를 지원합니다. 채널이 2개 이상인 미디어는 현재 지원되지 않습니다.

오디오의 한 채널에 여러 스피커가 포함되어 있고 트랜스크립션 출력에서 각 스피커를 파티셔닝하고 레이블을 지정하려는 경우 [스피커 파티셔닝 \(일자리화\)](#) 을 사용할 수 있습니다.

오디오에 별도의 두 채널의 음성이 포함된 경우 [채널 식별](#) 을 사용하여 트랜스크립트에서 각 채널을 개별적으로 텍스트로 변환할 수 있습니다.

이 두 옵션 모두 하나의 트랜스크립트 파일을 생성합니다.

#### Note

[스피커 파티셔닝](#) 또는 [채널 식별](#) 을 활성화하지 않은 경우 트랜스크립트 텍스트는 하나의 연속 섹션으로 제공됩니다.



## 샘플 속도

일괄 트랜스크립션 작업의 경우 샘플 속도를 제공하도록 선택할 수 있지만 이 매개 변수는 선택 사항입니다. 요청에 이를 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. 일괄 트랜스크립션 작업과 마찬가지로 제공하는 값이 오디오의 실제 샘플 속도와 일치하는지 확인하세요.

전화 녹음과 같은 저음질 오디오의 샘플 레이트는 일반적으로 8,000Hz를 사용합니다. 하이파이 오디오의 경우 16,000Hz에서 48,000Hz 사이의 값을 Amazon Transcribe 지원합니다.

## 출력

트랜스크립션 출력 형식은 JSON 형식입니다. 성적표의 첫 부분에는 단락 형식의 성적 증명서 자체가 포함되며, 그 뒤에 모든 단어와 문장 부호에 대한 추가 데이터가 포함됩니다. 제공되는 데이터는 요청에 포함된 기능에 따라 달라집니다. 성적표에는 최소한 모든 단어의 시작 시간, 종료 시간 및 신뢰도 점수가 포함됩니다. [다음 섹션은](#) 추가 옵션이나 기능을 포함하지 않은 기본 트랜스크립션 요청의 예제 출력을 보여줍니다.

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

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

Amazon S3 버킷을 지정하지 않는 경우 보안 서비스 관리 버킷을 Amazon Transcribe 사용하고 트랜스크립트를 다운로드하는 데 사용할 수 있는 임시 URI를 제공합니다. 임시 URI는 15분간 유효합니다. 제공된 URI를 사용할 때 AccessDenied 오류가 발생하면 트랜스크립트의 새 임시 URI를 GetTranscriptionJob 요청하세요.

기본 버킷을 선택하면 작업이 완료될 때 (90일) 트랜스크립트가 삭제됩니다. 이 만료일을 초과하여 성적표를 보관하려면 반드시 다운로드해야 합니다.

스트리밍 트랜스크립트는 스트림에 사용하는 것과 동일한 방법을 통해 반환됩니다.

**i** Tip

JSON 출력을 Word 형식의 turn-by-turn 트랜스크립트로 변환하려면 이 [GitHub 예제 \(Python3 용\)](#) 를 참조하십시오. 이 스크립트는 일괄화가 활성화된 통화 후 분석 트랜스크립트와 표준 배치 트랜스크립트와 함께 작동합니다.

**출력 예**

트랜스크립트는 단락 형식으로 완전한 필사본을 제공하며, 그 다음에는 모든 단어와 문장 부호에 대한 데이터를 제공하는 word-for-word 세분화가 이어집니다. 여기에는 시작 시간, 종료 시간, 신뢰도 점수 및 유형 (pronunciation 또는 punctuation) 이 포함됩니다.

다음 예제는 [추가 기능을](#) 포함하지 않은 간단한 일괄 트랜스크립션 작업에서 가져온 것입니다. 트랜스크립션 요청에 각 추가 기능을 적용할 때마다 트랜스크립트 출력 파일에 추가 데이터가 제공됩니다.

기본 배치 트랜스크립트에는 두 가지 주요 섹션이 있습니다.

1. transcripts: 전체 스크립트를 하나의 텍스트 블록에 포함합니다.
2. items: transcripts 섹션의 각 단어와 문장 부호에 대한 정보가 들어 있습니다.

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": [
      {
        "start_time": "0.64",
        "end_time": "1.09",
        "alternatives": [
          {
            "confidence": "1.0",
            "content": "Welcome"
          }
        ]
      }
    ]
  }
}
```

```
    ],
    "type": "pronunciation"
  },
  {
    "start_time": "1.09",
    "end_time": "1.21",
    "alternatives": [
      {
        "confidence": "1.0",
        "content": "to"
      }
    ],
    "type": "pronunciation"
  },
  {
    "start_time": "1.21",
    "end_time": "1.74",
    "alternatives": [
      {
        "confidence": "1.0",
        "content": "Amazon"
      }
    ],
    "type": "pronunciation"
  },
  {
    "start_time": "1.74",
    "end_time": "2.56",
    "alternatives": [
      {
        "confidence": "1.0",
        "content": "Transcribe"
      }
    ],
    "type": "pronunciation"
  },
  {
    "alternatives": [
      {
        "confidence": "0.0",
        "content": "."
      }
    ],
    "type": "punctuation"
  }
}
```

```

    }
  ],
},
"status": "COMPLETED"
}

```

## 숫자 표기 및 문장 부호

Amazon Transcribe 지원되는 모든 언어에 자동으로 구두점을 추가하고 표기법에서 대/소문자 구분을 사용하는 언어에 맞게 단어를 대문자로 표시합니다.

대부분의 언어에서 숫자는 단어 형태로 변환됩니다. 그러나 미디어가 영어나 독일어로 되어 있는 경우 숫자를 사용하는 상황에 따라 숫자를 다르게 Amazon Transcribe 취급합니다.

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 번역됩니다.

- 영어 및 독일어 방언: 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

영어와 독일어로 된 음성 번호와 관련된 모든 규칙을 보려면 다음 표를 참조하십시오.

규칙	영어 방언 (오디오 입력 → 텍스트 출력)	독일어 방언 (오디오 입력 → 텍스트 출력)
10보다 큰 기수를 숫자로 변환합니다.	<ul style="list-style-type: none"> <li>• "Fifty five" → 55</li> <li>• "a hundred" → 100</li> <li>• "One thousand and thirty one" → 1031</li> <li>• "One hundred twenty-three million four hundred fifty six thousand seven hundred eight nine" → 123,456,789</li> </ul>	<ul style="list-style-type: none"> <li>• "fünfundfünfzig" → 55</li> <li>• "vier tausend sechs hundert einundachtzig" → 4681</li> <li>• "eine Sache" → "eine Sache"</li> </ul>

규칙	영어 방언 (오디오 입력 → 텍스트 출력)	독일어 방언 (오디오 입력 → 텍스트 출력)
“million” 또는 “billion” 뒤에 숫자가 오지 않는 경우 뒤에 “million” 또는 “billion”이 있는 기수를 숫자로 변환합니다.	<ul style="list-style-type: none"> <li>• "one hundred million" → 100 million</li> <li>• "one billion" → 1 billion</li> <li>• "two point three million" → 2.3 million</li> </ul>	<ul style="list-style-type: none"> <li>• "zehn Millionen Menschen" → 10 Millionen Menschen</li> <li>• "zehn Millionen fünf hundert tausend" → 10.500.000</li> </ul>
10보다 큰 서수를 숫자로 변환합니다.	<ul style="list-style-type: none"> <li>• "Forty third" → 43rd</li> <li>• "twenty sixth avenue" → 26th avenue</li> </ul>	<ul style="list-style-type: none"> <li>• "dreiundzwanzigste" → 23</li> <li>• "vierzigster" → 40</li> <li>• "ich war Erster" → "ich war Erster"</li> </ul>
분수를 숫자 형식으로 변환합니다.	<ul style="list-style-type: none"> <li>• "a quarter" → 1/4</li> <li>• "three sixteenths" → 3/16</li> <li>• "a half" → 1/2</li> <li>• "a hundredth" → 1/100</li> </ul>	<p>분수는 숫자 형식으로 변환되지 않습니다.</p> <ul style="list-style-type: none"> <li>• "ein Drittel" → "ein Drittel"</li> </ul>
한 행에 둘 이상의 숫자가 있는 경우 10보다 작은 숫자를 숫자로 변환합니다.	<ul style="list-style-type: none"> <li>• "three four five" → 345</li> <li>• "My phone number is four two five five five five one two one two" → My phone number is 4255551212</li> </ul>	<ul style="list-style-type: none"> <li>• "eins zwei drei" → 123</li> <li>• "plus vier neun zwei vier eins" → +49241</li> </ul>
“점” 또는 “점”이라는 단어는 소수점으로 표시됩니다.	<ul style="list-style-type: none"> <li>• "three hundred and three dot five" → 303.5</li> <li>• "three point twenty three" → 3.23</li> <li>• "zero point four" → 0.4</li> <li>• "point three" → 0.3</li> </ul>	<p>소수점은 “,”로 표시됩니다.</p> <ul style="list-style-type: none"> <li>• "zweiundzwanzig komma drei" → 22,3</li> </ul>

규칙	영어 방언 (오디오 입력 → 텍스트 출력)	독일어 방언 (오디오 입력 → 텍스트 출력)
숫자 뒤의 "percent" 단어를 백분을 기호(%)로 변환합니다.	<ul style="list-style-type: none"> <li>• "twenty three percent" → 23%</li> <li>• "twenty three point four five percent" → 23.45%</li> </ul>	<ul style="list-style-type: none"> <li>• "fünf Prozent Hürde" → 5% Hürde</li> <li>• "dreiundzwanzig komma vier Prozent" → 23,4%</li> </ul>

규칙	영어 방언 (오디오 입력 → 텍스트 출력)	독일어 방언 (오디오 입력 → 텍스트 출력)
<p>화폐 단어를 기호로 변환합니다.</p>	<p>숫자 뒤의 “달러”, “미국 달러”, “호주 달러”, “AUD” 또는 “USD”라는 단어를 숫자 앞의 달러 기호 (\$) 로 변환합니다.</p> <ul style="list-style-type: none"> <li>• "one dollar and fifteen cents" → \$1.15</li> <li>• "twenty three USD" → \$23</li> <li>• "twenty three Australian dollars" → \$23</li> </ul> <p>숫자 뒤의 "pounds," "British pounds" 또는 "GDB" 단어를 숫자 앞의 파운드 기호 (£)로 변환합니다.</p> <ul style="list-style-type: none"> <li>• "twenty three pounds" → £23</li> <li>• "I have two thousand pounds" → I have £2,000</li> <li>• "five pounds thirty three pence" → £5.33</li> </ul> <p>숫자 뒤의 "rupees," "Indian rupees" 또는 "INR" 단어를 숫자 앞의 루피 기호(#)로 변환합니다.</p> <ul style="list-style-type: none"> <li>• "twenty three rupees" → #23</li> <li>• "fifty rupees thirty paise" → #50.30</li> </ul>	<p>“Euro”라는 단어를 유로 기호로 변환합니다.</p> <ul style="list-style-type: none"> <li>• "ein euro" → 1 €</li> <li>• "ein Euro vierzig" → 1,40 €</li> <li>• "ein Euro vierzig Cent" → 1,40 €</li> </ul>

규칙	영어 방언 (오디오 입력 → 텍스트 출력)	독일어 방언 (오디오 입력 → 텍스트 출력)
시간을 숫자로 변환합니다.	<ul style="list-style-type: none"> <li>• "seven a m eastern standard time" → 7 a.m. eastern standard time</li> <li>• "twelve thirty p m" → 12:30 p.m.</li> </ul>	<ul style="list-style-type: none"> <li>• "vierzehn Uhr fünfzehn" → 14:15 Uhr</li> </ul>
날짜를 숫자로 변환합니다.	<ul style="list-style-type: none"> <li>• "May fifth twenty twelve" → May 5th 2012</li> <li>• "May five twenty twelve" → May 5 2012</li> <li>• "five May twenty twelve" → 5 May 2012</li> </ul>	<ul style="list-style-type: none"> <li>• "dritter Dezember neunzehn hundert sechundfünzig" → 3. Dezember 1956</li> </ul>
숫자 범위를 “to”라는 단어로 구분합니다.	<ul style="list-style-type: none"> <li>• "twenty three to thirty seven" → 23 to 37</li> </ul>	해당 사항 없음
연도는 네 자리 숫자로 표시되며 20세기, 21세기 및 22세기의 연도에만 유효합니다.	<ul style="list-style-type: none"> <li>• "nineteen sixty two" → 1962</li> <li>• "the year is twenty twelve" → 연도는2012</li> <li>• "twenty nineteen" → 2019</li> <li>• "twenty one thirty" → 2130</li> </ul>	해당 사항 없음
슬래시와 대시를 표시합니다.	<ul style="list-style-type: none"> <li>• "fifty-five dash thirteen" → 55-13</li> </ul> <p>슬래시는 표시되지 않습니다.</p> <ul style="list-style-type: none"> <li>• "fifty-five slash thirteen" → 55 slash 13</li> </ul>	<ul style="list-style-type: none"> <li>• "fünfundfünzig Schrägstrich dreizehn" → 55/13</li> <li>• "fünfundfünzig Strich dreizehn" → 55-13</li> </ul>



규칙	영어 방언 (오디오 입력 → 텍스트 출력)	독일어 방언 (오디오 입력 → 텍스트 출력)
번호가 매겨진 단락을 표시합니다.	<p>번호가 매겨진 단락은 단락 기호 (§) 를 사용하여 표시되지 않습니다.</p> <ul style="list-style-type: none"> <li>"paragraph seventeen" → paragraph 17</li> </ul>	<ul style="list-style-type: none"> <li>"Paragraf siebzehn" → § 17</li> </ul>

# Amazon Transcribe 시작하기

트랜스크립션을 만들려면 먼저 몇 가지 전제 조건을 충족해야 합니다.

- [여기](#) 가입합니다.AWS 계정
- [AWS CLI 및 SDK 설치](#) (AWS Management Console 트랜스크립션을 사용하는 경우 이 단계를 건너뛰어도 됨)
- [IAM 자격 증명 구성](#)
- [Amazon S3 버킷 설정](#)
- [IAM 정책 생성](#)

이러한 전제 조건을 완료하면 트랜스크립션을 작성할 준비가 된 것입니다. 시작하려면 다음 목록에서 원하는 트랜스크립션 방법을 선택하세요.

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

## Tip

기능을 처음 Amazon Transcribe 사용하거나 살펴보려는 경우 [여기](#) 사용하는 것이 좋습니다 [AWS Management Console](#). 컴퓨터 마이크를 사용하여 스트리밍을 시작하려는 경우 가장 쉬운 옵션이기도 합니다.

HTTP/2를 사용한 WebSockets 스트리밍은 다른 트랜스크립션 방법보다 복잡하므로 이러한 방법을 시작하기 전에 [스트리밍 트랜스크립션 설정](#) 섹션을 검토하는 것이 좋습니다. 트랜스크립션 스트리밍에는 SDK를 사용하는 것이 좋습니다.

## 여기 가입 AWS 계정

[프리 티어](#) 계정 또는 [유료 계정에](#) 가입할 수 있습니다. 두 옵션 모두 모든 항목에 액세스할 수 있는 AWS 서비스 있습니다. 프리 티어에는 평가판 기간이 있으며, 이 기간 동안 사용량을 살펴보고 AWS 서비스 추

정할 수 있습니다. 평가판 기간이 만료되면 유료 계정으로 마이그레이션할 수 있습니다. 수수료는 pay-as-you-use 기준으로 부과됩니다. 자세한 내용은 [Amazon Transcribe요금](#)을 참조하십시오.

### Tip

계정을 설정할 때 AWS 계정 ID를 기록해 두세요. IAM 엔티티를 만들려면 ID가 필요하기 때문입니다.

## AWS CLI 및 SDK 설치

Amazon Transcribe API를 사용하려면 먼저 를 설치해야 합니다 AWS CLI. 현재 버전은 AWS CLI 2입니다. [AWS Command Line Interface 사용자 가이드](#)에서 [리눅스](#), [맥](#), [윈도우](#) 및 [도커](#)에 대한 설치 지침을 찾을 수 있습니다.

AWS CLI 설치가 완료되면 보안 자격 증명 및 에 [맞게 구성해야](#) AWS 리전 합니다.

Amazon Transcribe SDK와 함께 사용하려는 경우 설치 지침에 사용할 기본 언어를 선택하십시오.

- [.NET](#)
- [C++](#)
- [Go](#)
- [자바 V2](#)
- [JavaScript](#)
- [SDHP V3 V3](#)
- [AWS SDK for Python \(Boto3\)](#)(일괄 트랜스크립션)
- [파이썬](#) (스트리밍 트랜스크립션)
- [Ruby V3 V3](#)
- [Rust](#) (일괄 트랜스크립션)
- [Rust](#) (스트리밍 트랜스크립션)

## IAM 자격 증명 구성

을 생성할 때는 계정의 모든 AWS 서비스 및 리소스에 대한 완전한 액세스 권한이 있는 단일 로그인 ID로 시작합니다. AWS 계정 이 자격 증명은 AWS 계정 루트 사용자라고 하며, 계정을 생성할 때 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 사용자가 자격 증명 공급자와의 페더레이션을 사용하여 임시 보안 인증으로 에AWS 액세스하도록 요구합니다.

페더레이션형 ID는 ID 소스를 통해 제공된 보안 인증으로 제공된 보안 인증으로 에AWS 액세스합니다. 연동 자격 증명은 AWS 계정에 액세스할 때 역할을 수임하고 역할은 임시 자격 증명을 제공합니다.

중앙 집중식 액세스 관리를 위해 [AWS IAM Identity Center](#)를 사용하는 것이 좋습니다. 에서 사용자 및 그룹을 생성할 수IAM Identity Center 있습니다. 또는 자체 자격 증명 소스의 사용자 및 그룹 집합에 연결하고 동기화할 수AWS 계정 있습니다. 자세한 정보는 [Identity 및 Access Management에 대한 Amazon Transcribe](#)을 참조하세요.

모범 사례에 대한IAM 자세한 내용은 [의 보안 모범 사례](#)를 참조하십시오IAM.

## Amazon S3버킷 생성

Amazon S3안전한 오브젝트 스토리지 서비스입니다. Amazon S3파일 (객체라고 함) 을 컨테이너 (버킷이라고 함) 에 저장합니다.

일괄 트랜스크립션을 실행하려면 먼저 미디어 파일을Amazon S3 버킷에 업로드해야 합니다. 트랜스크립션 출력용Amazon S3 버킷을 지정하지 않은 경우 트랜스크립트를 임시AWS 관리형Amazon S3 버킷에Amazon Transcribe 넣습니다. AWS관리형 버킷의 트랜스크립션 출력은 90일 후에 자동으로 삭제됩니다.

[첫 번째 S3 버킷을 생성하고 버킷에 객체를 업로드하는](#) 방법을 알아봅니다.

## IAM 정책 생성

에서AWS 액세스를 관리하려면 정책을 생성하고IAM 자격 증명 (사용자, 역할) 또는AWS 리소스에 연결해야 합니다. 정책은 연결된 엔티티의 권한을 정의합니다. 예를 들어, 해당 역할에 액세스 권한을 부여하는 정책을 연결한 경우에만 역할이Amazon S3 버킷에 있는 미디어 파일에 액세스할 수 있습니다. 해당 역할을 추가로 제한하려면 대신Amazon S3 버킷 내 특정 파일에 대한 액세스를 제한할 수 있습니다.

AWS정책 사용에 대한 자세한 내용은 [섹션](#)을 참조하세요.

- [의 정책 및 권한IAM](#)
- [IAM정책 생성](#)

- [IAM의 Amazon Transcribe 작동 방식](#)

함께 사용할 수 있는 정책 예는 Amazon Transcribe 을 참조하십시오 [Amazon Transcribe 자격 증명 기반 정책 예제](#). 사용자 지정 정책을 생성하려면 [AWS정책 생성기](#)를 사용하는 것이 좋습니다.

AWS Management Console AWS CLI, 또는 AWS SDK를 사용하여 정책을 추가할 수 있습니다. 지침은 [IAM ID 권한 추가 및 제거](#)를 참조하십시오.

정책의 형식은 다음과 같습니다.

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

Amazon 리소스 이름 (ARN) 은 Amazon S3 버킷과 같은 모든 AWS 리소스를 고유하게 식별합니다. 정책에서 ARN을 사용하여 특정 리소스를 사용하기 위한 특정 작업에 대한 권한을 부여할 수 있습니다. 예를 들어 Amazon S3 버킷과 해당 하위 폴더에 대한 읽기 권한을 부여하려는 경우 신뢰 정책 Statement 섹션에 다음 코드를 추가할 수 있습니다.

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

}

다음은 Amazon S3 버킷과 해당 하위 폴더에 Amazon Transcribe 읽기 (GetObject, ListBucket, PutObject) 및 쓰기 (PutObject) 권한을 부여하는 예제 정책입니다. DOC-EXAMPLE-BUCKET

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

## 를 사용하여 필사하기 AWS Management Console

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

일괄 트랜스크립션을 시작하기 전에 먼저 미디어 파일을 Amazon S3 버킷에 업로드해야 합니다. 를 사용하여 트랜스크립션을 AWS Management Console 스트리밍하려면 컴퓨터 마이크를 사용해야 합니다.

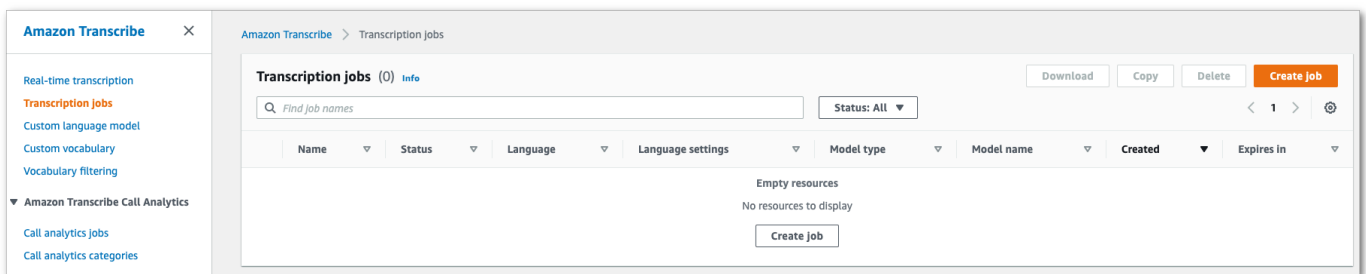
지원되는 미디어 형식과 기타 미디어 요구 사항 및 제약 조건을 보려면 을 참조하십시오 [데이터 입력 및 출력](#).

각 전사 방법에 대한 간략한 설명을 보려면 다음 섹션을 확장하십시오.

## Batch 트랜스크립션

먼저 필사하려는 미디어 파일을 Amazon S3 버킷에 업로드했는지 확인하세요. 이 작업을 수행하는 방법을 잘 모르겠으면 Amazon S3 사용 설명서: [버킷에 객체 업로드](#)를 참조하세요.

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



작업 생성을 선택합니다.

2. 작업 세부 정보 지정 페이지의 필드를 입력합니다.

## 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 내 객체여야 합니다. 출력 위치로 안전한 Amazon S3 서비스 관리 버킷을 선택하거나 자체 Amazon S3 버킷을 지정할 수 있습니다.

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

자체 Amazon S3 버킷을 선택하면 에서 미리 보기를 볼 수 AWS Management Console 없으며 해당 Amazon S3 버킷으로 이동하여 트랜스크립트를 다운로드해야 합니다.



### Input data [Info](#)

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

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

### Output data

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

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

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

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

SRT (SubRip)

VTT (WebVTT)

### Tags - optional

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

No tags associated with the resource.

You can add up to 50 more tags.

다음(Next)을 선택합니다.

3. 작업 구성 페이지에서 원하는 옵션을 선택합니다. [트랜스크립션과 사용자 지정 언어 모델 함께 사용자 지정 어휘](#) 또는 트랜스크립션을 사용하려면 트랜스크립션 작업을 시작하기 전에 이를 만들어야 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.

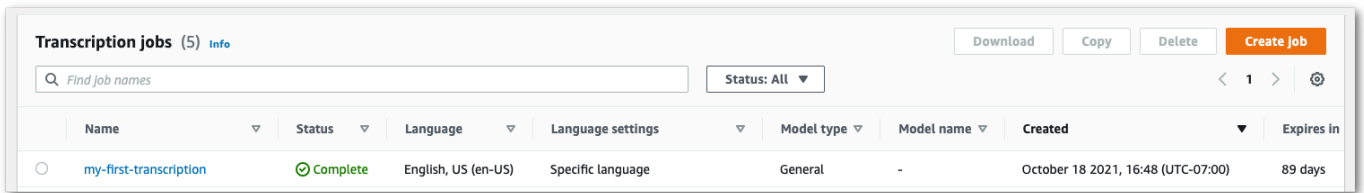
### Customization

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

Cancel Previous Create job

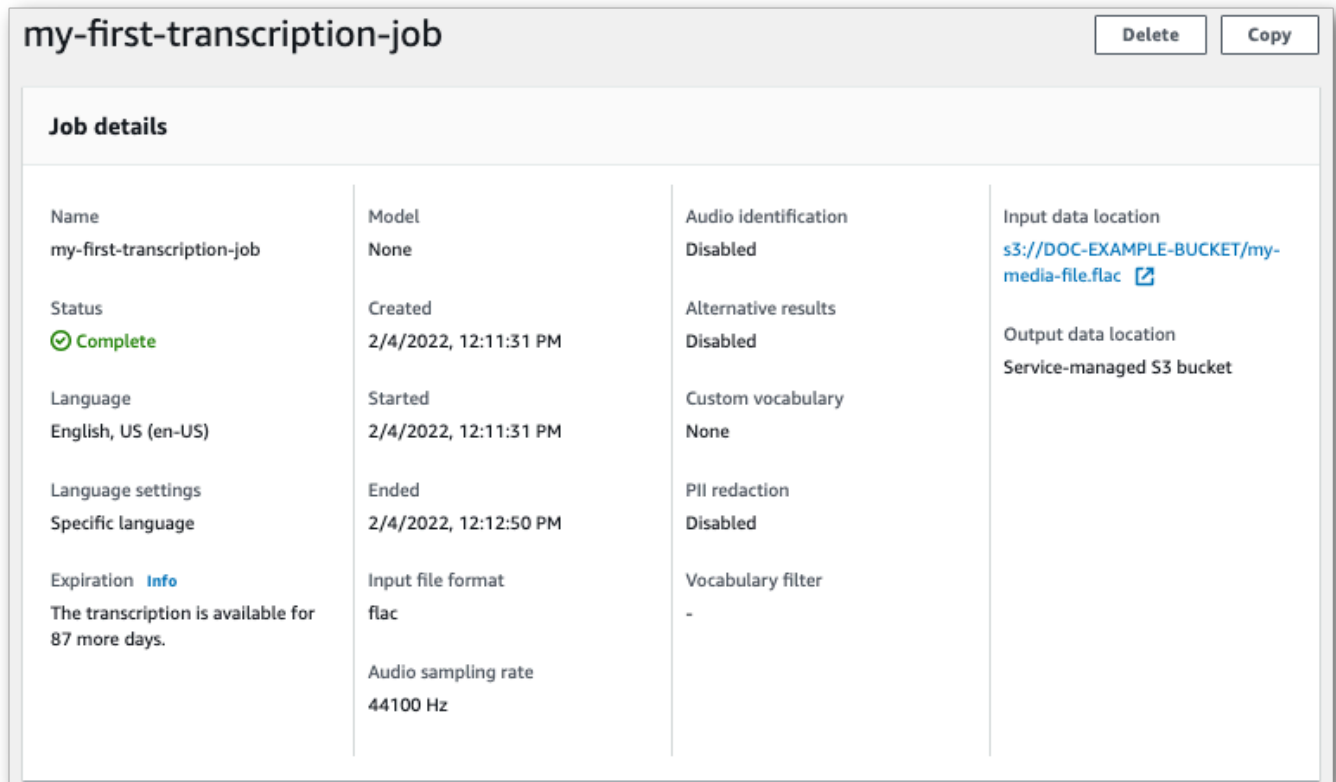
작업 생성을 선택합니다.

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



5. 이제 트랜스크립션의 Job 세부 정보 페이지가 표시됩니다. 여기에서 트랜스크립션 작업을 설정할 때 지정한 모든 옵션을 볼 수 있습니다.

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



6. 트랜스크립트를 다운로드하는 방법은 서비스 관리형 버킷을 선택했는지 아니면 자체 Amazon S3 버킷을 선택했는지에 따라 달라집니다.
- 서비스 관리형 버킷을 선택한 경우 트랜스크립션 작업의 정보 페이지에서 트랜스크립션 미리 보기 창과 다운로드 버튼을 볼 수 있습니다.

**my-first-transcription-job**

---

**Job details**

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

---

**Transcription preview**  ▼

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

Text
Audio identification
Subtitles

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

다운로드를 선택하고 트랜스크립트 다운로드를 선택합니다.

- b. 자체 Amazon S3 버킷을 선택한 경우 트랜스크립션 작업의 정보 페이지에 있는 트랜스크립션 미리 보기 패널에 텍스트가 표시되지 않습니다. 대신 선택한 Amazon S3 버킷으로 연결되는 링크가 있는 파란색 정보 상자가 표시됩니다.

**my-first-transcription-job**

---

**Job details**

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

---

**Transcription preview**  ▼

Select download to save a local copy of the transcription.

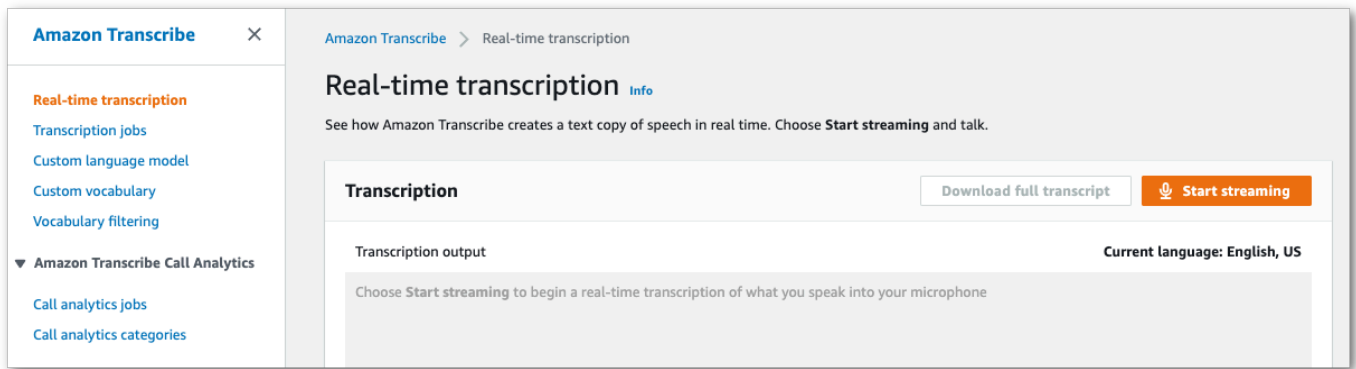
Text
Audio identification
Subtitles

❗ 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](#). [↗](#)

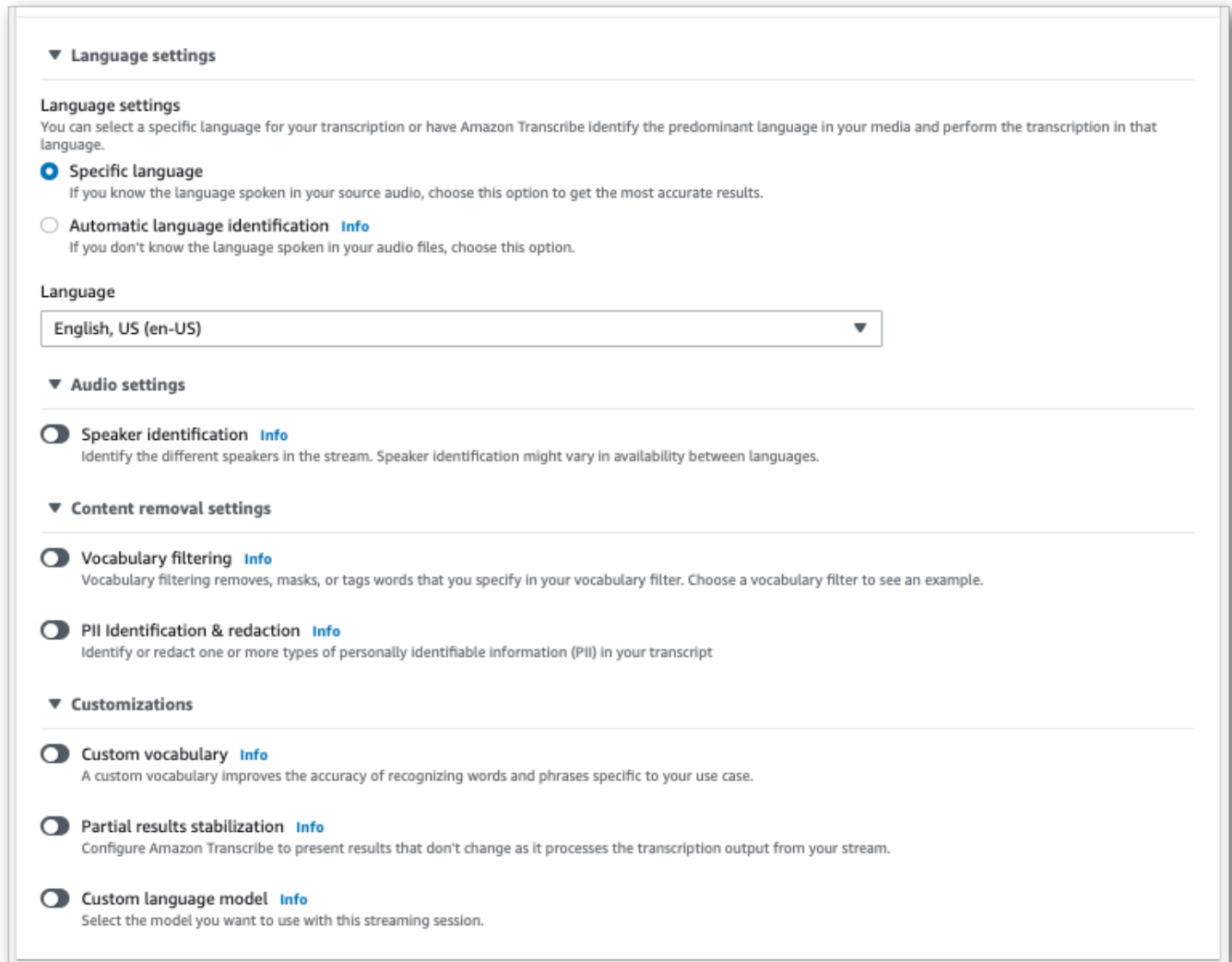
트랜스크립트에 액세스하려면 Job 세부 정보 창의 출력 데이터 위치 아래에 있는 링크 또는 트랜스크립션 미리 보기 패널의 파란색 정보 상자에 있는 S3 버킷 링크를 사용하여 지정된 Amazon S3 버킷으로 이동합니다.

## 스트리밍 트랜스크립션

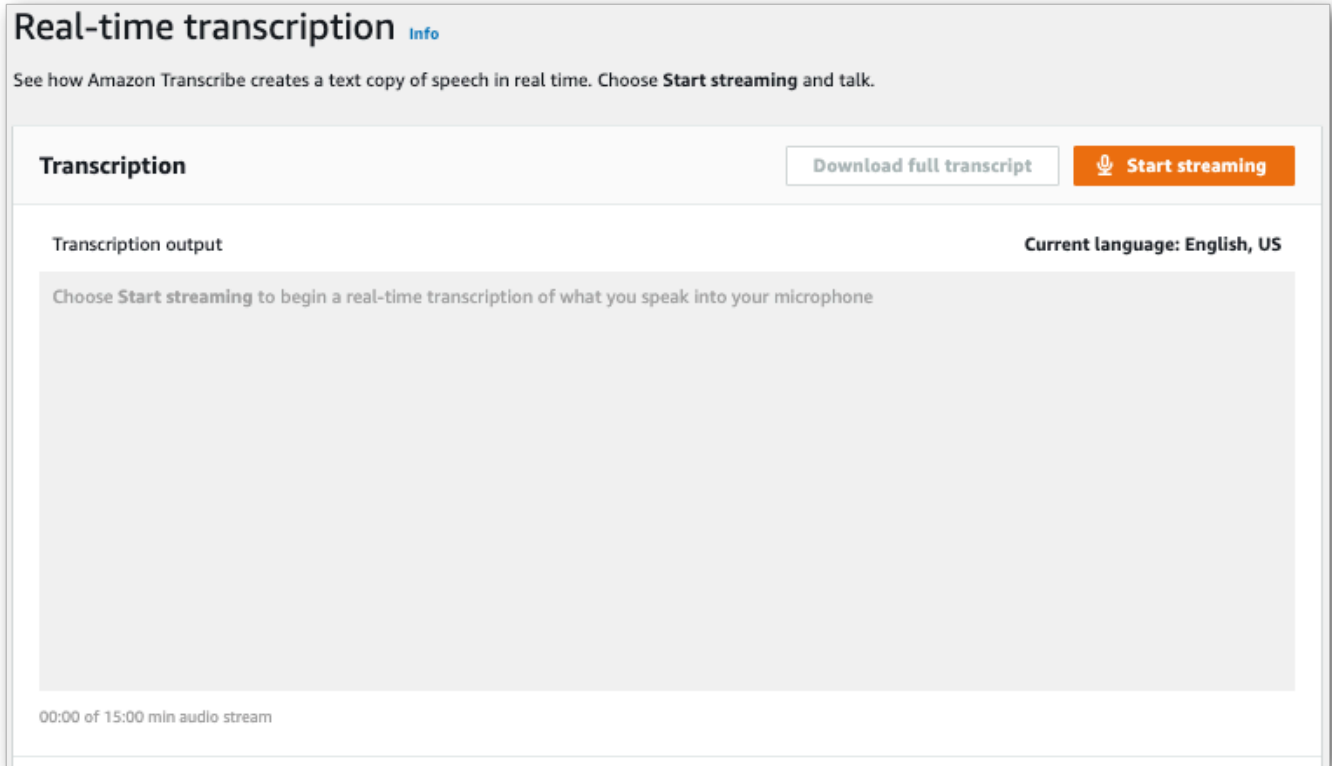
1. 왼쪽 탐색 창의 실시간 트랜스크립션을 선택합니다. [AWS Management Console](#) 그러면 스트리밍을 시작하기 전에 옵션을 선택할 수 있는 기본 스트리밍 페이지로 이동합니다.



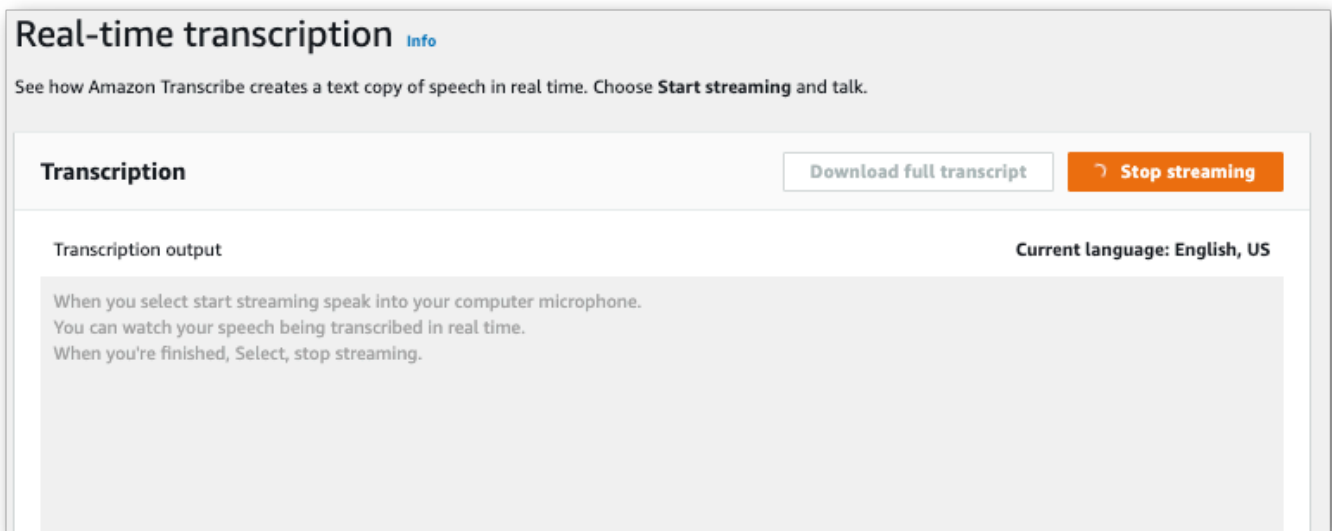
2. 트랜스크립션 출력 상자 아래에는 다양한 언어 및 오디오 설정을 선택할 수 있는 옵션이 있습니다.



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



4. 작업을 마쳤으면 스트리밍 중지를 선택합니다.



이제 전체 트랜스크립트 다운로드를 선택하여 트랜스크립트를 다운로드할 수 있습니다.

## 를 사용하여 필사하기AWS CLI

를AWS CLI 사용하여 트랜스크립션을 시작할 때 CLI 수준에서 모든 명령을 실행할 수 있습니다. 또는 사용하려는 명령을 실행한 다음 요청 본문이 포함된 JSON 파일의AWS 리전 및 위치를 실행할 수 있습니다. 이 안내서의 예에서는 두 방법을 모두 보여 주지만 이 섹션에서는 이전 방법을 중점적으로 다룹니다.

스트리밍 트랜스크립션은 지원되지AWS CLI 않습니다.

계속하기 전에 다음 사항을 확인하십시오.

- 미디어 파일을Amazon S3 버킷에 업로드했습니다. Amazon S3버킷을 만들거나 파일을 업로드하는 방법을 잘 모르겠으면 [첫 번째Amazon S3 버킷 만들기 및 버킷에 객체 업로드](#)를 참조하세요.
- 설치했습니다 [AWS CLI](#).

명령Amazon Transcribe [참조에서](#)에 대한 모든AWS CLI 명령을 찾을 수 있습니다.AWS CLI

### 새 트랜스크립션 작업 시작

새 트랜스크립션을 시작하려면start-transcription-job 명령을 사용합니다.

1. 터미널 창에서 다음을 입력합니다.

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

다음 줄에> "이 표시되며, 이제 다음 단계에 설명된 대로 필수 매개 변수를 계속 추가할 수 있습니다.

\\"`을 생략하고 모든 매개 변수를 공백으로 구분하여 추가할 수도 있습니다.`

2. start-transcription-job명령에는,regiontranscription-job-namemedia, 및language-code 또는 를 포함해야identify-language 합니다.

출력 위치를 지정하려면 요청에output-bucket-name 포함하십시오. 지정된 출력 버킷의 하위 폴더를 지정하려면 요청에 포함하십시오output-key.

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



```
--language-code en-US
```

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

를 사용하여 `output-bucket-name` 출력 버킷을 지정하지 않도록 선택한 경우는 트랜스크립션 출력을 서비스 관리형 버킷에 저장합니다. Amazon Transcribe 서비스 관리 버킷에 저장된 트랜스크립트는 90일 후에 만료됩니다.

Amazon Transcribe 다음과 같이 응답합니다.

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

에서 로 [TranscriptionJobStatus](#) 변경하면 트랜스크립션 작업이 성공한 `IN_PROGRESS` 것입 니다 `COMPLETED`. [TranscriptionJobStatus](#) 업데이트된 내용을 보려면 다음 섹션에 표시된 대 로 `get-transcription-job` 또는 `list-transcription-job` 명령을 사용하십시오.

## 트랜스크립션 작업의 상태 가져오기

트랜스크립션 작업에 대한 정보를 보려면 `get-transcription-job` 명령을 사용하십시오.

이 명령에 필요한 유일한 매개 변수는 작업이 AWS 리전 있는 위치 및 작업 이름입니다.

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

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

Amazon Transcribe 다음과 같이 응답합니다.

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

트랜스크립션 출력용으로 자체 Amazon S3 버킷을 선택한 경우 이 버킷이 다음과 같이 나열됩니다. TranscriptFileUri. 서비스 관리형 버킷을 선택한 경우 임시 URI가 제공됩니다. 이 URI를 사용하여 트랜스크립트를 다운로드하세요.

#### Note

서비스 관리형 Amazon S3 버킷의 임시 URI는 15분 동안만 유효합니다. URI를 사용할 때 AccessDenied 오류가 발생하면 get-transcription-job 요청을 다시 실행하여 새 임시 URI를 가져오세요.

## 트랜스크립션 작업 목록 작성

주어진 AWS 리전 모든 트랜스크립션 작업을 나열하려면 `list-transcription-jobs` 명령을 사용하십시오.

이 명령에 필요한 유일한 매개 변수는 트랜스크립션 작업이 있는 위치입니다. AWS 리전

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

Amazon Transcribe 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 삭제하기

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

이 명령에 필요한 유일한 매개 변수는 작업이 AWS 리전 있는 위치 및 작업 이름입니다.

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

삭제 요청이 성공적으로 완료되었는지 확인하려면 `list-transcription-jobs` 명령을 실행할 수 있습니다. 목록에 표시됩니다.

## SDK를 AWS 사용한 트랜스크립션

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

함께 사용할 수 있는 프로그래밍 언어 목록은 Amazon Transcribe을 참조하십시오. [지원되는 프로그래밍 언어](#) 모든 AWS SDK에서 스트리밍 트랜스크립션이 지원되는 것은 아닙니다. 지원되는 미디어 형식과 기타 미디어 요구 사항과 제약 조건을 보려면 [데이터 입력 및 출력](#)을 참조하세요.

사용 가능한 모든 AWS SDK 및 빌더 도구에 대한 자세한 내용은 빌드 기반 [도구](#)를 참조하십시오.  
AWS

### Tip

기능별, 시나리오, 크로스 서비스 예제를 포함하여 AWS SDK를 사용하는 추가 예제는 이 장을 참조하십시오. [SDK를 사용한 Amazon Transcribe의 코드 예제 AWS](#) 다음 리포지토리에서도 SDK 코드 샘플을 찾을 수 있습니다. GitHub

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

## 배치 트랜스크립션

Amazon S3 버킷에 있는 미디어 파일의 URI를 사용하여 배치 트랜스크립션을 생성할 수 있습니다. 버킷을 생성하거나 파일을 업로드하는 방법을 잘 모르는 경우 [첫 번째 S3 Amazon S3 버킷 생성 및 버킷에 객체 업로드](#)를 참조하십시오.

## Java

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

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

```
private static TranscribeClient client;

public static void main(String args[]) {

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

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

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

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

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

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

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

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

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

## JavaScript

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

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

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

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

```
    console.log(err);
  }
}

startTranscriptionRequest();
```

## Python

```
import time
import boto3

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

    max_tries = 60
    while max_tries > 0:
        max_tries -= 1
        job = transcribe_client.get_transcription_job(TranscriptionJobName =
job_name)
        job_status = job['TranscriptionJob']['TranscriptionJobStatus']
        if job_status in ['COMPLETED', 'FAILED']:
            print(f"Job {job_name} is {job_status}.")
            if job_status == 'COMPLETED':
                print(
                    f"Download the transcript from\n"
                    f"\t{job['TranscriptionJob']['Transcript']
['TranscriptFileUri']}")
                break
            else:
                print(f"Waiting for {job_name}. Current status is {job_status}.")
                time.sleep(10)

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

```

transcribe_file('Example-job', file_uri, transcribe_client)

if __name__ == '__main__':
    main()

```

## 스트리밍 트랜스크립션

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

Amazon Transcribe 스트리밍에는 AWS SDK for Python (Boto3) 표준이 지원되지 않는다는 점에 유의하십시오. Python을 사용하여 스트리밍 트랜스크립션을 시작하려면 이 [비동기 Python SDK](#)를 사용하십시오. Amazon Transcribe

### Java

다음 예시는 스트리밍 오디오를 트랜스크립션하는 Java 프로그램입니다.

이 예를 실행하려면 다음을 유의하세요.

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

참조: [Amazon Transcribe 스트리밍용 클라이언트 재시도](#) (Java SDK). 이 코드는 Amazon Transcribe에 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.

## SDK와 함께 이 서비스 사용 AWS

AWS 소프트웨어 개발 키트 (SDK) 는 널리 사용되는 여러 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 예시
<a href="#">AWS SDK for C++</a>	<a href="#">AWS SDK for C++ 코드 예제</a>
<a href="#">AWS CLI</a>	<a href="#">AWS CLI 코드 예제</a>
<a href="#">AWS SDK for Go</a>	<a href="#">AWS SDK for Go 코드 예제</a>
<a href="#">AWS SDK for Java</a>	<a href="#">AWS SDK for Java 코드 예제</a>
<a href="#">AWS SDK for JavaScript</a>	<a href="#">AWS SDK for JavaScript 코드 예제</a>
<a href="#">AWS SDK for Kotlin</a>	<a href="#">AWS SDK for Kotlin 코드 예제</a>
<a href="#">AWS SDK for .NET</a>	<a href="#">AWS SDK for .NET 코드 예제</a>
<a href="#">AWS SDK for PHP</a>	<a href="#">AWS SDK for PHP 코드 예제</a>
<a href="#">AWS Tools for PowerShell</a>	<a href="#">PowerShell 코드 예제를 위한 도구</a>
<a href="#">AWS SDK for Python (Boto3)</a>	<a href="#">AWS SDK for Python (Boto3) 코드 예제</a>
<a href="#">AWS SDK for Ruby</a>	<a href="#">AWS SDK for Ruby 코드 예제</a>
<a href="#">AWS SDK for Rust</a>	<a href="#">AWS SDK for Rust 코드 예제</a>
<a href="#">AWS SDK for SAP ABAP</a>	<a href="#">AWS SDK for SAP ABAP 코드 예제</a>
<a href="#">AWS SDK for Swift</a>	<a href="#">AWS SDK for Swift 코드 예제</a>

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

#### 예제 사용 가능 여부

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



## HTTP를 이용한 트랜스크립팅 또는 WebSockets

Amazon Transcribe 일괄 처리 (HTTP/1.1) 및 스트리밍 (HTTP/2) 트랜스크립션 모두에 대해 HTTP를 지원합니다. WebSockets 스트리밍 트랜스크립션이 지원됩니다.

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

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

### Batch 트랜스크립션

다음 헤더를 사용하여 일괄 HTTP 요청을 만들 수 있습니다.

- host
- x-amz-target
- 콘텐츠 유형
- x-amz-content-sha256
- x-amz-date
- authorization

다음은 `StartTranscriptionJob` 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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
  },
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",
  "OutputKey": "my-output-files/"
}
```

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

## 스트리밍 트랜스크립션

HTTP/2를 사용하여 트랜스크립션을 WebSockets 스트리밍하는 것은 SDK를 사용하는 것보다 더 복잡합니다. 첫 번째 스트림을 설정하기 전에 [스트리밍 트랜스크립션 설정](#) 섹션을 검토하는 것이 좋습니다.

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

### Note

트랜스크립션 스트리밍에는 SDK를 사용하는 것이 좋습니다. 지원되는 SDK 목록은 을 참조하십시오 [지원되는 프로그래밍 언어](#).

## 스트리밍 오디오 트랜스크립션

Amazon Transcribe 스트리밍을 사용하면 미디어 콘텐츠에 대한 실시간 트랜스크립션을 생성할 수 있습니다. 미디어 파일을 업로드하는 일괄 트랜스크립션과 달리 스트리밍 미디어는 실시간으로 제공됩니다. Amazon Transcribe Amazon Transcribe 그러면 트랜스크립트가 동시에 실시간으로 반환됩니다.

스트리밍에는 사전 녹화된 미디어(영화, 음악, 팟캐스트)와 실시간 미디어(실시간 뉴스 방송)가 포함될 수 있습니다. 의 일반적인 스트리밍 사용 Amazon Transcribe 사례로는 스포츠 경기를 위한 라이브 자막, 콜센터 오디오의 실시간 모니터링 등이 있습니다.

스트리밍 콘텐츠는 Amazon Transcribe 가 순식간에 변환하는 일련의 순차적 데이터 패킷 또는 '청크'로 전달됩니다. 스트리밍을 일괄 처리보다 사용하면 애플리케이션의 실시간 speech-to-text 기능이 지원되고 트랜스크립션 시간이 단축된다는 장점이 있습니다. 그러나 이러한 속도 향상으로 인해 경우에 따라 정확도가 제한될 수 있습니다.

Amazon Transcribe 다음과 같은 스트리밍 옵션을 제공합니다.

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

에서 스트리밍 오디오를 녹음하려면 컴퓨터 마이크에 AWS Management Console대고 말하십시오.

### Tip

SDK 코드 예제는 의 [AWS 샘플 저장소](#)를 참조하십시오. GitHub

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 지역 가용성을 보려면 [Amazon Transcribe 엔드포인트](#) 및 할당량을 참조하십시오.

## 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 자막이 일부 결과로 생성됩니다. Amazon Transcribe 화자 변경이 나 오디오 일시 중지와 같은 자연스러운 음성 세그먼트를 기준으로 들어오는 오디오 스트림을 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 개체 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 Transcribe 오디오 스트리밍을 시작하자마자 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 Transcribe 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 안정화는 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 largest rainfall.	4.015	true
The Amazon is the largest rain forest.	4.515	true
The Amazon is the largest rain forest on the planet.	5.015	true
The Amazon is the largest rain forest on the planet.	5.515	true
The Amazon is the largest rain forest on the planet.	6.015	true
The Amazon is the largest rain forest on the planet.	6.335	true
The Amazon is the largest rain forest on the planet.	6.335	false

부분 결과 안정화를 활성화하면 Stable 필드를 Amazon Transcribe 사용하여 항목이 안정적인지 여부를 나타냅니다. 여기서 '항목'은 필사된 단어 또는 문장 부호를 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.



### ① AWS Machine Learning 블로그를 통해 더 자세히 알아보기

실시간 트랜스크립션으로 정확도를 높이는 방법에 대해 자세히 알아보려면 다음을 참조하세요.

- [Amazon Transcribe 부분 결과 안정화를 통해 스트리밍 트랜스크립션 경험을 개선하세요.](#)
- [“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, SDK 대신 HTTP/2를 사용하거나 WebSockets 직접 스트림을 설정하려는 사용자에게 정보를 제공하기 위한 것입니다. AWS 이 섹션의 정보를 사용하여 자체 SDK를 빌드할 수도 있습니다.

### Important

HTTP/2를 직접 사용하는 대신 SDK를 사용하는 것이 좋습니다. WebSockets SDK는 데이터 스트림을 트랜스크립션하는 가장 간단하고 신뢰할 수 있는 방법입니다. AWS SDK를 사용하여 스트리밍을 시작하려면 [을 참조하십시오. SDK를 AWS 사용한 트랜스크립션](#)

## HTTP/2 스트림 설정

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

- 헤더 프레임. 여기에는 요청에 대한 HTTP/2 헤더와 데이터 프레임에 서명하기 위한 시드 서명으로 Amazon Transcribe 사용하는 인증 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 HTTP/2 세션당 스트림 하나만 지원합니다. 스트림을 여러 개 사용하려고 하면 트랜스크립션 요청이 실패합니다.

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

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "my-transcribe-http2-policy",
      "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 다음 응답을 보냅니다.

```
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"
Uk1GRjzxPQBxQVZFZm10IBAAAAAABAAEAgD4AAAB9AAACABAAZGF0YVVTwPQAAAAAAAAAAAAAAAAAAD//wIA/f8EAA==
```

- 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	타임스탬프

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

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

```
AAAA0gAAAIKVoRFcTTcjb250ZW50LXR5cGUHABhhcHBsaWNhdGlvbi9vY3RldC1zdHJlYW0L0mV2ZW50LXR5cGUHAAPBdWRpb0V2ZW50DTptZXNzYWdl1LXR5cGUHAAV1dmVudAxDb256ZW50LVR5cGUHABphcHBsaWNhdGlvbi94LWFtei1qc29uLTFEuMVJJRkY88T0AV0FWRWZtdCAQAAAAQAABAIA
+AAAAfQAAAQAQAGRhdGFU8D0AAAAA
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:** 입력의 SHA-256 해시를 먼저 생성한 다음 Hex 함수를 사용하여 해시를 인코딩하는 함수입니다.
  - **non SignatureHeaders:** 문자열로 인코딩된 DateTime 헤더.
  - **payload:** 오디오 이벤트 데이터가 포함된 바이트 버퍼.
- c. AWS 보안 액세스 키에서 서명 키를 추출하고 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.
- **파생 SigningKey:** 시그니처 버전 4 서명 키.
- **문자열 ToSign:** 데이터 프레임에 대해 계산한 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 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 Amazon Transcribe 응답을 보냅니다. 응답은 이벤트 스트림 인코딩됩니다.

응답에는 표준 서문과 다음 헤더가 포함됩니다.

헤더 이름 길이	헤더 이름(문자열)	헤더 값 유형	값 문자열 길이	값 문자열(UTF-8)
13	:content-type	7	16	application/json
11	:event-type	7	19	BadRequest예외
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

트랜스크립션 요청을 스트리밍하기 위한 [WebSocket프로토콜의](#) 주요 구성 요소는 Amazon Transcribe 다음과 같습니다.

- 업그레이드 요청. 여기에는 요청에 대한 쿼리 파라미터와 데이터 프레임에 서명하기 위한 시드 서명으로 Amazon Transcribe 사용하는 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 WebSocket 세션당 하나의 스트림만 지원합니다. 스트림을 여러 개 사용하려고 하면 트랜스크립션 요청이 실패합니다.



- 요청을 보내는 IAM 역할에 다음 정책을 연결합니다. 자세한 내용은 [IAM 정책 추가](#)를 참조하십시오.

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

- 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%2FYYYYYMMDD%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 하면 요청을 AWS 받을 때 URL에 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
datestamp = 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 추출해야 합니다. 생성된 키는 더 높은 수준의 보호를 위해 낱자, 서비스 및 AWS 리전에 고유합니다. 생성된 키를 사용하여 요청에 서명합니다. 자세한 내용은 [내용은 AWS 서명 버전 4의 서명 계산을 참조하십시오](#).

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
```

WebSocket 라이브러리의 요청 URL을 사용하여 요청하십시오 Amazon Transcribe.

6. 요청에는 다음 헤더가 Amazon Transcribe 포함되어야 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. WebSocket 요청을 Amazon Transcribe 받으면 WebSocket 업그레이드 응답으로 응답합니다. 일반적으로 WebSocket 라이브러리는 이 응답을 관리하고 통신을 위한 소켓을 설정합니다. Amazon Transcribe

의 응답은 다음과 같습니다 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 요청하세요.

WebSocket 연결이 설정되면 클라이언트는 각각 [이벤트 스트림](#) 인코딩을 사용하여 인코딩된 오디오 프레임 시퀀스를 전송할 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 발생하는 경우 이벤트 스트림으로 인코딩된 Amazon Transcribe 응답이 포함된 터미널 WebSocket 프레임으로 응답합니다. 이 응답에는 다음 테이블에 설명된 헤더가 포함되며, 응답 본문에는 서술적인 오류 메시지가 포함되어 있습니다. 예외 응답을 보낸 후 클로즈 Amazon Transcribe 프레임을 보냅니다.

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

헤더 이름 바이트 길이	헤더 이름(문자열)	헤더 값 유형	값 문자열 바이트 길이	값 문자열(UTF-8)
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**: Amazon Transcribe 클라이언트와의 핸드셰이크 중에 문제가 발생했습니다. 요청을 다시 시도하세요.
- **LimitExceededException**: 클라이언트가 동시 스트림 제한을 초과했습니다. 자세한 내용은 [Amazon Transcribe 제한](#)을 참조하세요. 트랜스크립션하는 스트림의 수를 줄입니다.
- **UnrecognizedClientException**: 잘못된 액세스 키 또는 비밀 키로 WebSocket 업그레이드 요청이 서명되었습니다. 액세스 키를 올바르게 생성하고 있는지 확인하고 요청을 다시 시도하세요.

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

## 이벤트 스트림 인코딩

Amazon Transcribe 스트리밍 트랜스크립션에 이벤트 스트림 인코딩이라는 형식을 사용합니다.

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

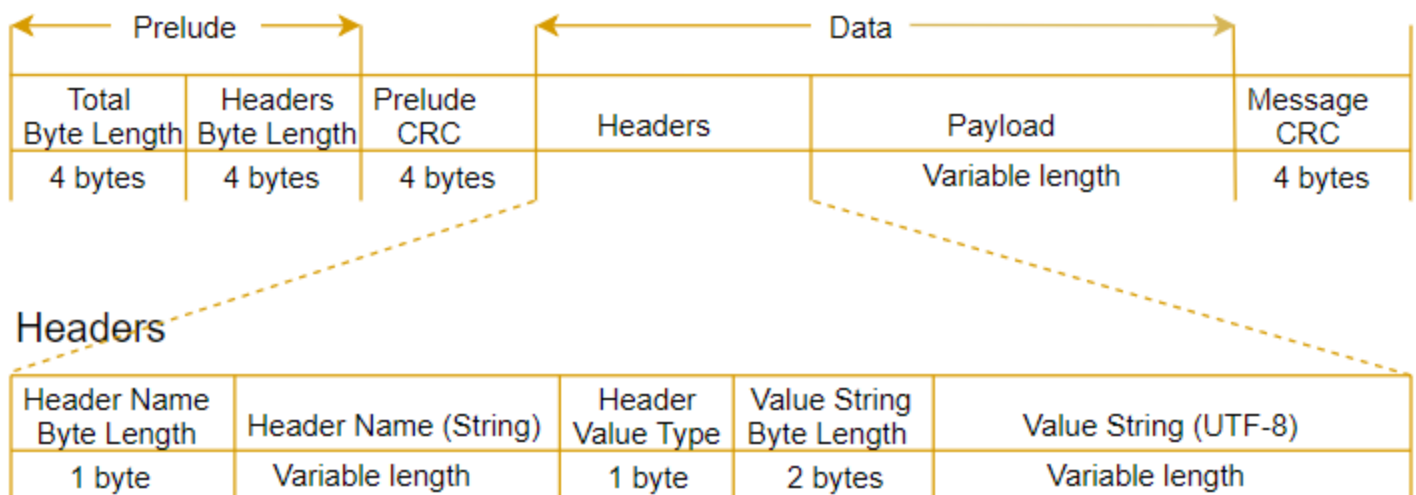
데이터 섹션의 구성 요소는 다음과 같습니다.

1. 헤더
2. 페이로드

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 CRC32라고도 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. 따라서 버퍼 오버런과 같은 오류가 발생하지 않고 손상된 바이트 길이 정보를 즉시 Amazon Transcribe 탐지할 수 있습니다.
- 헤더: 메시지 유형, 콘텐츠 유형 등의 메시지 주석 메타데이터. 메시지에 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 트랜스크립션의 인증 프레임입니다. Amazon Transcribe 권한 부여 헤더의 값을 시드로 사용하여 요청의 데이터 프레임에 대한 인증 헤더 체인을 생성합니다.

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

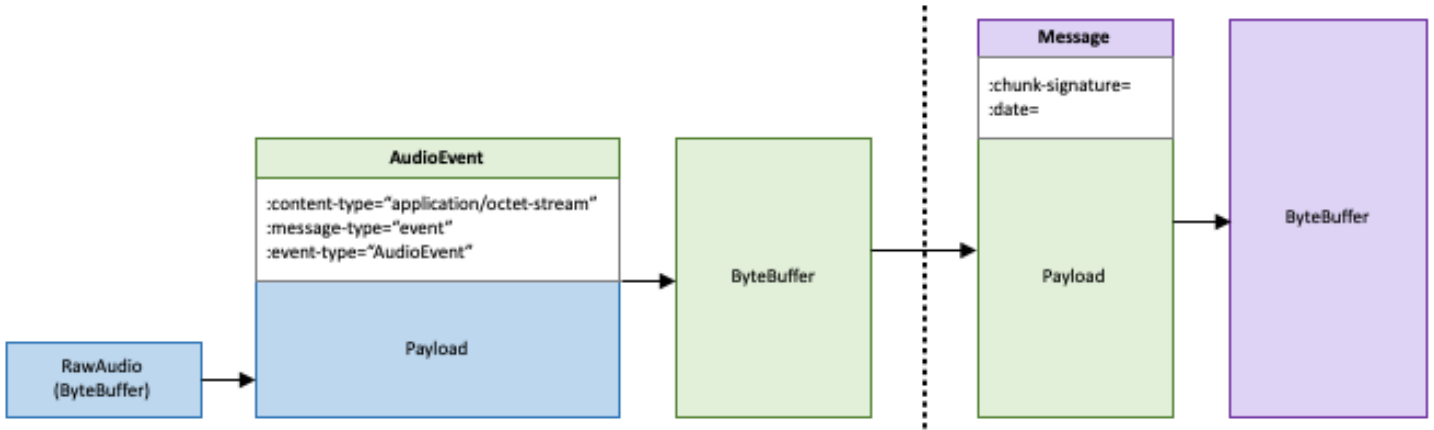
- 헤더 이름 바이트 길이: 헤더 이름의 바이트 길이.
- 헤더 이름: 헤더 유형을 나타내는 헤더 이름. 유효한 값은 다음 프레임 설명을 참조하세요.
- 헤더 값 유형: 헤더 값 유형을 나타내는 숫자. 다음 목록은 헤더에 사용할 수 있는 값과 해당 값의 의미를 보여줍니다.
  - 0 – TRUE(참)
  - 1 – FALSE(거짓)
  - 2 – BYTE(바이트)
  - 3 – SHORT(짧음)
  - 4 – INTEGER(정수)
  - 5 – LONG(길)
  - 6 – BYTE ARRAY(바이트 배열)
  - 7 – STRING(문자열)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



## Job 대기열

작업 큐를 사용하면 동시에 처리할 수 있는 것보다 더 많은 트랜스크립션 작업 요청을 제출할 수 있습니다. 작업 큐를 사용하지 않는 경우 허용된 동시 요청 할당량에 도달하면 새 요청을 제출하기 전에 하나 이상의 요청이 완료될 때까지 기다려야 합니다.

트랜스크립션 Job 요청의 경우 작업 큐는 선택 사항입니다. 통화 후 분석 요청에는 작업 대기열이 자동으로 활성화됩니다.

작업 큐를 활성화하면 제한을 초과하는 모든 요청이 포함된 큐가 Amazon Transcribe 생성됩니다. 요청이 완료되는 즉시 새 요청이 대기열에서 삭제되어 처리됩니다. 대기 중인 요청은 FIFO (선입선출) 순서로 처리됩니다.

대기열에 최대 10,000개의 작업을 추가할 수 있습니다. 이 한도를 초과하면 `LimitExceededConcurrentJobException` 오류가 발생합니다. 최적의 성능을 유지하려면 할당량의 최대 90% (대역폭 비율 0.9) Amazon Transcribe 까지만 대기열에 있는 작업을 처리하는 데 사용합니다. 특별한 언급이 없는 한 요청 시 수준 및 버스트 한도를 높일 수 있습니다.

### Tip

Amazon Transcribe 리소스의 기본 한도 및 할당량 목록은 [AWS 일반 참조에서](#) 확인할 수 있습니다. 요청 시 이러한 기본 수준 및 버스트 한도를 높일 수 있습니다.

작업 큐를 활성화했지만 동시 요청 할당량을 초과하지 않는 경우 모든 요청이 동시에 처리됩니다.

## 작업

AWS Management Console AWS CLI, 또는 AWSSDK를 사용하여 작업 큐를 활성화할 수 있습니다. 예를 보려면 다음을 참조하고 예제는 다음을 참조하십시오.

### 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. Job 설정 상자에는 추가 설정 패널이 있습니다. 이 패널을 확장하면 작업 큐에 추가 상자를 선택하여 작업 큐를 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.

#### Model type [Info](#)

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

**General model**

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

**Custom language model**

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

#### ▼ Additional settings

##### Job queue - *optional* [Info](#)

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

Add to job queue

4. 작업 세부 정보 지정 페이지에 포함하려는 다른 필드를 모두 입력한 후 다음을 선택합니다. 그러면 작업 구성 - 옵션 페이지로 이동합니다.
5. 작업 생성을 선택하여 트랜스크립션 작업을 실행합니다.

## AWS CLI

이 예제에서는 [start-transcription-job](#) 명령과 `job-execution-settings` 매개 변수를 `AllowDeferredExecution` 하위 매개 변수와 함께 사용합니다. 요청에 포함할 때는 반드시 `AllowDeferredExecution` 포함해야 한다는 점에 유의하십시오. `DataAccessRoleArn`.

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

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

다음은 [start-transcription-job](#) 명령을 사용하는 또 다른 예와 큐잉을 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 다음 요청 본문이 포함되어 있습니다.

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

## AWS SDK for Python (Boto3)

이 예제에서는 AWS SDK for Python (Boto3) 를 사용하여 [start\\_transcription\\_job](#) 메서드의 `AllowDeferredExecution` 인수를 사용하여 작업 큐를 활성화합니다. 요청에 포함할 때는 반드시 `AllowDeferredExecution` 포함해야 한다는 점에 유의하십시오. `DataAccessRoleArn`. 자세한 내용은 [StartTranscriptionJob](#) 및 [JobExecutionSettings](#) 단원을 참조하세요.

기능별, 시나리오 및 서비스 간 예제를 포함하여 AWS SDK를 사용하는 추가 예제는 이 [SDK를 사용한 Amazon Transcribe의 코드 예제 AWS](#) 장을 참조하십시오.

```

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

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

```

를 통해 AWS Management Console 또는 [GetTranscriptionJob](#) 요청을 제출하여 대기 중인 작업의 진행 상황을 볼 수 있습니다. 작업이 큐에 추가되면 해당 작업이 큐에 `Status` 추가됩니다. `QUEUED`. 작업이 처리를 시작하면 상태가 `IN_PROGRESS` 변경되고, 처리가 `FAILED` 완료되면 상태가 `COMPLETED` 변경됩니다.

## 리소스에 태그 지정

태그는 검색에서 쉽게 식별, 구성 및 찾을 수 있도록 리소스에 추가할 수 있는 사용자 지정 메타데이터 레이블입니다. 태그는 태그 키와 태그 값이라는 두 개의 개별 부분으로 구성됩니다. 이를 키:값 쌍이라고 합니다.

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

### Tip

AWS Billing and Cost Management 태그를 사용하여 청구서를 동적 범주로 구분할 수 있습니다. 예를 들어 Department:Sales 또는 Department:Legal 같이 회사 내 여러 부서를 나타내는 태그를 추가하면 부서별 비용 분배를 제공할 수 있습니다.

Amazon Transcribe에서 다음 리소스에 태그를 지정할 수 있습니다.

- 트랜스크립션 작업
- 의료 기록 작업
- 콜 애널리틱스 통화 후 트랜스크립션 작업
- Custom 어휘
- 맞춤형 의료 어휘
- 사용자 지정 어휘 필터
- 사용자 지정 언어 모델

태그 키는 최대 128개 문자이며, 태그 값은 최대 256개 문자입니다. Amazon Transcribe 리소스당 최대 50개의 태그를 지원합니다. 지정된 리소스에 대해 각 태그 키는 하나의 값만 갖는 고유해야 합니다. 이 접두사는 시스템 생성 태그로 aws:AWS 예약되므로 태그는 로 시작할 수 없습니다. aws:\*태그는 추가, 수정 또는 삭제할 수 없으며 tags-per-resource 제한에 포함되지 않습니다.

### 리소스 태깅과 관련된 API 작업

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

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

모범 사례를 비롯한 태깅에 대해 자세히 알아보려면 [AWS 리소스 태깅을](#) 참조하세요.

## 태그 기반 액세스 제어

태그를 사용하여 내 액세스를 제어할 수 있습니다. AWS 계정. 태그 기반 액세스 제어의 경우 IAM 정책의 조건 요소에 태그 정보를 제공합니다. 그런 다음 태그 및 연관된 태그 조건 키를 사용하여 다음에 대한 액세스를 제어할 수 있습니다.

- 리소스: 리소스에 할당된 태그를 기반으로 Amazon Transcribe 리소스에 대한 액세스를 제어합니다.
  - `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.
- 승인 프로세스: 승인 프로세스의 모든 부분에 대한 태그 기반 액세스를 제어합니다.
  - `aws:TagKeys/조건 키`를 사용하여 리소스, 요청 또는 보안 주체에 특정 태그 키를 사용할 수 있는지 여부를 제어합니다. 이 경우 키 값은 중요하지 않습니다.

태그 기반 액세스 제어 정책의 예는 단원을 참조하세요. [태그를 기준으로 트랜스크립션 작업 보기.](#)

태그 기반 액세스 제어에 대한 자세한 내용은 [태그를 사용한 AWS 리소스에 대한 액세스](#) 제어를 참조하세요.

## Amazon Transcribe 리소스에 태그 추가

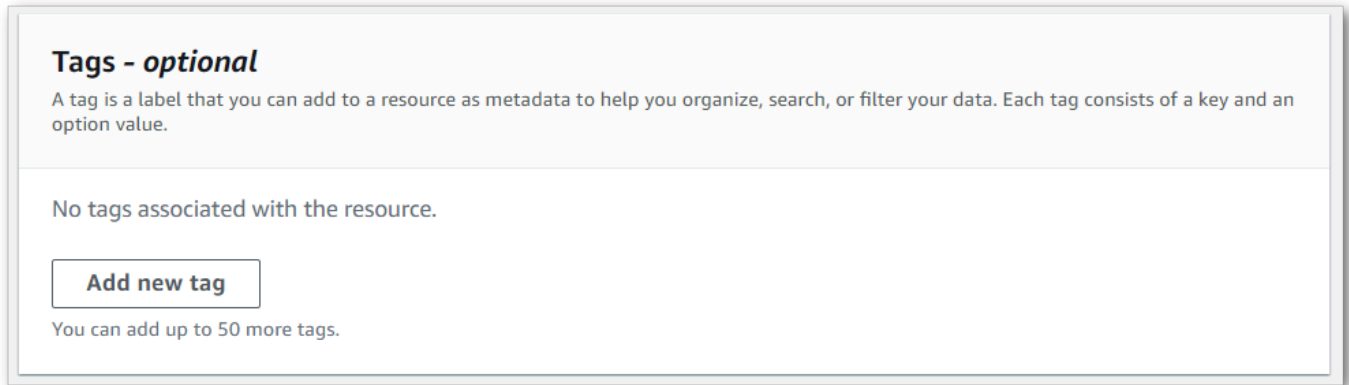
Amazon Transcribe 작업을 실행하기 전이나 후에 태그를 추가할 수 있습니다. 기존 `Create*` 및 `Start*` API를 사용하여 트랜스크립션 요청에 태그를 추가할 수 있습니다.

, 또는 AWSSDK를 사용하여 태그를 추가 AWS Management Console AWS CLI, 수정 또는 삭제할 수 있습니다. 예를 보려면 다음을 참조하십시오.

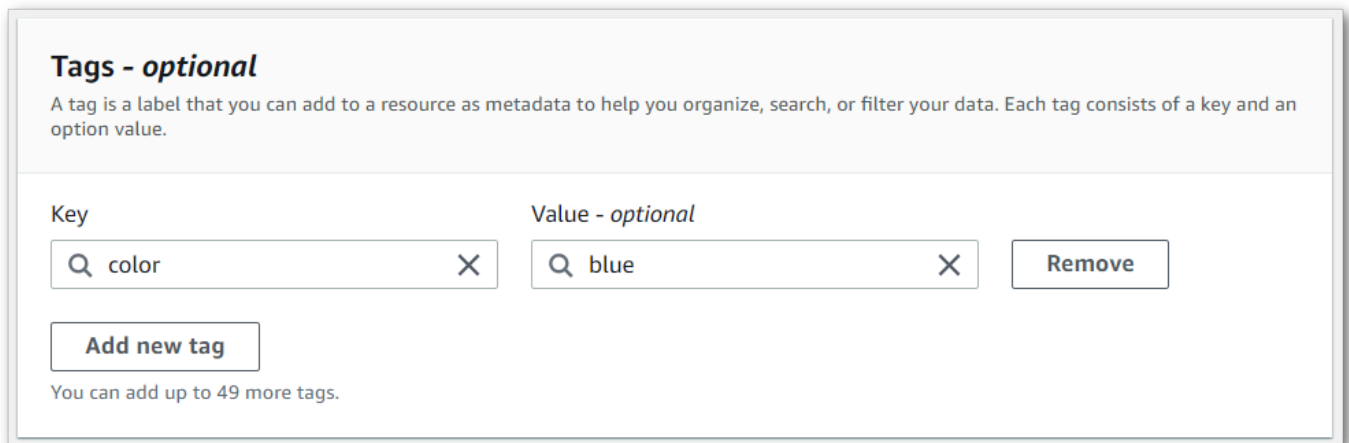


## AWS Management Console

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 페이지 하단으로 스크롤하여 태그 - 옵션 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 입력한 후 다음을 선택합니다. 그러면 작업 구성 - 옵션 페이지로 이동합니다.

작업 생성을 선택하여 트랜스크립션 작업을 실행합니다.

6. 트랜스크립션 작업 페이지로 이동하여 트랜스크립션 작업을 선택한 다음 해당 작업의 정보 페이지 하단으로 스크롤하여 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 예제에서는 [start-transcription-job](#) 명령과 Tags 매개 변수를 사용합니다. 자세한 내용은 [StartTranscriptionJob](#) 및 [Tag](#) 단원을 참조하세요.

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

다음은 [start-transcription-job](#) 명령을 사용하는 또 다른 예와 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 다음 요청 본문이 포함되어 있습니다.

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

```

        "Key": "color",
        "Value": "blue"
    },
    {
        "Key": "shape",
        "Value": "square"
    }
]
}

```

## AWS SDK for Python (Boto3)

다음 예제에서는  `boto3` 를 사용하여 [start\\_transcription\\_job](#) 메서드의 `Tags` 인수를 사용하여 태그를 추가합니다. AWS SDK for Python (Boto3) 자세한 내용은 [StartTranscriptionJob](#) 및 [Tag](#) 단원을 참조하세요.

기능별, 시나리오 및 서비스 간 예제를 포함하여 AWS SDK를 사용하는 추가 예제는 이 [SDK를 사용한 Amazon Transcribe의 코드 예제 AWS](#) 장을 참조하십시오.

```

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

while True:
    status = transcribe.get_transcription_job(TranscriptionJobName = job_name)

```

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

## 화자 파티셔닝(분할)

스피커 다이어라이제이션을 사용하면 트랜스크립션 출력에서 서로 다른 스피커를 구별할 수 있습니다. Amazon Transcribe 최대 30명의 고유 화자를 구분하고 각 화자의 텍스트에 고유한 값 (~까지) 으로 레이블을 지정할 수 있습니다. spk\_0 spk\_9

화자 파티셔닝이 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 파티셔닝(화자 2명)이 포함된 전체 트랜스크립트 예시를 보려면 [이분화 출력 예제 \(배치\)](#)를 참조하세요.

# 배치 트랜스크립션의 화자 파티셔닝

배치 트랜스크립션에서 화자를 파티셔닝하려면 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.

오디오 설정 패널에서 화자 파티셔닝('오디오 식별 유형' 제목 아래)을 선택합니다. 파티셔닝하려는 화자 수를 최대 10개까지 선택적으로 지정할 수 있습니다.

### 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 예제에서는 [start-transcription-job](#)를 사용합니다. 자세한 정보는 [StartTranscriptionJob](#)을 참조하세요.

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

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

다음은 [start-transcription-job](#) 명령을 사용하는 또 다른 예와 해당 작업에서 스피커 파티셔닝을 가능하게 하는 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 들어 있습니다.

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

## AWS SDK for Python (Boto3)

이 예제에서는 `run`을 사용하여 AWS SDK for Python (Boto3) [start\\_transcription\\_job](#) 메서드를 사용하여 채널을 식별합니다. 자세한 [StartTranscriptionJob](#) 내용은 [여기](#)를 참조하십시오.

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



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

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

## 스트리밍 트랜스크립션의 화자 파티셔닝

스트리밍 트랜스크립션에서 화자를 파티셔닝하려면 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.

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

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

**Transcription**

Download full transcript
🎤 Start streaming

Transcription output Current language: English, US

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

00:00 of 15:00 min audio stream

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

### 3. 화자 파티셔닝을 켭니다.

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

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

## HTTP/2 스트림

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

```
POST /stream-transcription HTTP/2
host: transcribestreaming.us-west-2.amazonaws.com
X-Amz-Target: com.amazonaws.transcribe.Transcribe.StartStreamTranscription
Content-Type: application/vnd.amazon.eventstream
X-Amz-Content-Sha256: string
X-Amz-Date: 20220208T235959Z
Authorization: AWS4-HMAC-SHA256 Credential=access-key/20220208/us-west-2/transcribe/
aws4_request, SignedHeaders=content-type;host;x-amz-content-sha256;x-amz-date;x-amz-
target;x-amz-security-token, Signature=string
x-amzn-transcribe-language-code: en-US
x-amzn-transcribe-media-encoding: flac
x-amzn-transcribe-sample-rate: 16000
x-amzn-transcribe-show-speaker-label: true
transfer-encoding: chunked
```

매개변수 정의는 [API 참조에서 찾을 수 있으며](#), 모든 AWS API 작업에 공통적인 매개변수는 [공통 매개변수](#) 섹션에 나열되어 있습니다.

## WebSocket 스트림

이 예시에서는 트랜스크립션 출력에서 화자를 구분하는 미리 서명된 URL을 생성합니다. 가독성을 높이기 위해 줄바꿈이 추가되었습니다. WebSocket 스트리밍을 사용하는 방법에 대한 자세한 내용은 Amazon Transcribe을 참조하십시오 [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=DICTION
&media-encoding=flac
&sample-rate=16000
&show-speaker-label=true
```

매개변수 정의는 [API 참조에서 찾을 수 있으며](#), 모든 AWS API 작업에 공통적인 [매개변수는 공통 매개변수](#) 섹션에 나열되어 있습니다.

## 이분화 출력 예제 (배치)

다음은 일괄변환이 활성화된 일괄 트랜스크립션의 출력 예제입니다.

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

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

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

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

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



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

## 멀티 채널 오디오 트랜스크립팅

오디오에 두 개의 채널이 있는 경우 채널 식별을 사용하여 각 채널의 음성을 개별적으로 변환할 수 있습니다. Amazon Transcribe 현재 2개 이상의 채널이 있는 오디오는 지원하지 않습니다.

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

[표준 트랜스크립트 섹션](#) (`transcripts` 및 `items`) 외에도 채널 식별이 활성화된 요청에는 `channel_labels` 섹션이 포함됩니다. 이 섹션에는 채널별로 그룹화된 각 발화 또는 문장 부호와 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 말하는 경우 각 채널의 타임스탬프가 서로 말을 하는 동안 해당 채널의 타임스탬프가 겹쳐집니다.

채널 식별이 포함된 전체 예제 스크립트를 보려면 [여기](#)를 참조하십시오 [채널 식별 출력 예시 \(배치\)](#).

## 일괄 트랜스크립션에서 채널 식별 사용

일괄 트랜스크립션에서 채널을 식별하려면 AWS Management Console, AWS CLI, 또는 AWSSDK를 사용할 수 있습니다. 예를 보려면 다음을 참조하십시오.

### 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 입력한 후 다음을 선택합니다. 그러면 작업 구성 - 옵션 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 예제에서는 [start-transcription-job](#)를 사용합니다. 자세한 정보는 [StartTranscriptionJob](#)을 참조하세요.

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

다음은 [start-transcription-job](#) 명령을 사용하는 또 다른 예와 해당 작업으로 채널 식별을 가능하게 하는 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 다음 요청 본문이 포함되어 있습니다.

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

## AWS SDK for Python (Boto3)

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

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

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

```
print(status)
```

## 스트리밍 트랜스크립션에서 채널 식별 사용

스트리밍 트랜스크립션에서 채널을 식별하려면 HTTP/2를 사용하거나 예를 보려면 다음을 참조하십시오 [WebSocket](#).

### HTTP/2 스트림

이 예시에서는 트랜스크립션 출력에서 채널을 구분하는 HTTP/2 요청을 생성합니다. 에서 HTTP/2 스트리밍을 사용하는 방법에 대한 자세한 내용은 [Amazon Transcribe 참조하십시오](#) [HTTP/2 스트림 설정](#). 관련 매개변수와 헤더에 대한 자세한 내용은 [Amazon Transcribe](#) 을 참조하십시오 [StartStreamTranscription](#).

```
POST /stream-transcription HTTP/2
host: transcribestreaming.us-west-2.amazonaws.com
X-Amz-Target: com.amazonaws.transcribe.Transcribe.StartStreamTranscription
Content-Type: application/vnd.amazon.eventstream
X-Amz-Content-Sha256: string
X-Amz-Date: 20220208T235959Z
Authorization: AWS4-HMAC-SHA256 Credential=access-key/20220208/us-west-2/transcribe/
aws4_request, SignedHeaders=content-type;host;x-amz-content-sha256;x-amz-date;x-amz-
target;x-amz-security-token, Signature=string
x-amzn-transcribe-language-code: en-US
x-amzn-transcribe-media-encoding: flac
x-amzn-transcribe-sample-rate: 16000
x-amzn-channel-identification: TRUE
transfer-encoding: chunked
```

파라미터 [정의는 API 레퍼런스에서](#) 찾을 수 있으며, 모든 AWS API 작업에 공통적인 [파라미터는 일반 파라미터](#) 섹션에 나열되어 있습니다.

### WebSocket 스트림

이 예시에서는 트랜스크립션 출력에서 채널을 구분하는 미리 서명된 URL을 만듭니다. 가독성을 높이기 위해 줄바꿈이 추가되었습니다. 에서 WebSocket 스트리밍을 사용하는 방법에 대한 자세한 내용은 [Amazon Transcribe](#) 을 참조하십시오 [WebSocket 스트림 설정](#). 매개변수에 대한 자세한 내용은 [Amazon Transcribe](#) 을 참조하십시오 [StartStreamTranscription](#).

```
GET wss://transcribestreaming.us-west-2.amazonaws.com:8443/stream-transcription-
websocket?
```

```

&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIAIOSFODNN7EXAMPLE%2F20220208%2Fus-
west-2%2Ftranscribe%2Faws4_request
&X-Amz-Date=20220208T235959Z
&X-Amz-Expires=300
&X-Amz-Security-Token=security-token
&X-Amz-Signature=string
&X-Amz-SignedHeaders=content-type%3Bhost%3Bx-amz-date
&language-code=en-US
&specialty=PRIMARYCARE
&type=DICTATION
&media-encoding=flac
&sample-rate=16000
&channel-identification=TRUE

```

파라미터 [정의는 API 레퍼런스에서](#) 찾을 수 있으며, 모든 AWS API 작업에 공통적인 [파라미터는 일반 파라미터](#) 섹션에 나열되어 있습니다.

## 채널 식별 출력 예시 (배치)

다음은 채널 식별이 활성화된 일괄 트랜스크립션의 출력 예제입니다.

```

{
  "jobName": "my-first-transcription-job",
  "accountId": "111122223333",
  "results": {
    "transcripts": [
      {
        "transcript": "I've been on hold for an hour. Sorry about that."
      }
    ],
    "channel_labels": {
      "channels": [
        {
          "channel_label": "ch_0",
          "items": [
            {
              "channel_label": "ch_0",
              "start_time": "4.86",
              "end_time": "5.01",
              "alternatives": [
                {
                  "confidence": "1.0",

```



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

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

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

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

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

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

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

## 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. 등의 Amazon Translate 다른 서비스를 사용하여 성적표를 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, AWS Python SDK를 사용한 사용 예제는 [을 참조하십시오](#) [배치 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-le, en-NZ, en-US, en-WL, en-ZA, es-ES, es-US, fr-CA, fr-Fr, zh-Cn, zh-TW, pt-Pt, de-ch, a-Ae, Da-DK, He-Il, 하이인, 아이디-아이디, 파에어, 잇-잇, Ja-jp, Ko-kr, MS-My, nl-nl, 루루, 타인, 테인, 테인, 티인, 티인, 티인, 티인, 티-티-R

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 언어에 일치시킬 때 지정한 언어만 사용하도록 Amazon Transcribe 제한되므로 언어 식별 속도가 빨라지고 올바른 언어 언어 지정과 관련된 정확성이 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 포함했지만 제공한 언어 코드 중 오디오에서 식별된 언어 또는 언어와 일치하지 않는 경우 지정된 언어 코드에서 가장 일치하는 언어를 Amazon Transcribe 선택합니다. 그런 다음 해당 언어로 트랜스크립트를 생성합니다. 예를 들어, 미디어가 미국 영어(en-US) Amazon Transcribe 로 되어 있고 해당 미디어가 독일어(de-DE)와 일치하는지 확인하는 언어 코드 zh-CN fr-FRde-DE, Amazon Transcribe 및 를 제공하여 독일어 자막을 작성할 수 있는 경우를 예로 들 수 있습니다. 언어 코드와 음성 언어가 일치하지 않으면 트랜스크립트가 정확하지 않을 수 있으므로 언어 코드를 포함할 때는 주의를 기울이는 것이 좋습니다.

## 언어 식별과 다른 Amazon Transcribe 기능 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 수정과 함께 언어 식별을 사용하는 경우 수정이 가능한 언어만 미국 영어 (en-US) 또는 미국 스페인어 (es-US) 로 제한됩니다. 자세한 내용은 [지원되는 언어 및 언어별 기능](#) 섹션을 참조하세요.

### Important

콘텐츠 교정이 활성화된 상태에서 자동 언어 식별을 사용하고 오디오에 미국 영어 (en-US) 또는 미국 스페인어 (es-US) 이외의 언어가 포함된 경우 미국 영어 또는 미국 스페인어 콘텐츠만 자막에서 수정됩니다. 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 en-US 사용자 지정 어휘를 지정했지만 미디어에서 사용되는 언어가 해당 언어라고 Amazon Transcribe 판단되면 사용자 지정 어휘는 *en-AU* 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 ▲

Q |

- 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 예제에서는 [start-transcription-job](#) 명령과 파라미터를 사용합니다. IdentifyLanguage 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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac \
--output-bucket-name DOC-EXAMPLE-BUCKET \
--output-key my-output-files/ \
--identify-language \ (or --identify-multiple-languages) \
--language-options "en-US" "hi-IN"
```

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

다음은 [start-transcription-job](#) 명령어를 사용하는 또 다른 예와 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. `LanguageOptions`은 선택 사항이며, 권장 사항은 아닙니다.

```
{
  "TranscriptionJobName": "my-first-transcription-job",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
  },
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",
  "OutputKey": "my-output-files/",
  "IdentifyLanguage": true, (or "IdentifyMultipleLanguages": true),
  "LanguageOptions": [
    "en-US", "hi-IN"
  ]
}
```

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

## AWS SDK for Python (Boto3)

[이 예제에서는 AWS SDK for Python \(Boto3\) 를 사용하여 start\\_transcription\\_job 메서드의 IdentifyLanguage 인수를 사용하여 파일의 언어를 식별합니다.](#) 자세한 내용은 [StartTranscriptionJob](#) 및 [LanguageIdSettings](#) 섹션을 참조하세요.

기능별, 시나리오 및 크로스 서비스 예제를 포함하여 AWS SDK를 사용하는 추가 예제는 이 장을 참조하십시오. [SDK를 사용한 Amazon Transcribe의 코드 예제 AWS](#)

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

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



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

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

## 스트리밍 트랜스크립션을 통한 언어 식별

스트리밍 언어 식별을 통해 미디어 스트림에서 사용되는 지배적 언어를 식별할 수 있습니다. Amazon Transcribe는 언어를 식별하기 위해 3초 이상의 음성을 필요로 합니다.

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. 트랜스크립트를 번역하는 데 Amazon Transcribe과 같은 다른 서비스를 사용할 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 코드 중 어떤 것도 오디오에서 식별된 언어와 일치하지 않는 경우 Amazon Transcribe는 지정된 언어 코드 중 가장 가깝게 일치하는 언어를 선택합니다. 그런 다음 해당 언어로 트랜스크립트를 생성합니다. 예를 들어 미디어가 미국 영어(en-US)로 되어 있고 Amazon Transcribe에 언어 코드 zh-CN, fr-FR 및 de-DE를 제공하면 Amazon Transcribe는 미디어를 독일어(de-DE)와 일치시켜 독일어 트랜스크립션을 작성할 수 있습니다. 언어 코드와 음성 언어가 일치하지 않으면 트랜스크립트가 정확하지 않을 수 있으므로 언어 코드를 포함할 때는 주의를 기울이는 것이 좋습니다.

미디어에 채널이 두 개 있는 경우 Amazon Transcribe는 각 채널에서 사용되는 지배적 언어를 식별할 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 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.

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

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

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

Transcription output **Current language: Automatic, Confidence: N/A**

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

00:00 of 15:00 min audio stream

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

3. 자동 언어 식별 또는 자동 다국어 식별을 선택합니다.

### ▼ Language settings

#### Language settings

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

**Specific language**

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

**Automatic language identification** [Info](#)

If you don't know the language spoken in your audio files, choose this option.

**Automatic multiple languages identification** [Info](#)

If there are multiple languages spoken in your audio files and you're not sure what these languages are, choose this option. This selection provides limited additional processing options compared to **Specific language**.

#### Language options for automatic language identification

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

Choose language(s) ▼

#### Preferred language - *optional*

Specify one preferred language from your previous selection.

Choose language ▼

▶ **Audio settings**

▶ **Content removal settings**

▶ **Customizations**

4. 트랜스크립션에 사용할 언어 코드를 최소 두 개 제공합니다. 언어당 방언을 하나만 제공할 수 있다는 점에 유의하세요. 예를 들어 동일한 트랜스크립션에 대해 en-US 및 fr-CA 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) X French, CA (fr-CA) X

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

Choose language ▲

None

English, US (en-US)

French, CA (fr-CA)

► Customizations

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

## HTTP/2 스트림

이 예시에서는 언어 식별이 활성화된 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-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. 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-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 있으며, 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. 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
&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. 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
&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 있으며, 모든 AWS API 작업에 공통적인 파라미터는 [공통 파라미터](#) 섹션에 나열되어 있습니다.

## 대체 트랜스크립션

오디오를 Amazon Transcribe 텍스트로 변환하면 동일한 트랜스크립트의 여러 버전이 생성되고 각 버전에 신뢰도 점수가 할당됩니다. 일반적인 트랜스크립션에서는 신뢰도 점수가 가장 높은 버전만 얻을 수 있습니다.

대체 트랜스크립션을 켜면 신뢰도가 낮은 다른 버전의 트랜스크립트가 Amazon Transcribe 반환됩니다. 최대 10개의 대체 트랜스크립션을 반환하도록 선택할 수 있습니다. Amazon Transcribe 식별된 것보다 더 많은 대안을 지정하면 실제 대안 수만 반환됩니다.

모든 대안은 동일한 트랜스크립션 출력 파일에 있으며 세그먼트 수준에서 표시됩니다. 세그먼트는 스피커가 바뀌거나 오디오가 일시 중지되는 등 음성이 자연스럽게 멈추는 것을 말합니다.

대체 트랜스크립션은 일괄 트랜스크립션에만 사용할 수 있습니다.

트랜스크립션 출력문의 구조는 다음과 같습니다. 타원 (... 코드 예제의) 는 간결함을 위해 콘텐츠가 제거된 위치를 나타냅니다.

1. 주어진 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 점수입니다.

```
"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 트랜스크립션은 트랜스크립션 출력segments 부분에 있습니다. 각 세그먼트에 대한 대안은 신뢰도 점수가 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 끝에 표시되는 상태입니다.

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

## 대체 트랜스크립션 요청

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

### 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**.

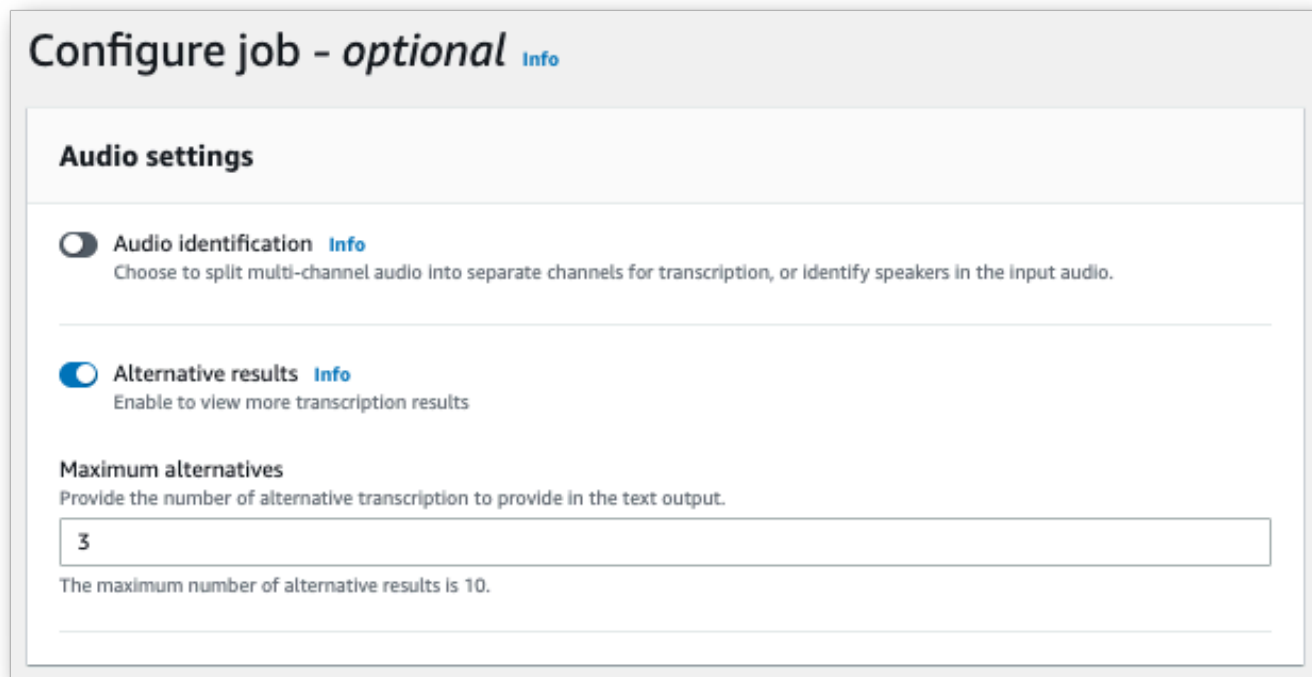
**Language**

Choose the language of the input audio.

▶ **Additional settings**

- 작업 세부 정보 지정 페이지에 포함하려는 필드를 모두 입력한 후 다음을 선택합니다. 그러면 작업 구성 - 옵션 페이지로 이동합니다.

대체 결과를 선택하고 성적표에 원하는 대체 트랜스크립션 결과의 최대 수를 지정합니다.



4. 작업 생성을 선택하여 트랜스크립션 작업을 실행합니다.

## AWS CLI

이 예제에서는 [start-transcription-job](#) 명령과 `ShowAlternatives` 매개 변수를 사용합니다. 자세한 내용은 [StartTranscriptionJob](#) 및 [ShowAlternatives](#) 단원을 참조하세요.

요청에 포함하는 `ShowAlternatives=true` 경우 반드시 포함해야 한다는 점에 유의하십시오. `MaxAlternatives`.

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

다음은 [start-transcription-job](#) 명령을 사용하는 또 다른 예와 대체 트랜스크립션이 포함된 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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"  
  },  
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",  
  "OutputKey": "my-output-files/",  
  "LanguageCode": "en-US",  
  "Settings": {  
    "ShowAlternatives": true,  
    "MaxAlternatives": 4  
  }  
}
```

## AWS SDK for Python (Boto3)

다음 예제에서는 AWS SDK for Python (Boto3) 를 사용하여 [start\\_transcription\\_job](#) 메서드의 ShowAlternatives 인수를 사용하여 대체 트랜스크립션을 요청합니다. 자세한 내용은 [StartTranscriptionJob](#) 및 [ShowAlternatives](#) 단원을 참조하세요.

기능별, 시나리오 및 서비스 간 예제를 포함하여 AWS SDK를 사용하는 추가 예제는 이 [SDK를 사용한 Amazon Transcribe의 코드 예제 AWS](#) 장을 참조하십시오.

요청에 포함하는 'ShowAlternatives': True 경우 반드시 포함해야 한다는 점에 유의하십시오. MaxAlternatives.

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

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

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



# 맞춤형 어휘와 맞춤형 언어 모델을 통한 트랜스크립션 정확도 향상

미디어에 브랜드 이름, 두문자어, 전문 용어, 전문 용어 등 도메인 관련 용어나 비표준 용어가 포함된 경우 트랜스크립션 출력물에 해당 용어가 제대로 표시되지 Amazon Transcribe 않을 수 있습니다.

트랜스크립션의 부정확한 내용을 수정하고 특정 사용 사례에 맞게 출력을 사용자 지정하려면 [사용자 지정 어휘](#) 및 [를 만들 수 있습니다](#) [사용자 지정 언어 모델](#).

- [사용자 지정 어휘](#) 모든 상황에서 특정 단어의 인식과 서식을 조정하고 강화하도록 설계되었습니다. 여기에는 단어 Amazon Transcribe 제공과 선택적으로 발음 및 표시 양식이 포함됩니다.

성적표의 특정 용어가 제대로 표현되지 않는 경우 Amazon Transcribe 해당 용어를 Amazon Transcribe 어떻게 표시할지 알려주는 사용자 지정 어휘 파일을 만들 수 있습니다. 이러한 단어별 접근 방식은 브랜드 이름 및 두문자어와 같은 용어를 수정하는 데 가장 적합합니다.

- [사용자 지정 언어 모델](#) 용어와 관련된 컨텍스트를 캡처하도록 설계되었습니다. 여기에는 Amazon Transcribe 대량의 도메인별 텍스트 데이터를 제공하는 것이 포함됩니다.

기술 용어를 올바르게 표현하지 못하거나 스크립트에서 잘못된 동음이의어를 사용하는 경우 Amazon Transcribe 도메인별 언어를 가르치는 Amazon Transcribe 사용자 지정 언어 모델을 만들 수 있습니다. 예를 들어 사용자 지정 언어 모델은 'floe' (아이스 플루)와 'flow' (선형 흐름)를 사용하는 시기를 학습할 수 있습니다.

이러한 상황 인식 접근 방식은 대량의 영역별 음성을 필사하는 데 가장 적합합니다. 사용자 지정 언어 모델은 사용자 지정 어휘만 사용하는 것보다 정확도가 크게 향상될 수 있습니다. 일괄 트랜스크립션을 사용하는 경우 요청에 사용자 지정 언어 모델과 사용자 지정 어휘를 모두 포함할 수 있습니다.

## Tip

필사 정확도를 극대화하려면 사용자 지정 어휘를 사용자 지정 언어 모델과 함께 사용하십시오.

를 사용하여 사용자 지정 어휘를 만드는 방법에 대한 비디오 데모를 AWS Management Console 보려면 [사용자 지정 어휘 사용](#)을 참조하십시오.

사용자 지정 언어 모델을 만들고 사용하는 방법에 대한 비디오 데모를 보려면 사용자 [지정 언어 모델 \(CLM\)](#) 을 사용하여 트랜스크립션 정확도를 높이세요.

### AWSMachine Learning 블로그를 통해 더 자세히 알아보기

#### Custom 커스터마이징

- [를 사용한 F1 레이스의 라이브 트랜스크립션Amazon Transcribe](#)

#### 사용자 지정 언어 모델:

- [speech-to-text 성능을 극대화하기 위한 사용자 지정 언어 모델 구축Amazon Transcribe](#)
- [맞춤형 언어 모델을 사용하여 수업 강의의 필사 정확도를 높입니다.Amazon Transcribe](#)

## 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 Transcribe 가 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:

- 사용자 지정 어휘 파일은 한 개당 최대 100개까지 만들 수 있습니다. AWS 계정
- 각 사용자 지정 어휘 파일의 크기 한도는 50Kb입니다.
- API를 사용하여 사용자 지정 어휘를 만드는 경우 어휘 파일은 텍스트(\*.txt) 형식이어야 합니다. 를 사용하는 경우 어휘 파일은 텍스트 (\*.txt) 형식 또는 쉼표로 구분된 값 (\*.csv) 형식일 수 있습니다. AWS Management Console
- 사용자 지정 어휘 내의 각 항목은 256자를 초과할 수 없습니다.
- 사용자 지정 어휘를 사용하려면 필사본과 동일하게 생성되어야 합니다. AWS 리전

**i** Tip

를 사용하여 사용자 지정 어휘를 테스트할 수 있습니다. AWS Management Console 사용자 지정 어휘를 사용할 준비가 되면 에 로그인하고 실시간 트랜스크립션을 선택하고 사용자 지정으로 스크롤한 다음 사용자 지정 어휘를 켜고 드롭다운 목록에서 사용자 지정 어휘를 선택합니다. AWS Management Console 그런 다음 스트리밍 시작을 선택합니다. 사용자 지정 어휘의 일부 단어를 마이크에 대고 말하여 제대로 렌더링되는지 확인합니다.

## 사용자 지정 어휘 테이블과 목록 비교

**A** 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 같은 방식으로 사용자 지정 어휘 표를 사용합니다. 목록은 각 방법마다 다르게 사용되므로 두 방법 간에 성공적으로 사용하려면 추가 서식이 필요할 수 있습니다.

자세한 내용은 [테이블을 사용하여 사용자 지정 어휘 생성 및 목록을 사용하여 사용자 지정 어휘 만들기](#) 섹션을 참조하세요.

Amazon Augmented AI를 사용자 지정 어휘와 함께 사용하는 방법을 좀 더 자세히 알아보려면 [Amazon Transcribe와 함께 인적 검토 구축 시작](#)을 참조하세요.

**i** 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 사용자 지정 어휘에서는 더 이상 지원되지 않습니다. 열을 비워 두십시오.	IPA 사용자 지정 어휘는 더 이상 지원되지 않습니다. 열을 비워 두십시오. 이 열의 모든 값은 무시됩니다.	선택 사항입니다. 이 열의 행은 비워 둘 수 있습니다.
이 열에는 스페이스를 사용하지 마세요.	이 열의 모든 값은 무시됩니다. 향후 이 컬럼에 대한 지원을 제거할 예정입니다.	향후 이 컬럼에 대한 지원을 제거할 예정입니다.	이 열에는 스페이스를 사용할 수 있습니다.
항목에 여러 단어가 포함된 경우 각 단어를 하이픈(-)으로 구분합니다. 예: <b>Andorra-la-Vella</b> 또는 <b>Los-Angeles</b> .			트랜스크립션 출력에서 항목이 어떻게 보이길 원하는지 정의합니다. 예를 들어, Phrase 열의 <b>Andorra-la-Vella</b> 는 DisplayAs 열의 <b>Andorra la Vella</b> 입니다.
두문자어의 경우 발음되는 모든 문자를 마침표로 구분해야 합니다. 마지막 마침표도 발음해야 합니다. 두문자어가 복수형인 경우 두문자어와 's' 사이에 하이픈을 사용해야 합니다. 예를 들어, 'CLI'는 <b>C.L.I.</b> ( <b>C.L.I</b> 아님)이고 'ABC'는 <b>A.B.C.-s</b> ( <b>A.B.C-s</b> 아님)입니다.			이 열의 행이 비어 있는 경우 Phrase 열의 내용을 Amazon Transcribe 사용하여 결과를 결정합니다.
구절이 단어와 두문자어로 구성된 경우			이 열에 숫자(0-9)를 포함할 수 있습니다.

구절	SoundsLike	IPA	DisplayAs
이 두 구성 요소를 하이픈으로 구분해야 합니다. 예를 들어, 'DynamoDB'는 <b>Dynamo-D.B.</b> 입니다.			
이 열에 숫자를 포함하지 마세요. 숫자는 철자로 입력해야 합니다. 예를 들어, 'VX02Q'는 <b>V.X.-zero-two-Q.</b> 입니다.			

테이블을 만들 때 참고할 사항:

- 테이블에는 네 개의 열 머리글이 (Phrase, SoundsLike, IPA, and DisplayAs) 모두 포함되어야 합니다. Phrase 열에는 각 행에 항목이 하나씩 있어야 합니다. 를 통해 IPA 발음 입력을 제공하는 기능은 더 이상 지원되지 않으므로 열을 비워 두어도 됩니다. SoundsLike 이 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 사용자 지정 SoundsLike 어휘에는 더 이상 지원되지 않습니다. 열을 비워 두십시오. 이 열의 모든 값은 무시됩니다. 향후 이 칼럼에 대한 지원을 제거할 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 를 사용하여 [CreateVocabulary](#) 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](#)을 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.

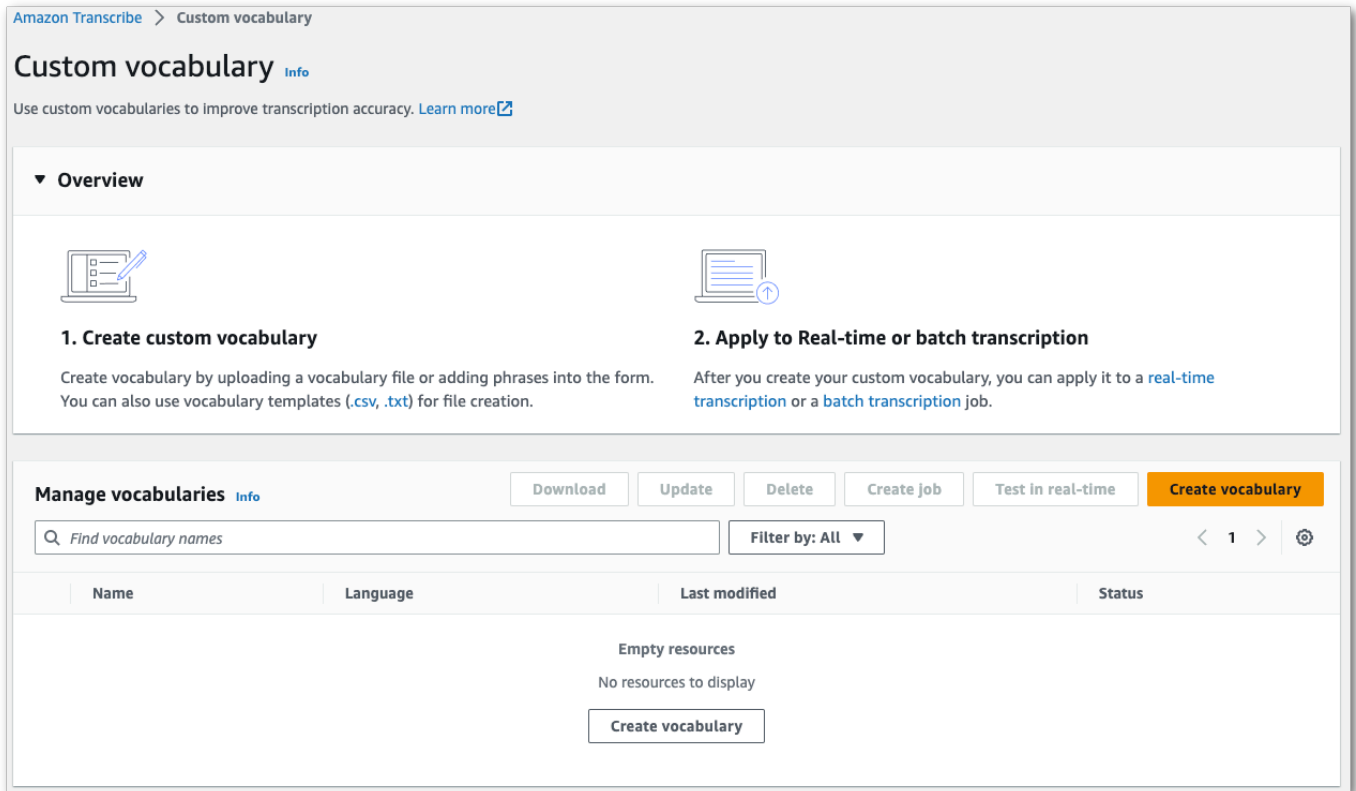
Phrase	[TAB]SoundsLike	[TAB]IPA	[TAB]DisplayAs
Los-Angeles	[TAB]	[TAB]	[TAB]Los Angeles
Eva-Maria	[TAB]	[TAB]	[TAB]
A.B.C.-s	[TAB]	[TAB]	[TAB]ABCs
amazon-dot-com	[TAB]	[TAB]	[TAB]amazon.com
C.L.I.	[TAB]	[TAB]	[TAB]CLI
Andorra-la-Vella	[TAB]	[TAB]	[TAB]Andorra la Vella
Dynamo-D.B.	[TAB]	[TAB]	[TAB]DynamoDB
V.X.-zero-two	[TAB]	[TAB]	[TAB]VX02
V.X.-zero-two-Q.	[TAB]	[TAB]	[TAB]VX02Q

## 사용자 지정 어휘 테이블 생성

에서 사용할 사용자 지정 어휘 표를 처리하려면 다음 예를 참조하십시오. Amazon Transcribe

## AWS Management Console

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

**Import from file**

File format: txt, csv, maximum size 50 KB.

b. 특정 위치에서 txt 또는 csv 파일을 가져옵니다. Amazon S3

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 >

<input type="checkbox"/>	<a href="#">Phrase</a>	<a href="#">SoundsLike (optional)</a>	<a href="#">IPA (optional)</a>	<a href="#">DisplayAs (optional)</a>
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 ▼

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

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

Add row

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 ▼

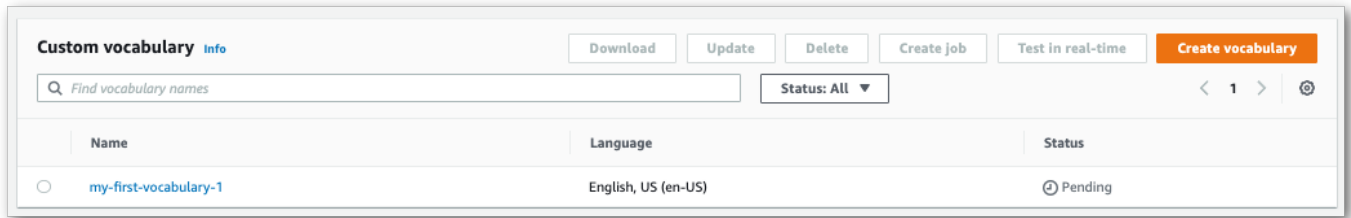
Q Filter Phrase, SoundsLike, IPA or DisplayAs Show all ▼ < 1 >

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

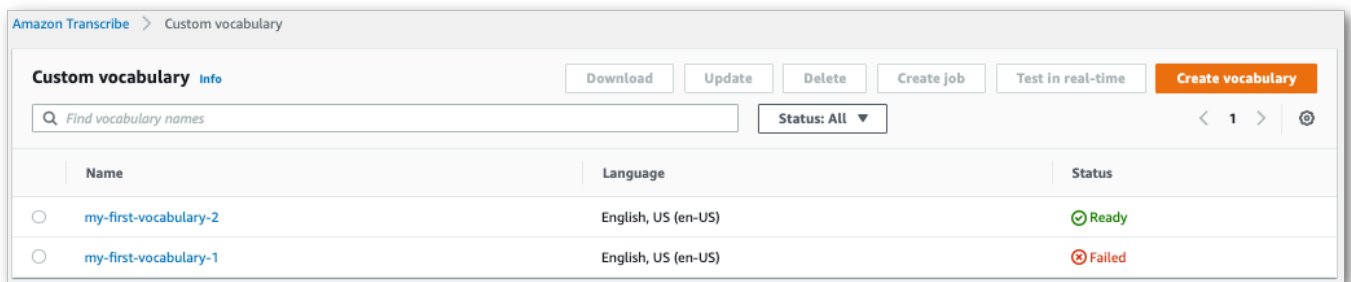
Add row

변경 내용을 저장하려면 체크 표시(✓)를 선택하고 변경 내용을 삭제하려면 'X'를 선택합니다.

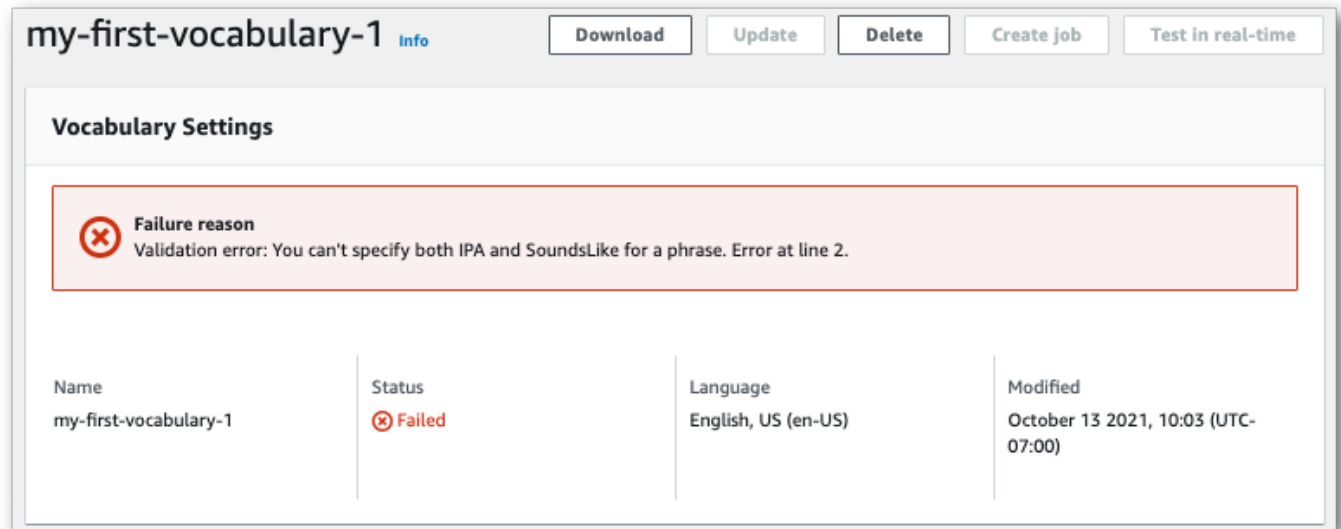
5. 필요에 따라 사용자 지정 어휘에 태그를 추가합니다. 모든 필드를 작성하고 어휘에 만족하면 페이지 하단에서 어휘 생성을 선택합니다. 그러면 사용자 지정 어휘 페이지로 돌아가서 사용자 지정 어휘의 상태를 볼 수 있습니다. 상태가 '보류 중'에서 '준비'로 변경되면 사용자 지정 어휘를 트랜스크립션과 함께 사용할 수 있습니다.



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 정보는 [CreateVocabulary](#)을 참조하세요.

트랜스크립션 작업에서 기존 사용자 지정 어휘를 사용하려면 작업을 호출할 때 [Settings](#) 필드에 를 설정하거나 에서 드롭다운 목록에서 사용자 지정 어휘를 선택하십시오. [VocabularyName](#) [StartTranscriptionJob](#) AWS Management Console

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

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 다음 요청 본문이 포함되어 있습니다.

```
{
  "VocabularyName": "my-first-vocabulary",
  "VocabularyFileUri": "s3://DOC-EXAMPLE-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 예제에서는 AWS SDK for Python (Boto3) 를 사용하여 [create\\_vocabulary](#) 메서드를 사용하여 테이블에서 사용자 지정 어휘를 생성합니다. 자세한 정보는 [CreateVocabulary](#)을 참조하세요.

트랜스크립션 작업에서 기존 사용자 정의 어휘를 사용하려면 작업을 호출할 때 [Settings](#) 필드에 를 설정하거나 에서 드롭다운 목록에서 사용자 지정 어휘를 선택하십시오. [VocabularyName StartTranscriptionJob](#) AWS Management Console

기능별, 시나리오 및 크로스 서비스 예제를 포함하여 AWS SDK를 사용하는 추가 예제는 이 장을 참조하십시오. [SDK를 사용한 Amazon Transcribe의 코드 예제 AWS](#)

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

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

### Note

사용자 지정 어휘 파일을 위한 새 Amazon S3 버킷을 만드는 경우 [CreateVocabulary](#) 요청을 하는 IAM 역할에 이 버킷에 액세스할 수 있는 권한이 있는지 확인하세요. 역할에 올바른 권한이 없는 경우 요청이 실패합니다. `DataAccessRoleArn` 파라미터를 포함하여 요청 내에서 IAM 역할을 지정할 수도 있습니다. 의 IAM 역할 및 정책에 대한 자세한 내용은 Amazon Transcribe을 참조하십시오 [Amazon Transcribe 자격 증명 기반 정책 예제](#).

## 목록을 사용하여 사용자 지정 어휘 만들기

### Important

목록 형식의 사용자 지정 어휘는 더 이상 사용되지 않으므로 새 사용자 지정 어휘를 만들려면 표 형식을 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 만들어 업로드해야 합니다. 줄로 구분된 항목이나 쉼표로 구분된 항목을 사용할 수 있습니다. 목록은 텍스트 (\*.txt) 형식의 파일로 저장해야 합니다. LF 등의 다른 형식을 사용하는 CRLF 경우 사용자 지정 어휘는 에서 허용되지 않습니다 Amazon Transcribe.
- AWS CLI 및 AWSSDK: 플래그를 사용하여 API 호출 내에 쉼표로 구분된 항목으로 사용자 지정 어휘를 포함해야 합니다. [Phrases](#)

항목에 여러 단어가 포함된 경우 각 단어에 하이픈을 넣어야 합니다. 예를 들어 '로스앤젤레스'를, '안도라 라 **Los-Angeles** 베야'를 차례로 포함합니다. **Andorra-la-Vella**

다음은 두 가지 유효한 목록 형식의 예입니다. 메서드별 [맞춤 어휘 목록 만들기](#) 예는 를 참조하십시오.

- 쉼표로 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 참조하십시오.

사용자 지정 어휘 목록은 [CreateMedicalVocabulary](#) 작업에서 지원되지 않습니다. 사용자 지정 의학 어휘를 만들려면 표 형식을 사용해야 합니다. 자세한 [테이블을 사용하여 사용자 지정 어휘 생성](#) 내용은 을 참조하십시오.

**맞춤 어휘 목록 만들기**

와 함께 Amazon Transcribe 사용할 사용자 지정 어휘 목록을 처리하려면 다음 예를 참조하십시오.

**AWS CLI**

이 예제에서는 [create-vocabulary](#) 명령을 목록 형식의 사용자 지정 어휘 파일과 함께 사용합니다. 자세한 정보는 [CreateVocabulary](#)을 참조하세요.

```
aws transcribe create-vocabulary \
  --vocabulary-name my-first-vocabulary \
  --language-code en-US \
  --phrases {CLI,Eva-Maria,ABCs}
```

다음은 [create-vocabulary 명령어](#)와 사용자 지정 어휘를 생성하는 요청 본문을 사용하는 또 다른 예제입니다.

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

my-first-vocab-list.json 파일에는 다음과 같은 요청 본문이 들어 있습니다.

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

에서 PENDING 로 VocabularyState READY 변경되면 맞춤형 어휘를 트랜스크립션과 함께 사용할 수 있습니다. 사용자 지정 어휘의 현재 상태를 확인합니다.

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

## AWS SDK for Python (Boto3)

이 예제에서는 [create\\_vocabulary](#) 메서드를 사용하여 목록에서 사용자 지정 어휘를 만듭니다. AWS SDK for Python (Boto3) 자세한 정보는 [CreateVocabulary](#)을 참조하세요.

기능별, 시나리오 및 크로스 서비스 예제를 포함하여 AWS SDK를 사용하는 추가 예제는 이 장을 참조하십시오. [SDK를 사용한 Amazon Transcribe의 코드 예제 AWS](#)

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

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

### Note

사용자 지정 어휘 파일을 위한 새 Amazon S3 버킷을 만드는 경우 [CreateVocabulary](#) 요청을 하는 IAM 역할에 이 버킷에 액세스할 권한이 있는지 확인하세요. 역할에 올바른 권한이 없으면 요청이 실패합니다. `DataAccessRoleArn` 파라미터를 포함하여 요청 내에서 IAM 역할을



지정할 수도 있습니다. 의 IAM 역할 및 정책에 대한 자세한 내용은 Amazon Transcribe 을 참조하십시오 [Amazon Transcribe 자격 증명 기반 정책 예제](#).

## 사용자 지정 어휘 사용

사용자 지정 어휘가 생성되면 이를 트랜스크립션 요청에 포함시킬 수 있습니다. 예를 보려면 다음 섹션을 참조하십시오.

요청에 포함하는 사용자 지정 어휘의 언어는 미디어에 지정한 언어 코드와 일치해야 합니다. 언어가 일치하지 않으면 사용자 지정 어휘가 트랜스크립션에 적용되지 않으며 경고나 오류가 발생하지 않습니다.

### 일괄 트랜스크립션에 사용자 지정 어휘 사용

일괄 트랜스크립션과 함께 사용자 지정 어휘를 사용하려면 다음 예제를 참조하십시오.

#### 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 포함시킨 후 다음을 선택합니다.

3. 작업 구성 페이지 하단의 사용자 지정 패널에서 사용자 지정 어휘를 켜십시오.

## 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 예제에서는 [start-transcription-job](#) 명령과 Settings 매개 변수를 VocabularyName 하위 매개 변수와 함께 사용합니다. 자세한 내용은 [StartTranscriptionJob](#) 및 [Settings](#) 단원을 참조하세요.

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

다음은 [start-transcription-job](#) 명령을 사용하는 또 다른 예와 해당 작업에 대한 사용자 지정 어휘가 포함된 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 다음 요청 본문이 포함되어 있습니다.

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

## AWS SDK for Python (Boto3)

이 예제에서는 AWS SDK for Python (Boto3) 를 사용하여 [start\\_transcription\\_job](#) 메서드의 Settings 인수를 사용하여 사용자 지정 어휘를 포함합니다. 자세한 내용은 [StartTranscriptionJob](#) 및 [Settings](#) 단원을 참조하세요.

기능별, 시나리오 및 서비스 간 예제를 포함하여 AWS SDK를 사용하는 추가 예제는 이 [SDK를 사용한 Amazon Transcribe의 코드 예제 AWS](#) 장을 참조하십시오.

```

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

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

```

## 스트리밍 트랜스크립션에서 사용자 지정 어휘 사용하기

스트리밍 트랜스크립션과 함께 사용자 지정 어휘를 사용하려면 다음 예제를 참조하십시오.

### AWS Management Console

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창에서 실시간 트랜스크립션을 선택합니다. 아래로 스크롤하여 사용자 지정으로 이동하고 이 필드가 최소화된 경우 확장하십시오.

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

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

**Transcription** Download full transcript Start streaming

Transcription output Current language: English, US

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

00:00 of 15:00 min audio stream

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

3. 사용자 지정 어휘를 켜고 드롭다운 메뉴에서 사용자 지정 어휘를 선택합니다.

▼ Customizations

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

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

Choose a vocabulary ▼

스트림에 적용하려는 기타 설정을 포함합니다.

4. 이제 스트림을 트랜스크립션 할 준비가 되었습니다. 스트리밍 시작을 선택하고 말하기 시작하세요. 받아쓰기를 종료하려면 스트리밍 중지를 선택합니다.

## HTTP/2 스트림

이 예제는 사용자 지정 어휘를 포함하는 HTTP/2 요청을 생성합니다. 에서 HTTP/2 스트리밍을 사용하는 방법에 대한 자세한 내용은 [Amazon Transcribe 참조하십시오](#) [HTTP/2 스트림 설정](#). 관련 매개변수 및 헤더에 대한 자세한 내용은 [Amazon Transcribe](#) 을 참조하십시오 [StartStreamTranscription](#).

```
POST /stream-transcription HTTP/2
host: transcribestreaming.us-west-2.amazonaws.com
X-Amz-Target: com.amazonaws.transcribe.Transcribe.StartStreamTranscription
Content-Type: application/vnd.amazon.eventstream
X-Amz-Content-Sha256: string
X-Amz-Date: 20220208T235959Z
Authorization: AWS4-HMAC-SHA256 Credential=access-key/20220208/us-west-2/transcribe/aws4_request, SignedHeaders=content-type;host;x-amz-content-sha256;x-amz-date;x-amz-target;x-amz-security-token, Signature=string
x-amzn-transcribe-language-code: en-US
x-amzn-transcribe-media-encoding: flac
x-amzn-transcribe-sample-rate: 16000
x-amzn-transcribe-vocabulary-name: my-first-vocabulary
transfer-encoding: chunked
```

매개변수 정의는 [API 참조에서](#) 찾을 수 있습니다. 모든 AWS API 작업에 공통적인 [매개변수는 일반 매개변수](#) 섹션에 나열되어 있습니다.

## WebSocket 스트림

이 예시에서는 사용자 지정 어휘를 WebSocket 스트림에 적용하는 미리 서명된 URL을 만듭니다. 가독성을 높이기 위해 줄바꿈이 추가되었습니다. 에서 WebSocket 스트리밍을 사용하는 방법에 대한 자세한 내용은 [Amazon Transcribe](#) 을 참조하십시오 [WebSocket 스트림 설정](#). 매개변수에 대한 자세한 내용은 [StartStreamTranscription](#) 을 참조하십시오 [StartStreamTranscription](#).

```
GET wss://transcribestreaming.us-west-2.amazonaws.com:8443/stream-transcription-websocket?
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIAIOSFODNN7EXAMPLE%2F20220208%2Fus-west-2%2Ftranscribe%2Faws4_request
&X-Amz-Date=20220208T235959Z
&X-Amz-Expires=300
&X-Amz-Security-Token=security-token
&X-Amz-Signature=string
&X-Amz-SignedHeaders=content-type%3Bhost%3Bx-amz-date
```

```
&language-code=en-US
&media-encoding=flac
&sample-rate=16000
&vocabulary-name=my-first-vocabulary
```

매개변수 [정의는 API 참조에서](#) 찾을 수 있습니다. 모든 AWS API 작업에 공통적인 [매개변수는 일반 매개변수](#) 섹션에 나열되어 있습니다.

## 사용자 지정 언어 모델

사용자 지정 언어 모델은 도메인별 음성의 트랜스크립션 정확도를 향상시키도록 설계되었습니다. 여기에는 일상적이고 일상적인 대화에서 들을 수 있는 내용 이외의 모든 콘텐츠가 포함됩니다. 예를 들어, 과학 컨퍼런스의 논문집을 필사하는 경우 표준 필사본으로는 발표자가 사용하는 많은 과학 용어를 인식하지 못할 수 있습니다. 이 경우 해당 분야에서 사용되는 특수 용어를 인식하도록 사용자 지정 언어 모델을 학습시킬 수 있습니다.

힌트 (예: 발음) 를 제공하여 단어 인식률을 높이는 사용자 지정 어휘와 달리 사용자 지정 언어 모델은 주어진 단어와 관련된 컨텍스트를 학습합니다. 여기에는 단어가 언제 어떻게 사용되는지, 단어와 다른 단어와의 관계가 포함됩니다. 예를 들어 기후 과학 연구 논문을 사용하여 모델을 학습시키면 모델은 '얼음 흐름'보다 '빙류'가 단어 쌍일 가능성이 더 높다는 것을 알게 될 수 있습니다.

사용자 지정 언어 모델에 지원되는 언어를 보려면 [참조하십시오 지원되는 언어 및 언어별 기능](#). 요청에 사용자 지정 언어 모델을 포함하는 경우 언어 식별을 활성화할 수 없습니다 (언어 코드를 지정해야 함).

### 사용자 지정 언어 모델 전용 API 작업

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

## 데이터 원본

원하는 모든 유형의 텍스트 데이터를 사용하여 모델을 학습시킬 수 있습니다. 그러나 텍스트 내용이 오디오 콘텐츠에 가까울수록 모델의 정확도가 높아집니다. 따라서 오디오와 동일한 컨텍스트에서 동일한 용어를 사용하는 텍스트 데이터를 선택하는 것이 중요합니다.

모델 학습에 가장 적합한 데이터는 정확한 성적표입니다. 이는 도메인 내 데이터로 간주됩니다. 도메인 내 텍스트 데이터는 트랜스크립션하려는 오디오와 용어, 용법 및 컨텍스트가 동일합니다.



정확한 녹취록이 없는 경우 저널 기사, 기술 보고서, 백서, 컨퍼런스 자료집, 사용 설명서, 뉴스 기사, 웹사이트 콘텐츠 및 오디오와 유사한 상황에서 사용되는 원하는 용어가 포함된 기타 텍스트를 사용하십시오. 이는 도메인 관련 데이터로 간주됩니다.

강력한 사용자 지정 언어 모델을 만들려면 상당한 양의 텍스트 데이터가 필요할 수 있으며, 여기에는 오디오에서 사용되는 용어가 포함되어야 합니다. 모델 학습을 위해 최대 2GB의 텍스트 데이터를 제공할 수 있습니다. 이를 학습 데이터라고 합니다. 필요에 따라 도메인 내 트랜스크립트가 없거나 적은 경우 최대 200MB의 텍스트 데이터를 제공하여 Amazon Transcribe 모델을 튜닝할 수 있습니다. 이를 튜닝 데이터라고 합니다.

## 트레이닝 vs. 튜닝 데이터

학습 데이터의 목적은 새로운 용어를 인식하고 이러한 용어가 사용되는 맥락을 Amazon Transcribe 학습하도록 가르치는 것입니다. 견고한 모델을 만들려면 대량의 관련 텍스트 데이터가 Amazon Transcribe 필요할 수 있습니다. 최대 2GB 한도까지 가능한 한 많은 훈련 데이터를 제공하는 것이 좋습니다.

데이터 조정의 목적은 훈련 데이터에서 학습한 상황별 관계를 구체화하고 최적화하는 데 있습니다. 사용자 지정 언어 모델을 생성할 수 있는 튜닝 데이터는 필요하지 않습니다.

학습을 선택하고 선택적으로 데이터를 튜닝하는 최선의 방법을 결정하는 것은 사용자의 몫입니다. 각 사례는 고유하며 보유한 데이터의 유형과 양에 따라 달라집니다. 도메인 내 학습 데이터가 부족한 경우 데이터를 조정하는 것이 좋습니다.

두 데이터 유형을 모두 포함하려는 경우 훈련 데이터와 조정 데이터를 중첩하지 마십시오. 훈련 데이터와 조정 데이터는 고유해야 합니다. 데이터가 중복되면 사용자 지정 언어 모델이 편향되고 왜곡되어 정확성에 영향을 미칠 수 있습니다.

일반적인 지침으로 가능하면 정확한 도메인 내 텍스트를 학습 데이터로 사용하는 것이 좋습니다. 다음은 선호도에 따라 나열된 몇 가지 일반적인 시나리오입니다.

- 10,000단어 이상의 정확한 도메인 내 트랜스크립트 텍스트가 있는 경우 이를 학습 데이터로 사용하십시오. 이 경우 튜닝 데이터를 포함할 필요가 없습니다. 사용자 지정 언어 모델을 학습할 수 있는 이상적인 시나리오입니다.
- 10,000단어 미만인 정확한 도메인 내 트랜스크립트 텍스트가 있지만 원하는 결과를 얻지 못하는 경우 기술 보고서와 같은 도메인 관련 서면 텍스트로 교육 데이터를 보강하는 것이 좋습니다. 이 경우 도메인 내 트랜스크립트 데이터의 일부 (10~25%)를 튜닝 데이터로 사용하도록 예약해 두십시오.

- 도메인 내 트랜스크립트 텍스트가 없는 경우 도메인 관련 텍스트를 모두 학습 데이터로 업로드하세요. 이 경우 서면 텍스트보다 스크립트 스타일의 텍스트를 사용하는 것이 좋습니다. 이는 사용자 지정 언어 모델 학습에 가장 효과적인 시나리오입니다.

모델을 생성할 준비가 되었을 때 [여기](#)를 참조하십시오 사용자 지정 언어 모델 만들기.

## 사용자 지정 언어 모델 만들기

사용자 지정 언어 모델을 만들려면 먼저 다음을 수행해야 합니다.

- 데이터 준비 데이터는 일반 텍스트 형식으로 저장해야 하며 특수 문자를 포함할 수 없습니다.
- 데이터를 Amazon S3 버킷에 업로드합니다. 학습 및 조정 데이터를 위한 별도의 폴더를 만드는 것이 좋습니다.
- Amazon S3 버킷에 대한 액세스 Amazon Transcribe 권한이 있는지 확인하세요. 데이터를 사용하려면 액세스 권한이 있는 IAM 역할을 지정해야 합니다.

## 데이터 준비

모든 데이터를 하나의 파일에 컴파일하거나 여러 파일로 저장할 수 있습니다. 튜닝 데이터를 포함하도록 선택한 경우 해당 데이터를 훈련 데이터와 별도의 파일에 저장해야 합니다.

훈련 또는 데이터 튜닝에 사용하는 텍스트 파일의 수는 중요하지 않습니다. 100,000개의 단어가 포함된 파일 하나를 업로드하면 10,000개의 단어가 포함된 파일 10개를 업로드하는 것과 동일한 결과를 얻을 수 있습니다. 가장 편리한 방식으로 텍스트 데이터를 준비하세요.

모든 데이터 파일이 다음 기준을 충족하는지 확인합니다.

- 모두 만들려는 모델과 동일한 언어로 되어 있습니다. 예를 들어 오디오를 미국 영어(en-US)로 변환하는 사용자 지정 언어 모델을 만들려면 모든 텍스트 데이터가 미국 영어로 되어 있어야 합니다.
- UTF-8 인코딩의 일반 텍스트 형식입니다.
- HTML 태그와 같은 특수 문자나 서식은 포함되지 않습니다.
- 최대 합산 크기는 훈련 데이터의 경우 2GB, 튜닝 데이터의 경우 200MB입니다.

이 기준 중 하나라도 충족되지 않으면 모델이 실패합니다.

## 데이터 업로드

데이터를 업로드하기 전에 훈련 데이터를 저장할 새 폴더를 만드세요. 조정 데이터를 사용하는 경우 별도의 폴더를 하나 더 생성하십시오.

버킷의 URI는 다음과 같을 수 있습니다.

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

학습 및 튜닝 데이터를 적절한 버킷에 업로드합니다.

나중에 이러한 버킷에 데이터를 더 추가할 수 있습니다. 하지만 그럴 경우 새 데이터로 모델을 다시 만들어야 합니다. 기존 모델은 새 데이터로 업데이트할 수 없습니다.

## 데이터에 대한 액세스 허용

사용자 지정 언어 모델을 만들려면 Amazon S3 버킷에 액세스할 권한이 있는 IAM 역할을 지정해야 합니다. 학습 데이터를 배치한 Amazon S3 버킷에 액세스할 수 있는 역할이 아직 없는 경우 역할을 만들어야 합니다. 역할을 생성한 후 해당 역할에 정책을 연결해 해당 역할을 연결할 수 있습니다. 사용자에게 정책을 연결하지 마십시오.

예제 정책은 [Amazon Transcribe 자격 증명 기반 정책 예제](#) 섹션을 참조하세요.

새 IAM ID를 만드는 방법을 알아보려면 ID ([사용자, 사용자 그룹 및 역할](#)) 를 참조하십시오IAM.

정책에 대한 자세한 내용은 다음을 참조하세요.

- [의 정책 및 권한 IAM](#)
- [IAM정책 생성](#)
- [AWS 리소스에 대한 액세스 관리](#)

## 사용자 지정 언어 모델 만들기

사용자 지정 언어 모델을 만들 때는 기본 모델을 선택해야 합니다. 기본 모델 옵션에는 두 가지가 있습니다.

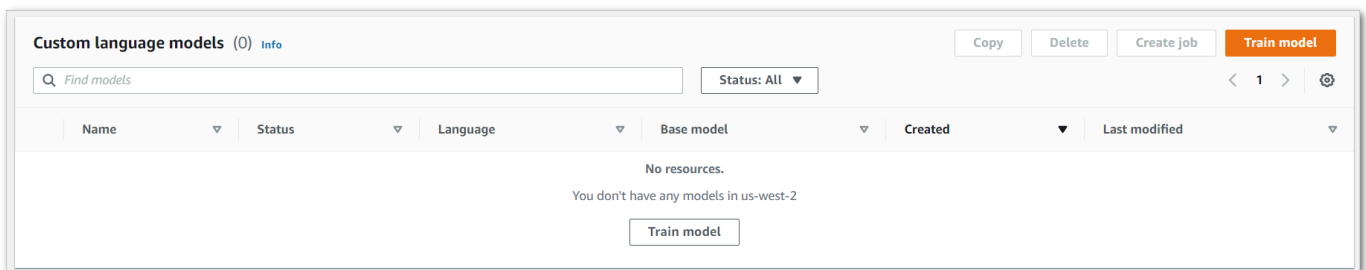
- **NarrowBand**: 샘플링 레이트가 16,000Hz 미만인 오디오에 이 옵션을 사용하십시오. 이 모델 유형은 일반적으로 8,000Hz로 녹음된 전화 통화에 사용됩니다.

- **WideBand**: 샘플링 레이트가 16,000Hz 이상인 오디오에 이 옵션을 사용하십시오.

AWS Management Console AWS CLI, 또는 AWS SDK를 사용하여 사용자 지정 언어 모델을 만들 수 있습니다. 다음 예를 참조하십시오.

## AWS Management Console

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창에서 [Customer Language Model] 을 선택합니다. 그러면 기존 사용자 지정 언어 모델을 보거나 새 사용자 지정 언어 모델을 학습할 수 있는 사용자 지정 언어 모델 페이지가 열립니다.
3. 새 모델을 학습시키려면 모델 학습을 선택합니다.



그러면 Train 모델 페이지로 이동합니다. 이름을 추가하고 언어를 지정한 다음 모델에 사용할 기본 모델을 선택합니다. 그런 다음 훈련 경로를 추가하고 선택적으로 튜닝 데이터를 추가합니다. 데이터에 액세스할 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 Train model (훈련 모델) 을 선택합니다.

## AWS CLI

이 예제에서는 [create-language-model](#) 명령을 사용합니다. 자세한 내용은 [CreateLanguageModel](#) 및 [LanguageModel](#) 단원을 참조하세요.

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

다음은 [create-language-model](#) 명령을 사용하는 또 다른 예와 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 들어 있습니다.

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

## AWS SDK for Python (Boto3)

이 예제에서는 AWS SDK for Python (Boto3) 를 사용하여 [create\\_language\\_model](#) 메서드를 사용하여 CLM을 생성합니다. 자세한 내용은 [CreateLanguageModel](#) 및 [LanguageModel](#) 단원을 참조하세요.

기능별, 시나리오 및 크로스 서비스 예제를 포함하여 AWS SDK를 사용하는 추가 예제는 이 장을 참조하십시오. [SDK를 사용한 Amazon Transcribe의 코드 예제 AWS](#)

```

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

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

```

## 사용자 지정 언어 모델 업데이트

Amazon Transcribe 사용자 지정 언어 모델에 사용할 수 있는 기본 모델을 지속적으로 업데이트합니다. 이러한 업데이트를 활용하려면 6~12개월마다 새로운 사용자 지정 언어 모델을 교육하는 것이 좋습니다.

사용자 지정 언어 모델이 최신 기본 모델을 사용하는지 확인하려면 AWS CLI 또는 AWS SDK를 사용하여 [DescribeLanguageModel](#) 요청을 실행한 다음 응답에서 UpgradeAvailability 필드를 찾으십시오.

그렇다면 UpgradeAvailability 모델이 최신 버전의 기본 모델을 실행하고 있지 않은 것입니다. true 사용자 지정 언어 모델에서 최신 기본 모델을 사용하려면 새 사용자 지정 언어 모델을 만들어야 합니다. 사용자 지정 언어 모델은 업그레이드할 수 없습니다.

## 사용자 지정 언어 모델 사용

사용자 지정 언어 모델을 만든 후에는 이를 트랜스크립션 요청에 포함시킬 수 있습니다. 예를 보려면 다음 섹션을 참조하십시오.

요청에 포함하려는 모델의 언어는 미디어에 지정한 언어 코드와 일치해야 합니다. 언어가 일치하지 않으면 사용자 지정 언어 모델이 트랜스크립션에 적용되지 않으며 경고나 오류가 발생하지 않습니다.

## 일괄 트랜스크립션에 사용자 지정 언어 모델 사용

일괄 트랜스크립션과 함께 사용자 지정 언어 모델을 사용하려면 다음 예제를 참조하십시오.

### 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. Job 설정 패널의 모델 유형에서 사용자 지정 언어 모델 상자를 선택합니다.

**Job settings**

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

**Model type** [Info](#)  
 Choose the type of model to use for the transcription job.

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

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

**Language**  
 Choose the language of the input audio.

**Custom model selection**  
 Choose an existing model or [create a new one.](#) [↗](#)

▶ **Additional settings**

드롭다운 메뉴에서 입력 언어도 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, US (en-US)

English, AU (en-AU)

English, UK (en-GB)

Hindi, IN (hi-IN)

Spanish, US (es-US)

- 사용자 지정 모델 선택에서 드롭다운 메뉴에서 기존 사용자 지정 언어 모델을 선택하거나 새 언어 모델 만들기를 선택합니다.

입력 데이터 패널에 입력 파일의 Amazon S3 위치를 추가합니다.

- 추가 구성 옵션을 보려면 다음을 선택합니다.

작업 생성을 선택하여 트랜스크립션 작업을 실행합니다.

## AWS CLI

이 예제에서는 [start-transcription-job](#) 명령과 `ModelSettings` 매개 변수를 `VocabularyName` 하위 매개 변수와 함께 사용합니다. 자세한 내용은 [StartTranscriptionJob](#) 및 [ModelSettings](#) 단원을 참조하세요.

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

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

다음은 [start-transcription-job](#) 명령을 사용하는 또 다른 예와 해당 작업에 대한 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 다음 요청 본문이 포함되어 있습니다.

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

## AWS SDK for Python (Boto3)

이 예제에서는 AWS SDK for Python (Boto3) 를 사용하여 [start\\_transcription\\_job](#) 메서드의 `ModelSettings` 인수를 사용하는 사용자 지정 언어 모델을 포함합니다. 자세한 내용은 [StartTranscriptionJob](#) 및 [ModelSettings](#) 단원을 참조하세요.

기능별, 시나리오 및 서비스 간 예제를 포함하여 AWS SDK를 사용하는 추가 예제는 이 [SDK를 사용한 Amazon Transcribe의 코드 예제 AWS](#) 장을 참조하십시오.

```
from __future__ import print_function
import time
```

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

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

## 스트리밍 트랜스크립션에 사용자 지정 언어 모델 사용

스트리밍 트랜스크립션과 함께 사용자 지정 언어 모델을 사용하려면 다음 예제를 참조하십시오.

### AWS Management Console

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창에서 실시간 트랜스크립션을 선택합니다. 아래로 스크롤하여 사용자 지정으로 이동하고 이 필드가 최소화된 경우 확장하십시오.

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

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

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

**Transcription output** **Current language: English, US**

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

00:00 of 15:00 min audio stream

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

3. 사용자 지정 언어 모델을 켜고 드롭다운 메뉴에서 모델을 선택합니다.

▼ **Customizations**

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

**Partial results stabilization** [Info](#)  
Configure Amazon Transcribe to present results that don't change as it processes the transcription output from your stream.

**Custom language model** [Info](#)  
Select the model you want to use with this streaming session.

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

Choose model ▼

스트림에 적용하려는 기타 설정을 포함하세요.

4. 이제 스트림을 텍스트로 변환할 준비가 되었습니다. 스트리밍 시작을 선택하고 말하기 시작하세요. 받아쓰기를 종료하려면 스트리밍 중지를 선택합니다.

## HTTP/2 스트림

이 예제는 사용자 지정 언어 모델을 포함하는 HTTP/2 요청을 만듭니다. 에서 HTTP/2 스트리밍을 사용하는 방법에 대한 자세한 내용은 [Amazon Transcribe 참조하십시오](#) [HTTP/2 스트림 설정](#). 관련 매개변수 및 헤더에 대한 자세한 내용은 [Amazon Transcribe](#) 을 참조하십시오 [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
```

매개변수 정의는 [API 참조에서](#) 찾을 수 있습니다. 모든 AWS API 작업에 공통적인 [매개변수는 일반 매개변수](#) 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 스트리밍을 사용하는 방법에 대한 자세한 내용은 [Amazon Transcribe](#) 을 참조하십시오 [WebSocket 스트림 설정](#). 매개변수에 대한 자세한 내용은 [Amazon Transcribe](#) 을 참조하십시오 [StartStreamTranscription](#).

```
GET wss://transcribestreaming.us-west-2.amazonaws.com:8443/stream-transcription-websocket?
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIAIOSFODNN7EXAMPLE%2F20220208%2Fus-west-2%2Ftranscribe%2Faws4_request
&X-Amz-Date=20220208T235959Z
&X-Amz-Expires=300
&X-Amz-Security-Token=security-token
&X-Amz-Signature=string
&X-Amz-SignedHeaders=content-type%3Bhost%3Bx-amz-date
```

```
&language-code=en-US  
&media-encoding=flac  
&sample-rate=16000  
&language-model-name=my-first-language-model
```

매개변수 [정의는 API 참조에서](#) 찾을 수 있습니다. 모든 AWS API 작업에 공통적인 [매개변수는 일반 매개변수](#) 섹션에 나열되어 있습니다.

## 사용자 지정 어휘 필터를 사용하여 단어를 삭제, 마스킹 또는 플래그 지정

사용자 지정 어휘 필터는 트랜스크립션 출력물에서 수정하려는 개별 단어의 사용자 지정 목록이 포함된 텍스트 파일입니다.

일반적인 사용 사례는 모욕적이거나 욕설적인 용어를 제거하는 것입니다. 그러나 사용자 지정 어휘 필터는 완전히 사용자 지정되므로 원하는 단어를 선택할 수 있습니다. 예를 들어 출시를 앞둔 신제품이 있는 경우 회의 녹취록에서 제품 이름을 숨길 수 있습니다. 이 경우 출시 전까지 제품 이름을 비밀로 up-to-date 유지하면서 이해관계자를 보호해야 합니다.

어휘 필터링에는 maskremove, 및 세 가지 표시 방법이 tag 있습니다. 각각의 작동 방식을 보려면 다음 예를 참조하십시오.

- **마스킹:** 지정된 단어를 세 개의 별표 (\*\*\*) 로 바꿉니다.

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

- **제거:** 지정된 단어를 삭제하고 그 자리에 아무 것도 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. 태깅을 통해 빠르게 트랜스크립트를 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 적용할 사용자 지정 어휘 필터와 필터링 방법을 지정할 수 있습니다. Amazon Transcribe 그런 다음 지정한 필터링 방법에 따라 트랜스크립트에 나타나는 정확한 단어 일치 항목을 수정합니다.

사용자 지정 어휘 필터를 일괄 처리 및 스트리밍 트랜스크립션 요청에 적용할 수 있습니다. 사용자 지정 어휘 필터를 생성하는 방법은 단원을 참조하십시오 [어휘 필터 만들기](#). 사용자 지정 어휘 필터를 적용하는 방법을 알아보려면 을 참조하십시오 [사용자 지정 단어 필터 사용](#).

### Note

Amazon Transcribe 인종적으로 민감한 용어는 자동으로 마스킹되지만 [AWS 기술 Support](#) 문의 하여 이 기본 필터를 사용하지 않도록 설정할 수 있습니다.

어휘 필터링에 대한 비디오 설명을 보려면 [어휘 필터 사용을](#) 참조하십시오.

### 어휘 필터링과 관련된 API 작업

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

## 어휘 필터 만들기

사용자 지정 어휘 필터를 만들기 위한 두 가지 옵션이 있습니다.

- 줄로 구분된 단어 목록을 UTF-8 인코딩을 사용하여 일반 텍스트 파일로 저장합니다.
  - 이 접근 방식은 AWS Management Console AWS CLI, 또는 AWS SDK와 함께 사용할 수 있습니다.
  - 를 AWS Management Console 사용하는 경우 사용자 지정 어휘 파일의 로컬 경로 또는 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.
- 쉼표로 구분된 단어 목록을 API 요청에 직접 포함하세요.
  - [Words](#) 파라미터를 사용하는 AWS CLI 또는 AWS SDK와 함께 이 접근 방식을 사용할 수 있습니다.

각 방법의 예는 을 참조하십시오. [사용자 지정 어휘 필터 만들기](#)

사용자 지정 어휘 필터를 만들 때 참고할 사항:



- 단어는 대/소문자를 구분하지 않습니다. 예를 들어, “저주”와 “CURSE”는 동일하게 취급됩니다.
- 정확히 일치하는 단어만 필터링됩니다. 예를 들어 필터에는 '옥설'이 포함되어 있지만 미디어에 '옥 설' 또는 '옥설'이라는 단어가 포함된 경우 이러한 단어는 필터링되지 않습니다. “옥설” 인스턴스만 필터링됩니다. 따라서 필터링하려는 단어의 모든 변형을 포함해야 합니다.
- 필터는 다른 단어에 포함된 단어에는 적용되지 않습니다. 예를 들어 사용자 지정 어휘 필터에 “해양”은 포함되지만 “잠수함”은 포함되어 있지 않은 경우 스크립트에서 “잠수함”은 변경되지 않습니다.
- 각 항목은 한 단어만 포함할 수 있습니다 (공백 없음).
- 사용자 지정 어휘 필터를 텍스트 파일로 저장하는 경우 UTF-8 인코딩이 적용된 일반 텍스트 형식이어야 합니다.
- 사용자 지정 어휘 필터는 한 개당 최대 100개까지 사용할 수 AWS 계정 있으며 각 필터의 크기는 최대 50Kb입니다.
- 해당 언어에 지원되는 문자만 사용할 수 있습니다. 자세한 내용은 해당 언어의 [문자 세트](#)를 참조하십시오.

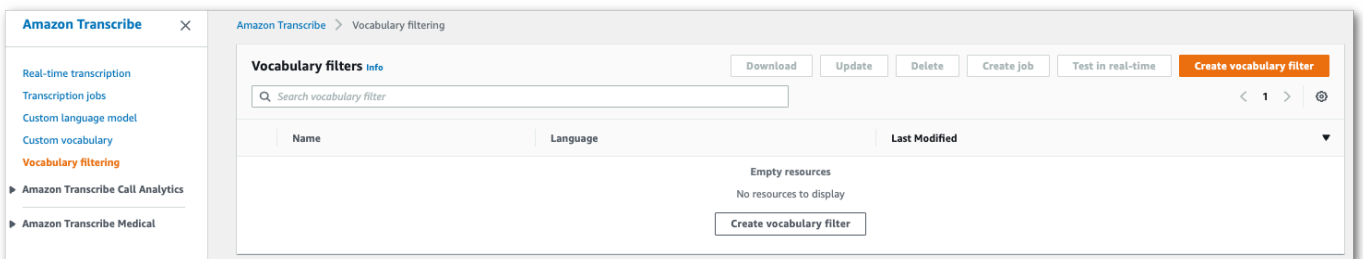
## 사용자 지정 어휘 필터 만들기

에서 사용할 사용자 지정 어휘 필터를 처리하려면 다음 예를 참조하십시오. Amazon Transcribe

### AWS Management Console

계속하기 전에 사용자 지정 어휘 필터를 텍스트 (\*.txt) 파일로 저장하세요. 선택적으로 파일을 Amazon S3 버킷에 업로드할 수 있습니다.

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창에서 어휘 필터링을 선택합니다. 그러면 기존 사용자 지정 어휘 필터를 보거나 새 사용자 지정 어휘 필터를 만들 수 있는 어휘 필터 페이지가 열립니다.
3. 어휘 필터 만들기를 선택합니다.



그러면 어휘 필터 만들기 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.

### Create vocabulary filter Info

#### Vocabulary filtering settings

**Name**

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

**Language**

English, US (en-US) ▼

**Vocabulary input source Info**

File upload

S3 location

**Vocabulary filter file location on S3**

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

Browse S3

File format: txt, maximum size 50 KB.

#### Tags - optional

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

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel
Create vocabulary filter

- 원하는 경우 사용자 지정 어휘 필터에 태그를 추가할 수 있습니다. 모든 필드를 작성했으면 페이지 하단에서 어휘 필터 만들기를 선택합니다. 파일을 처리하는 중 오류가 없으면 어휘 필터 페이지로 돌아갑니다.

이제 사용자 지정 어휘 필터를 사용할 준비를 마쳤습니다.

## AWS CLI

이 예제에서는 [create-vocabulary-filter](#) 명령을 사용하여 단어 목록을 사용 가능한 사용자 지정 어휘 필터로 처리합니다. 자세한 정보는 [CreateVocabularyFilter](#)을 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 포함할 수 있습니다.

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

다음은 [create-vocabulary-filter](#) 명령을 사용하는 또 다른 예와 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 다음 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 포함할 수 있습니다.

```
{
  "VocabularyFilterName": "my-first-vocabulary-filter",
```

```

"LanguageCode": "en-US",
"VocabularyFilterFileUri": "s3://DOC-EXAMPLE-BUCKET/my-vocabulary-filters/my-
vocabulary-filter.txt"
}

```

### Note

요청에 포함하는 VocabularyFilterFileUri 경우 사용할 Words 수 없으며 둘 중 하나를 선택해야 합니다.

## AWS SDK for Python (Boto3)

이 예제에서는 AWS SDK for Python (Boto3) 를 사용하여 [create\\_vocabulary\\_ary\\_filter](#) 메서드를 사용하여 사용자 지정 어휘 필터를 만듭니다. 자세한 정보는 [CreateVocabularyFilter](#) 을 참조하세요.

기능별, 시나리오 및 서비스 간 예제를 포함하여 AWS SDK를 사용하는 추가 예제는 이 [SDK를 사용한 Amazon Transcribe의 코드 예제 AWS](#) 장을 참조하십시오.

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 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,
    VocabularyFilterFileUri = 's3://DOC-EXAMPLE-BUCKET/my-vocabulary-filters/my-
vocabulary-filter.txt'
)
```

### Note

요청에 포함하는 VocabularyFilterFileUri 경우 사용할 Words 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 권한이 있는지 확인하세요. 역할에 올바른 권한이 없는 경우 요청이 실패합니다. DataAccessRoleArn 파라미터를 포함하여 요청 내에서 IAM 역할을 선택적으로 지정할 수 있습니다. 의 IAM 역할 및 정책에 대한 자세한 내용은 Amazon Transcribe 을 참조하십시오 [Amazon Transcribe 자격 증명 기반 정책 예제](#).

## 사용자 지정 단어 필터 사용

사용자 지정 어휘 필터가 생성되면 이를 트랜스크립션 요청에 포함시킬 수 있습니다. 예를 보려면 다음 섹션을 참조하십시오.

요청에 포함하는 사용자 지정 어휘 필터의 언어는 미디어에 지정한 언어 코드와 일치해야 합니다. 언어 식별을 사용하고 여러 언어 옵션을 지정하는 경우 지정된 언어당 하나의 사용자 지정 어휘 필터를 포함할 수 있습니다. 사용자 지정 어휘 필터의 언어가 오디오에서 식별된 언어와 일치하지 않는 경우 필터가 트랜스크립션에 적용되지 않으며 경고나 오류가 발생하지 않습니다.

## 일괄 트랜스크립션에 사용자 지정 어휘 필터 사용

일괄 트랜스크립션과 함께 사용자 지정 어휘 필터를 사용하려면 다음 예제를 참조하십시오.

## 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 포함시킨 후 다음을 선택합니다.

3. 작업 구성 페이지의 콘텐츠 제거 패널에서 어휘 필터링을 켜십시오.

## 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](#).

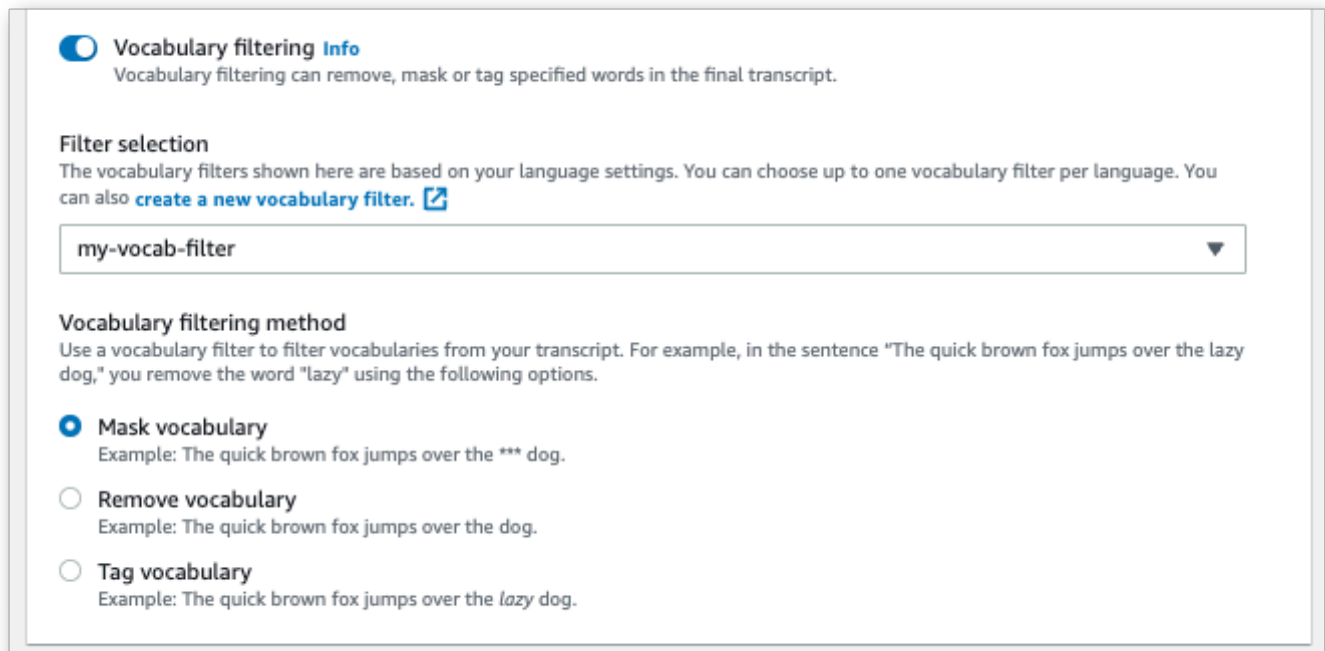
Choose a vocabulary filter ▼

### Customization

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

Cancel Previous Create job

- 드롭다운 메뉴에서 사용자 지정 어휘 필터를 선택하고 필터링 방법을 지정합니다.



5. 작업 생성을 선택하여 트랜스크립션 작업을 실행합니다.

## AWS CLI

이 예제에서는 [start-transcription-job](#) 명령 및 Settings 매개 변수를 VocabularyFilterName 및 VocabularyFilterMethod 하위 매개 변수와 함께 사용합니다. 자세한 내용은 [StartTranscriptionJob](#) 및 [Settings](#) 단원을 참조하세요.

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

다음은 [start-transcription-job](#) 명령을 사용하는 또 다른 예와 해당 작업에 대한 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 다음 요청 본문이 포함되어 있습니다.

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

### AWS SDK for Python (Boto3)

이 예제에서는 AWS SDK for Python (Boto3) 를 사용하여 [start\\_transcription\\_job](#) 메서드의 Settings 인수를 사용하는 사용자 지정 어휘 필터를 포함합니다. 자세한 내용은 [StartTranscriptionJob](#) 및 [Settings](#) 단원을 참조하세요.

기능별, 시나리오 및 서비스 간 예제를 포함하여 AWS SDK를 사용하는 추가 예제는 이 [SDK를 사용한 Amazon Transcribe의 코드 예제 AWS](#) 장을 참조하십시오.

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

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

## 스트리밍 트랜스크립션에서 사용자 지정 어휘 필터 사용

스트리밍 트랜스크립션과 함께 사용자 지정 어휘 필터를 사용하려면 다음 예제를 참조하십시오.

### AWS Management Console

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창에서 실시간 트랜스크립션을 선택합니다. 아래로 스크롤하여 콘텐츠 제거 설정으로 이동한 다음 이 필드가 최소화된 경우 확장하십시오.

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

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

**Transcription**

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

Transcription output Current language: English, US

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

00:00 of 15:00 min audio stream

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

- 어휘 필터링을 켜세요. 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 설정을 추가합니다.

4. 이제 스트림을 변환할 준비가 되었습니다. 스트리밍 시작을 선택하고 말하기 시작하세요. 받아쓰기를 종료하려면 스트리밍 중지를 선택합니다.

## HTTP/2 스트림

이 예제에서는 사용자 지정 어휘 필터 및 필터 메서드가 포함된 HTTP/2 요청을 생성합니다. 에서 HTTP/2 스트리밍을 사용하는 방법에 대한 자세한 내용은 [Amazon Transcribe 참조하십시오](#) [HTTP/2 스트림 설정](#). 관련 매개변수와 헤더에 대한 자세한 내용은 [Amazon Transcribe](#) 을 참조하십시오 [StartStreamTranscription](#).

```
POST /stream-transcription HTTP/2
host: transcribestreaming.us-west-2.amazonaws.com
X-Amz-Target: com.amazonaws.transcribe.Transcribe.StartStreamTranscription
Content-Type: application/vnd.amazon.eventstream
X-Amz-Content-Sha256: string
X-Amz-Date: 20220208T235959Z
Authorization: AWS4-HMAC-SHA256 Credential=access-key/20220208/us-west-2/transcribe/aws4_request, SignedHeaders=content-type;host;x-amz-content-sha256;x-amz-date;x-amz-target;x-amz-security-token, Signature=string
x-amzn-transcribe-language-code: en-US
x-amzn-transcribe-media-encoding: flac
x-amzn-transcribe-sample-rate: 16000
x-amzn-transcribe-vocabulary-filter-name: my-first-vocabulary-filter
x-amzn-transcribe-vocabulary-filter-method: mask
transfer-encoding: chunked
```

파라미터 [정의는 API 레퍼런스에서](#) 찾을 수 있으며, 모든 AWS API 작업에 공통적인 [파라미터는 일반 파라미터](#) 섹션에 나열되어 있습니다.

## WebSocket 스트림

이 예시에서는 사용자 지정 어휘 필터를 WebSocket 스트림에 적용하는 미리 서명된 URL을 만듭니다. 가독성을 높이기 위해 줄바꿈이 추가되었습니다. 에서 WebSocket 스트림을 사용하는 방법에 대한 자세한 내용은 [Amazon Transcribe](#) 을 참조하십시오 [WebSocket 스트림 설정](#). 매개변수에 대한 자세한 내용은 [Amazon Transcribe](#) 을 참조하십시오 [StartStreamTranscription](#).

```
GET wss://transcribestreaming.us-west-2.amazonaws.com:8443/stream-transcription-websocket?
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIAIOSFODNN7EXAMPLE%2F20220208%2Fus-west-2%2Ftranscribe%2Faws4_request
```

```
&X-Amz-Date=20220208T235959Z
&X-Amz-Expires=300
&X-Amz-Security-Token=security-token
&X-Amz-Signature=string
&X-Amz-SignedHeaders=content-type%3Bhost%3Bx-amz-date
&language-code=en-US
&media-encoding=flac
&sample-rate=16000
&vocabulary-filter-name=my-first-vocabulary-filter
&vocabulary-filter-method=mask
```

파라미터 [정의는 API 레퍼런스에서](#) 찾을 수 있으며, 모든 AWS API 작업에 공통적인 [파라미터는 일반 파라미터](#) 섹션에 나열되어 있습니다.

## 독성 음성 감지

독성 음성 감지는 관련 소셜 미디어 플랫폼을 조정하는 데 도움이 되도록 설계되었습니다. peer-to-peer 온라인 게임 및 소셜 채팅 플랫폼과 같은 대화 유해한 언어의 사용은 개인, 또래 집단 및 지역 사회에 심각한 해를 끼칠 수 있습니다. 유해 언어를 신고하면 조직에서 예의 바른 대화를 유지하고 사용자가 자유롭게 만들고 공유하고 참여할 수 있는 안전하고 포용적인 온라인 환경을 유지할 수 있습니다.

Amazon Transcribe 독성 탐지는 음성 및 텍스트 기반 신호를 모두 활용하여 성희롱, 증오심 발언, 위협, 학대, 욕설, 모욕, 그래픽 등 7가지 범주에 걸쳐 음성 기반 유해 콘텐츠를 식별하고 분류합니다. 텍스트 외에도 Amazon Transcribe 독성 감지는 어조 및 음높이와 같은 음성 신호를 사용하여 음성의 독성 의도를 파악합니다. 이는 의도를 고려하지 않고 특정 용어에만 초점을 맞추도록 설계된 표준 콘텐츠 조정 시스템을 개선한 것입니다.

Amazon Transcribe 유해한 음성에 플래그를 지정하고 분류하여 수동으로 처리해야 하는 데이터의 양을 최소화합니다. 이를 통해 콘텐츠 중재자는 플랫폼에서 담론을 빠르고 효율적으로 관리할 수 있습니다.

독성 음성 범주에는 다음이 포함됩니다.

- 욕설: 무례하거나 저속하거나 모욕적인 단어, 문구 또는 두문자어가 포함된 말.
- 헤이트 스피치: 정체성 (예: 인종, 민족, 성별, 종교, 성적 취향, 능력, 출신 국가) 을 근거로 개인이나 집단을 비판, 모욕, 비인간적으로 표현하는 발언.
- 성적: 신체 부위, 신체적 특징 또는 성별을 직간접적으로 언급하여 성적 관심, 활동 또는 흥분을 나타내는 말.
- 모욕: 비하, 모욕, 조롱, 모욕 또는 얕보는 언어를 포함하는 말. 이러한 유형의 언어는 괴롭힘으로도 분류됩니다.
- 폭력 또는 위협: 개인이나 집단에 고통, 상해 또는 적대감을 주려는 위협을 포함하는 말.
- 그래픽: 시각적으로 묘사적이고 불쾌할 정도로 생생한 이미지를 사용하는 연설. 이러한 유형의 언어는 종종 수신자의 불편함을 증폭시키기 위해 의도적으로 장황하게 사용됩니다.
- 괴롭힘 또는 학대: 비하 및 객관적인 용어를 포함하여 받는 사람의 심리적 안녕에 영향을 미치기 위한 발언. 이러한 유형의 언어는 괴롭힘으로도 분류됩니다.

독성 감지는 음성 세그먼트 (자연스러운 일시 중지 사이의 음성) 를 분석하고 이러한 세그먼트에 신뢰도 점수를 할당합니다. 신뢰도 점수는 0에서 1 사이의 값입니다. 신뢰도 점수가 클수록 관련 카테고리에서 해당 콘텐츠가 유해한 언어일 가능성이 더 높다는 것을 나타냅니다. 이러한 신뢰도 점수를 사용하여 사용 사례에 적합한 독성 검출 임계값을 설정할 수 있습니다.

**Note**

독성 검출은 미국 영어로 된 배치 전사에만 사용할 수 있습니다.(en-US).

보기예제 [출력](#) JSON 형식으로.

## 독성 음성 감지 사용

### 일괄 트랜스크립션에 독성 음성 감지 기능 사용

일괄 트랜스크립션과 함께 독성 음성 감지를 사용하려면 다음의 예를 참조하십시오.

#### AWS Management Console

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창에서 선택트랜스크립션 작업, 그런 다음 선택작업 생성(오른쪽 상단). 이렇게 하면 열립니다. 직무 세부 정보 지정페이지.

## 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. 에릭무 세부 정보 지정페이지에서 원하는 경우 PII 교정을 활성화할 수도 있습니다. 목록에 있는 다른 옵션은 독성 검출과 함께 지원되지 않습니다. 다음(Next)을 선택합니다. 이렇게 하면 다음 페이지로 이동합니다.작업 구성 - 선택 사항페이지. 에서오디오 설정패널, 선택독성 감지.



## Audio settings

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

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

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

## Content removal

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

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

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

## Customization

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

Cancel

Previous

Create job

4. 선택작업 생성트랜스크립션 작업을 실행하려면
5. 트랜스크립션 작업이 완료되면 다음 사이트에서 트랜스크립트를 다운로드할 수 있습니다. 다운로드 트랜스크립션 작업의 세부 정보 페이지에 있는 드롭다운 메뉴.

## AWS CLI

이 예제에서는 [start-transcription-job](#) 명령 및 ToxicityDetection 매개 변수. 자세한 내용은 [StartTranscriptionJob](#) 및 [ToxicityDetection](#) 단원을 참조하세요.

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

다음은 를 사용하는 또 다른 예입니다. [start-transcription-job](#) 명령 및 독성 탐지를 포함하는 요청 본문.

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

파일 `my-first-toxicity-job.json` 다음 요청 본문을 포함합니다.

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

## AWS SDK for Python (Boto3)

이 예제에서는 AWS SDK for Python (Boto3) 활성화하려면 [ToxicityDetection...](#)에 대한 [시작\\_트랜스크립션\\_작업](#) 방법. 자세한 내용은 [StartTranscriptionJob](#) 및 [ToxicityDetection](#) 단원을 참조하세요.

를 사용한 추가 예제는 AWS 기능별, 시나리오 및 크로스 서비스 예제를 포함한 SDK는 다음을 참조하십시오. [SDK를 사용한 Amazon Transcribe의 코드 예제 AWS](#) 장.

```

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

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

```

## 출력 예

유해 언어는 트랜스크립션 출력물에 태그가 지정되고 분류됩니다. 독성 발언의 각 사례는 분류되고 신뢰도 점수 (0~1 사이의 값) 가 할당됩니다. 신뢰도 값이 클수록 해당 콘텐츠가 지정된 범주 내에서 유해 언어일 가능성이 더 크다는 것을 나타냅니다.

### 예제 출력 (JSON)

다음은 분류된 유해 언어를 관련 신뢰도 점수와 함께 보여주는 JSON 형식의 예제 출력입니다.

```
{
  "jobName": "my-toxicity-job",
  "accountId": "111122223333",
  "results": {
    "transcripts": [...],
    "items": [...],
    "toxicity_detection": [
      {
        "text": "What the * are you doing man? That's why I didn't want to play
with your * . man it was a no, no I'm not calming down * man. I well I spent I spent
too much * money on this game.",
        "toxicity": 0.7638,
        "categories": {
          "profanity": 0.9913,
          "hate_speech": 0.0382,
          "sexual": 0.0016,
          "insult": 0.6572,
          "violence_or_threat": 0.0024,
          "graphic": 0.0013,
          "harassment_or_abuse": 0.0249
        },
        "start_time": 8.92,
        "end_time": 21.45
      },
      {
        "text": "What? Who? What the * did you just say to me? What's your
address? What is your * address? I will pull up right now on your * * man. Take your *
back to , tired of this **.",
        "toxicity": 0.9816,
        "categories": {
          "profanity": 0.9865,
```

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

## 식별 정보

교정은 성적 증명서에서 개인 식별 정보 (PII) 형태로 민감한 콘텐츠를 숨기거나 제거하는 데 사용됩니다. 수정할 Amazon Transcribe 수 있는 PII 유형은 일괄 처리와 스트리밍 트랜스크립션에 따라 다릅니다. 각 전사 방법에 대한 PII 목록을 보려면 [배치 작업의 PII 편집 및](#) 를 참조하십시오. [실시간 스트림에서 PII 편집 또는 식별](#). 스트리밍 트랜스크립션을 사용하면 편집하지 않고 PII에 플래그를 지정할 수도 있습니다. 출력 예는 [을 PII 식별 출력 예시](#) 참조하십시오.

교정이 활성화되면 수정된 사본만 생성하거나 수정된 사본과 수정되지 않은 대본을 모두 생성할 수 있습니다. 편집된 트랜스크립트만 생성하기로 선택한 경우 전체 대화가 저장되는 곳은 미디어뿐이라는 점에 유의하세요. 원본 미디어를 삭제하면 수정되지 않은 PII에 대한 기록이 남지 않습니다. 따라서 편집된 성적표와 함께 편집되지 않은 성적표를 생성하는 것이 현명할 수 있습니다.

일괄 트랜스크립션을 사용한 PII 편집에 대한 자세한 내용은 다음을 참조하십시오. [배치 작업의 PII 편집](#).

스트리밍 트랜스크립션을 통한 PII 수정 또는 식별에 대한 자세한 내용은 다음을 참조하십시오. [실시간 스트림에서 PII 편집 또는 식별](#).

### Important

수정 기능은 민감한 데이터를 식별하고 제거하도록 설계되었습니다. 그러나 머신 러닝의 예측 특성으로 인해, 는 트랜스크립트에서 민감한 데이터의 모든 인스턴스를 식별하여 제거하지 Amazon Transcribe 못할 수 있습니다. 편집된 출력을 검토하여 요구 사항을 충족하는지 확인하는 것이 좋습니다.

수정 기능은 HIPAA (Health Insurance Portability and Accountability Act of 1996), 같은 HIPAA (Health Insurance Portability and Accountability Act of 1996), 같은

수정 기능에 대한 동영상 설명을 보려면 [콘텐츠 수정을 통한 PII 식별 및 수정을](#) 참조하십시오. Amazon Transcribe

## 배치 작업의 PII 편집

일괄 트랜스크립션 작업 중에 트랜스크립트에서 개인 식별 정보 (PII) 를 삭제하는 경우 식별된 각 PII 인스턴스를 트랜스크립트의 본문에 있는 [PII] 것으로 Amazon Transcribe 대체합니다. 트랜스크립

선 출력 부분에서 수정된 PII의 유형도 볼 수 있습니다. word-for-word 출력 샘플은 [편집된 출력 예제 \(일괄 처리\)](#)를 참조하세요.

일괄 트랜스크립션을 통한 교정은 미국 영어 (en-US) 및 미국 스페인어 () 로 제공됩니다. es-US 수정은 [언어 식별](#)과 호환되지 않습니다.

편집된 성적표와 수정되지 않은 대본은 모두 동일한 출력 버킷에 저장됩니다. Amazon S3 Amazon Transcribe 지정한 버킷 또는 서비스에서 관리하는 기본 Amazon S3 버킷에 저장합니다.

일괄 트랜스크립션을 위해 Amazon Transcribe 인식할 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 길 이이지만 Amazon Transcribe 는 마지막 4자리 만 있으면 은행 계좌 번호도 인식합니다.
BANK_ROUTING	미국 은행 계좌 라우팅 번호. 일반적으로 9자리 길 이이지만 Amazon Transcribe 는 마지막 4자리 만 있으면 라우팅 번호도 인식합니다.
CREDIT_DEBIT_CVV	VISA와 Discover 신용카드 및 직불카드에 있는 3자리 카드 확인 코드 (CVV). MasterCard 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, 01/21, 2021년 1월 등의 만료일을 Amazon Transcribe 인식할 수 있습니다.

PII 유형	설명
CREDIT_DEBIT_NUMBER	신용카드 또는 직불카드 번호. 이 번호의 길이는 13~16자리까지 다양하지만 마지막 4자리만 있는 경우 신용카드 또는 직불카드 Amazon Transcribe 번호도 인식합니다.
EMAIL	이메일 주소(예: efua.owusu@email.com).
NAME	개인의 이름. 이 개체 유형에는 Mr., Mrs., Miss 또는 Dr.와 같은 직함은 포함되지 Amazon Transcribe 않습니다. 또는 Dr.는 조직 또는 주소의 일부인 이름에는 이 개체 유형을 적용하지 않습니다. 예를 들어, John Doe 조직을 조직으로 Amazon Transcribe 인식하고 Jane Doe Street 를 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 시민권자, 영주권자, 임시 근로 거주자에게 발급되는 9자리 숫자입니다. Amazon Transcribe 또한 마지막 4자리만 있는 경우 사회보장번호를 인식합니다.

AWS Management Console AWS CLI, 또는 SDK를 사용하여 일괄 트랜스크립션 작업을 시작할 수 있습니다. AWS

## 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)**

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

**Personal (5 of 5 selected)**

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

---

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

5. 작업 생성을 선택하여 트랜스크립션 작업을 실행합니다.

## AWS CLI

이 예제에서는 명령과 매개변수를 사용합니다. [start-transcription-job](#)content-redaction 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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac \
--output-bucket-name DOC-EXAMPLE-BUCKET \
--output-key my-output-files/ \
--language-code en-US \
```

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

다음은 [start-transcription-job](#) 메서드를 사용하는 또 다른 예시입니다. 요청 본문은 해당 작업에 대한 PII를 수정합니다.

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

## AWS SDK for Python (Boto3)

[이 예제에서는 AWS SDK for Python \(Boto3\) 를 사용하여 start\\_transcription\\_job 메서드의 ContentRedaction 인수를 사용하여 내용을 수정합니다.](#) 자세한 내용은 [StartTranscriptionJob](#) 및 [ContentRedaction](#) 섹션을 참조하세요.

기능별, 시나리오 및 크로스 서비스 예제를 포함하여 AWS SDK를 사용하는 추가 예제는 이 장을 참조하십시오. [SDK를 사용한 Amazon Transcribe의 코드 예제 AWS](#)

```
from __future__ import print_function
```

```

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

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

```

### Note

일괄 작업에 대한 PII 수정은 아시아 태평양 (홍콩), 아시아 태평양 (뭄바이), 아시아 태평양 (서울), 아시아 태평양 (싱가포르), 아시아 태평양 (시드니), 아시아 태평양 (도쿄), (미국 서부), 캐나다 GovCloud (중부), EU (프랑크푸르트), EU (아일랜드), EU (런던), EU (파리), 중동 (바레인), 남아메리카 (사오) 에서만 지원됩니다. 파울로, 미국 동부 (버지니아 북부), 미국 동부 (오하이오), 미국 서부 (오레곤), 미국 서부 (캘리포니아 북부) AWS 리전

## 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 트랜스크립션 결과의 PII에 개체 아래에 Amazon Transcribe 레이블을 지정합니다. Entities 출력 샘플은 [편집된 스트리밍 출력 예시](#) 및 [PII 식별 출력 예시](#)를 참조하세요.

스트리밍 트랜스크립션을 통한 PII의 수정 및 식별은 호주 (en-AU), 영국 (), 미국 () 및 스페인 미국 방언 () 과 같은 영어 방언을 사용하여 사용할 수 있습니다. en-GB en-US es-US

스트리밍 작업에 대한 PII 식별 및 수정은 오디오 세그먼트의 완전한 트랜스크립션 시에만 수행됩니다.

스트리밍 트랜스크립션에서 인식할 수 있는 Amazon Transcribe PII의 유형

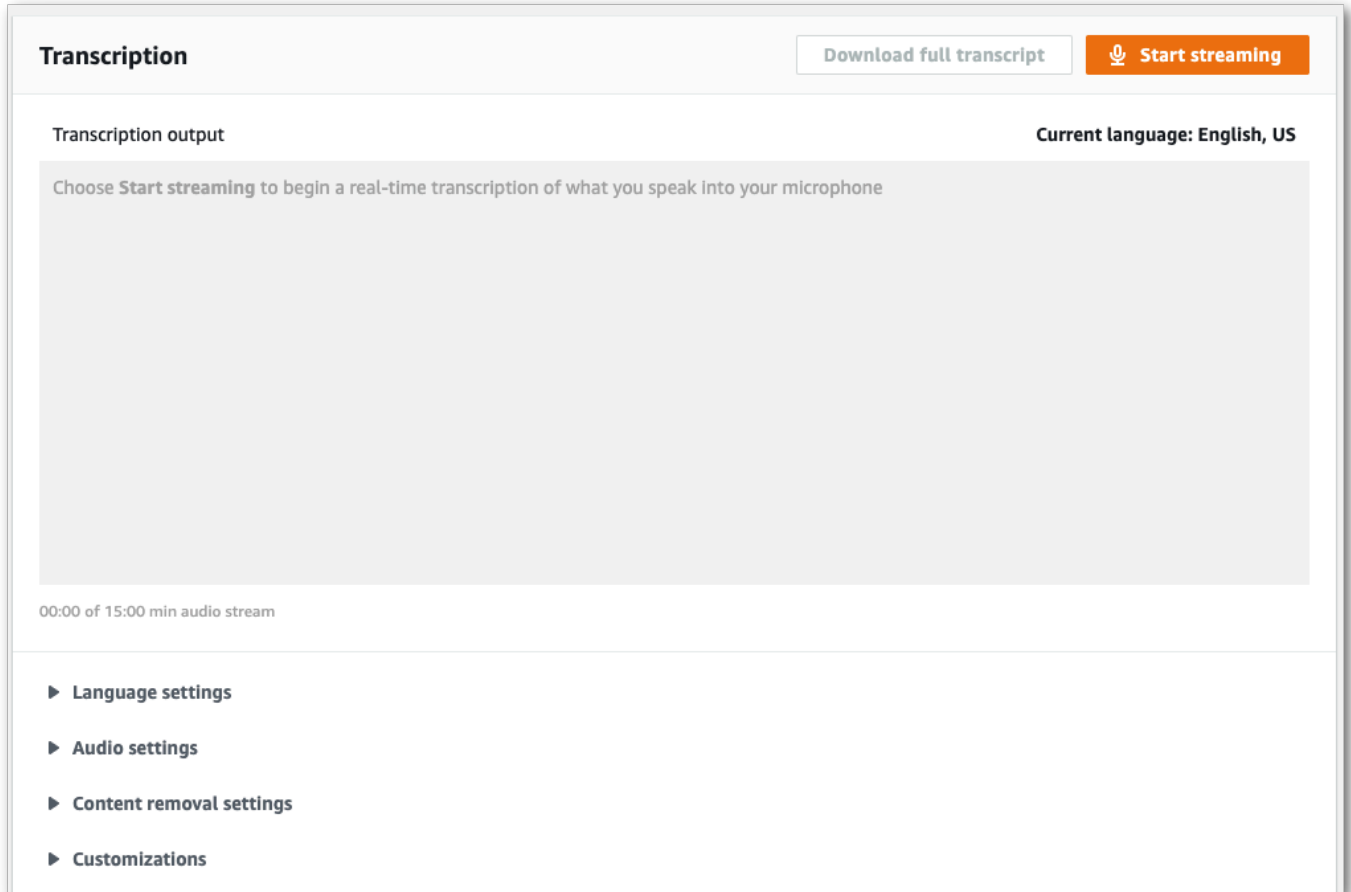
PII 유형	설명
ADDRESS	실제 주소(예: 100 Main Street, Anytown, USA 또는 Suite #12, Building 123). 주소에는 거리, 건물, 위치, 도시, 주, 국가, 카운티, 우편번호, 구역, 지역 등의 정보가 포함될 수 있습니다.
ALL	이 테이블에 나열된 모든 PII 유형을 편집하거나 식별합니다.
BANK_ACCOUNT_NUMBER	미국 은행 계좌 번호. 일반적으로 10~12자리 길 이이지만 Amazon Transcribe 는 마지막 4자리 만 있으면 은행 계좌 번호도 인식합니다.
BANK_ROUTING	미국 은행 계좌 라우팅 번호. 일반적으로 9자리 길 이이지만 Amazon Transcribe 는 마지막 4자리 만 있으면 라우팅 번호도 인식합니다.
CREDIT_DEBIT_CVV	VISA MasterCard, Discover 신용카드 및 직불카드에 있는 3자리 카드 확인 코드 (CVV). 아메리칸 익스프레스 신용카드나 직불카드의 경우 4자리 숫자 코드입니다.

PII 유형	설명
CREDIT_DEBIT_EXPIRY	신용카드 또는 직불카드 만료 날짜. 이 숫자는 보통 4자리 숫자이며, 월/년 또는 MM/YY 형식으로 지정됩니다. 예를 들어 01/21, 2021년 1월, 2021년 1월과 같은 만료일을 Amazon Transcribe 인식할 수 있습니다.
CREDIT_DEBIT_NUMBER	신용카드 또는 직불카드 번호. 이 번호의 길이는 13~16자리까지 다양하지만 마지막 4자리만 있는 경우 신용카드 또는 직불카드 Amazon Transcribe 번호도 인식합니다.
EMAIL	이메일 주소(예: efua.owusu@email.com).
NAME	개인의 이름. 이 개체 유형에는 Mr., Mrs., Miss 또는 Dr.와 같은 직함은 포함되지 않습니다. 또는 Dr.는 조직 또는 주소의 일부인 이름에는 이 개체 유형을 적용하지 않습니다. 예를 들어, John Doe 조직을 조직으로 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 시민권자, 영주권자, 임시 근로 거주자에게 발급되는 9자리 숫자입니다. Amazon Transcribe 또한 마지막 4자리만 있는 경우 사회보장번호를 인식합니다.

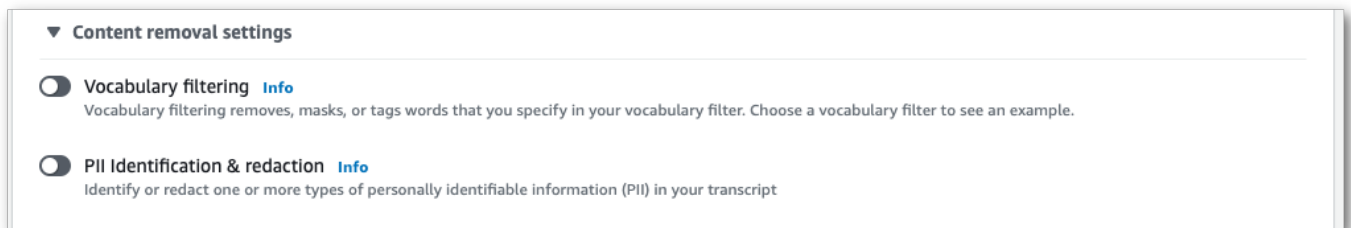
AWS Management Console WebSocket, 또는 HTTP/2를 사용하여 스트리밍 트랜스크립션을 시작할 수 있습니다.

## 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 켭니다.



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)**

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

**Personal (8 of 8 selected)**

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

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

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

**Other (1 of 1 selected)**

DATE\_TIME

**▶ Customizations**

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

## WebSocket 스트림

이 예제는 스트림에서 PII 수정 (또는 PII 식별) 을 사용하는 미리 서명된 URL을 만듭니다. WebSocket 가독성을 높이기 위해 줄바꿈이 추가되었습니다. 에서 WebSocket 스트림을 사용하는 방법에 대한 자세한 내용은 을 참조하십시오. Amazon Transcribe [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.

매개변수 정의는 [API 참조에서 찾을 수 있으며](#), 모든 AWS API 작업에 공통적인 [매개변수는 공통 매개변수](#) 섹션에 나열되어 있습니다.

## HTTP/2 스트림

이 예시에서는 PII 식별 또는 PII 수정이 활성화된 HTTP/2 요청을 생성합니다. HTTP/2 스트리밍을 사용하는 방법에 대한 자세한 내용은 [Amazon Transcribe 참조하십시오](#). [HTTP/2 스트림 설정](#) 관련 파라미터 및 헤더에 대한 자세한 내용은 [Amazon Transcribe StartStreamTranscription](#)

```
POST /stream-transcription HTTP/2
host: transcribestreaming.us-west-2.amazonaws.com
X-Amz-Target: com.amazonaws.transcribe.Transcribe.StartStreamTranscription
Content-Type: application/vnd.amazon.eventstream
X-Amz-Content-Sha256: string
X-Amz-Date: 20220208T235959Z
Authorization: AWS4-HMAC-SHA256 Credential=access-key/20220208/us-west-2/transcribe/aws4_request, SignedHeaders=content-type;host;x-amz-content-sha256;x-amz-date;x-amz-target;x-amz-security-token, Signature=string
x-amzn-transcribe-language-code: en-US
x-amzn-transcribe-media-encoding: flac
x-amzn-transcribe-sample-rate: 16000
x-amzn-transcribe-content-identification-type: PII (or x-amzn-transcribe-content-redaction-type: PII)
x-amzn-transcribe-pii-entity-types: NAME,ADDRESS
transfer-encoding: chunked
```

동일한 요청에서 content-identification-type 및 content-redaction-type 둘 다 사용할 수는 없습니다.

매개변수 정의는 [API 참조에서 찾을 수 있으며](#), 모든 AWS API 작업에 공통적인 매개변수는 [공통 매개변수](#) 섹션에 나열되어 있습니다.

### Note

스트리밍을 위한 PII 수정은 아시아 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. AWS 리전

## PII 수정 및 식별 출력의 예

다음 예에서는 배치 및 스트리밍 작업의 편집된 출력과 스트리밍 작업의 PII 식별을 보여 줍니다.

콘텐츠 수정을 사용하는 트랜스크립션 작업은 두 가지 유형의 confidence 값을 생성합니다. 자동 음성 인식(ASR) 신뢰도는 pronunciation 또는 punctuation type을 갖는 항목이 특정 발화임을 나타냅니다. 다음 트랜스크립트 출력에서 해당 Good 단어에는 가 confidence 표시됩니다 1.0. 이 신뢰도 값은 이 성적표의 단어가 '좋은'이라고 100% 확신한다는 것을 나타냅니다. Amazon Transcribe [PII] 태그의 confidence 값은 편집하도록 플래그가 지정된 음성이 실제로 PII라는 것의 신뢰도입니다. 다음 트랜스크립트 출력에서 Amazon Transcribe o는 트랜스크립트에서 수정한 엔티티가 PII라고 99.99% 확신한다는 것을 0.9999 나타냅니다. confidence

### 편집된 출력 예제 (일괄 처리)

```
{
  "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": [
    {
      "start_time": "2.86",
      "end_time": "3.35",
      "alternatives": [
        {
          "confidence": "1.0",
          "content": "Good"
        }
      ],
      "type": "pronunciation"
    },
    Items removed for brevity
    {
      "start_time": "5.56",
      "end_time": "6.25",
      "alternatives": [
        {
          "content": "[PII]",
          "redactions": [
            {
              "confidence": "0.9999",
              "type": "NAME",
              "category": "PII"
            }
          ]
        }
      ],
      "type": "pronunciation"
    },
    Items removed for brevity
  ],
},
"status": "COMPLETED"
}

```

비교를 위해 편집되지 않은 대본은 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": [
        {
          "start_time": "2.86",
          "end_time": "3.35",
          "alternatives": [
            {
              "confidence": "1.0",
              "content": "Good"
            }
          ],
          "type": "pronunciation"
        },
        Items removed for brevity
        {
          "start_time": "5.56",
          "end_time": "6.25",
          "alternatives": [
            {
              "confidence": "0.9999",
              "content": "Mike",
            }
          ],
          "type": "pronunciation"
        },
        Items removed for brevity
      ],
    },
    "status": "COMPLETED"
  }

```

## 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,
```



```
    },
    {
      "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 생성

Amazon Transcribe 비디오 자막으로 사용하기 위한 WebVTT (\*.vtt) 및 SubRip (\*.srt) 출력을 지원합니다. 일괄 비디오 트랜스크립션 작업을 설정할 때 파일 형식을 하나 또는 두 개 모두 선택할 수 있습니다. 자막 기능을 사용하면 선택한 자막 파일과 일반 자막 파일 (추가 정보 포함) 이 생성됩니다. 자막 및 트랜스크립션 파일이 동일한 대상으로 출력됩니다.

자막은 텍스트가 음성과 동시에 표시되며 자연스럽게 일시 중지되거나 화자가 말을 마칠 때까지 계속 표시됩니다. 트랜스크립션 요청에서 자막을 활성화했는데 오디오에 음성이 없으면 자막 파일이 생성되지 않습니다.

### Important

Amazon Transcribe는 자막 0 출력에 더 널리 사용되는 값과는 다른 기본 시작 인덱스를 사용합니다. 1 의 1 시작 인덱스가 필요한 경우 [OutputStartIndex](#) 파라미터를 사용하여 API AWS Management Console 요청이나 API 요청에서 이를 지정할 수 있습니다.

잘못된 시작 색인을 사용하면 다른 서비스와의 호환성 오류가 발생할 수 있으므로 자막을 만들기 전에 필요한 시작 색인이 무엇인지 확인해야 합니다. 어떤 값을 사용해야 할지 확실하지 않은 경우 선택하는 1 것이 좋습니다. 자세한 [Subtitles](#) 내용은 를 참조하십시오.

자막과 함께 지원되는 기능:

- 콘텐츠 수정 — 편집된 모든 콘텐츠는 자막 및 일반 자막 출력 파일 모두에 PII "로 반영됩니다. 오디오는 변경되지 않습니다.
- 어휘 필터 — 자막 파일은 트랜스크립션 파일에서 생성되므로 표준 트랜스크립션 출력에서 필터링한 모든 단어도 자막에서 필터링됩니다. 필터링된 콘텐츠는 공백으로 표시되거나 트랜스크립트 및 \*\*\* 자막 파일에 표시됩니다. 오디오는 변경되지 않습니다.
- 화자 분음 — 특정 자막 세그먼트에 화자가 여러 명인 경우 대시를 사용하여 각 화자를 구분합니다. 이는 WebVTT와 SubRip 형식 모두에 적용됩니다. 예를 들면 다음과 같습니다.
  - -- 사람이 말한 텍스트 1
  - -- 사람 2가 말한 텍스트

자막 파일은 Amazon S3 트랜스크립션 출력물과 동일한 위치에 저장됩니다.



[Amazon Transcribe 비디오 스넵: 코드를 작성하지 않고 비디오 자막 만들기를 참조하십시오.](#) 자막 만들기에 대한 비디오 안내를 참조하십시오.

## 자막 파일 생성

AWS Management Console, AWS CLI, 또는 AWSSDK를 사용하여 자막 파일을 만들 수 있습니다. 다음 예를 참조하십시오.

### AWS Management Console

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창에서 트랜스크립션 작업을 선택한 다음 작업 만들기 (오른쪽 상단) 를 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 0 이지만 더 널리 사용됩니다. 어떤 값을 사용해야 할지 확실하지 않은 경우 다른 서비스와의 호환성이 향상될 수 있으므로 선택하는 1 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

4. 작업 세부 정보 지정 페이지에 포함하려는 다른 필드를 모두 입력한 후 다음을 선택합니다. 그러면 작업 구성 - 옵션 페이지가 표시됩니다.
5. 트랜스크립션 작업을 실행하려면 작업 생성을 선택합니다.

## AWS CLI

이 예제에서는 [start-transcription-job](#) 명령과 Subtitles 매개 변수를 사용합니다. 자세한 내용은 [StartTranscriptionJob](#) 및 [Subtitles](#) 단원을 참조하세요.

```
aws transcribe start-transcription-job \  
--region us-west-2 \  
--transcription-job-name my-first-transcription-job \  
--media MediaFileUri=s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac \  
--output-bucket-name DOC-EXAMPLE-BUCKET \  
--output-key my-output-files/ \  
--language-code en-US \  
--subtitles Formats=vtt,srt,OutputStartIndex=1
```

다음은 [start-transcription-job](#) 명령을 사용하는 또 다른 예와 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 들어 있습니다.

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

## AWS SDK for Python (Boto3)

이 예제에서는 [start\\_transcription\\_job](#) 메서드의 Subtitles 인수를 사용하여 자막을 AWS SDK for Python (Boto3) 추가하는 데 사용합니다. 자세한 내용은 [StartTranscriptionJob](#) 및 [Subtitles](#) 단원을 참조하세요.

기능별, 시나리오 및 크로스 서비스 예제를 포함하여 AWS SDK를 사용하는 추가 예제는 이 장을 참조하십시오. [SDK를 사용한 Amazon Transcribe의 코드 예제 AWS](#)

```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
job_name = "my-first-transcription-job"
job_uri = "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
transcribe.start_transcription_job(
    TranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'DOC-EXAMPLE-BUCKET',
    OutputKey = 'my-output-files/',
    LanguageCode = 'en-US',
    Subtitles = {
        'Formats': [
            'vtt', 'srt'
        ],
        'OutputStartIndex': 1
    }
)

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

## Call Analytics를 사용한 콜센터 오디오 분석

Amazon Transcribe Call Analytics를 사용하여 고객-상담원 상호 작용에 대한 통찰력을 얻으십시오. Call Analytics는 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.

Call Analytics는 통화 후 트랜스크립션과 실시간 트랜스크립션에 사용할 수 있습니다. Amazon S3 버킷에 있는 파일을 트랜스크립션하는 경우 통화 후 트랜스크립션을 수행하게 됩니다. 오디오 스트림을 트랜스크립션하는 경우 실시간 트랜스크립션을 수행하는 것입니다. 이 두 가지 트랜스크립션 방법은 서로 다른 Call Analytics 인사이트와 기능을 제공합니다. 각 방법에 대한 자세한 내용은 [통화 후 분석](#) 및 [실시간 Call Analytics](#) 섹션을 참조하세요.

실시간 Call Analytics 트랜스크립션을 사용하면 요청에 [통화 후 분석](#)을 포함할 수도 있습니다. 통화 후 분석 기록은 요청에서 지정한 Amazon S3 버킷에 저장됩니다. 자세한 내용은 [실시간 트랜스크립션을 통한 통화 후 분석](#) 섹션을 참조하세요.

### Call Analytics와 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 Call Analytics 및 에이전트 지원](#)을 참조하세요.

Call Analytics에서 사용할 수 있는 기능을 Amazon Transcribe 및 Amazon Transcribe 의료용 기능과 비교하려면 [기능 표](#)를 참조하십시오.

시작하려면 [통화 후 분석 트랜스크립션 시작](#) 및 [실시간 Call Analytics 트랜스크립션 시작](#) 섹션을 참조하세요. Call Analytics 출력은 표준 트랜스크립션 작업의 출력과 비슷하지만 추가 분석 데이터를 포함합니다. 샘플 출력을 보려면 [통화 후 분석 출력](#) 및 [실시간 Call Analytics 출력](#)을 참조하세요.

## 고려 사항 및 추가 정보

Call Analytics를 사용하기 전에 다음 사항을 참고하세요.

- Call Analytics는 2채널 오디오만 지원합니다. 한 채널에는 에이전트가 있고 두 번째 채널에는 고객이 있습니다.
- [Job 대기열](#)은 통화 후 분석 작업에 항상 사용할 수 있으므로 동시 Call Analytics 작업은 100개로 제한됩니다. 할당량 증가를 요청하려면 [AWS 서비스 할당량](#) 섹션을 참조하세요.
- 통화 후 분석 작업의 입력 파일은 500MB를 초과할 수 없으며 4시간 미만이어야 합니다. WAV가 아닌 특정 압축 오디오 파일 형식의 경우 파일 크기 제한이 더 작을 수 있다는 점에 유의하십시오.
- 범주를 사용하는 경우 Call Analytics 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 Amazon Transcribe 의료와 다릅니다. 자세한 내용은 [AWS 일반](#) 참조를 참조하십시오.

### AWS Machine Learning 블로그를 통해 더 자세히 알아보기

Call Analytics 옵션에 대한 자세한 내용은 다음을 참조하세요.

- [Amazon 언어 AI 서비스를 사용하는 고객 센터의 사후 Call Analytics](#)
- [Amazon 언어 AI 서비스를 사용하는 고객 센터의 실시간 Call Analytics 및 에이전트 지원](#)

샘플 Call Analytics 출력 및 기능을 보려면 [GitHub데모](#)를 참조하십시오. 또한 스크립트를 형식으로 [변환할 수 있는 JSON-Word 문서](#) 응용 프로그램도 제공합니다. easy-to-read

## 리전 가용성 및 할당량

통화 분석은 다음에서 지원됩니다 AWS 리전.

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 Call Analytics에 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

Call Analytics는 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 단계, 다음 단계 등 주요 구성 요소를 캡처하여 전체 통화에 대한 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 전환율이 50% 이상 (두 발화자 back-and-forth 간) 이 지정된センチメント와 일치하면センチメント가 일치하는 Amazon Transcribe 것으로 간주합니다.
- 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 수정은 미국 영어 (en-US) 및 미국 스페인어 () 에서 지원됩니다. es-US

이 기능을 사용하여 수정된 PII 목록을 보거나 Amazon Transcribe를 사용한 수정에 대해 자세히 알아보려면 [식별 정보](#) 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 턴별로 제공됩니다.

이 지표는 통화가 끝날 때까지 에이전트가 화가 난 고객을 기쁘게 할 수 있는지 파악하는 데 도움이 될 수 있습니다.

감정 분석은 작동하므로 모델 교육 또는 사용자 지정 out-of-the-box 카테고리과 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.

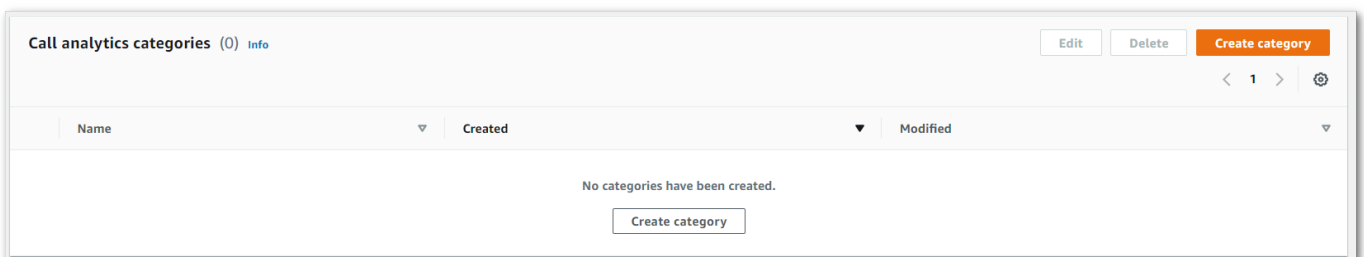
## 통화 후 범주와 실시간 범주의 비교

새 범주를 만들 때 해당 범주를 통화 후 분석 범주(POST\_CALL)로 생성할지 아니면 실시간 Call Analytics 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 Amazon Transcribe 통화 분석을 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.

- 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** Delete rule

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

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

Interruption time ▼

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

When the duration of the interruption was more than   ▼

during the  ▼

when the speaker was  ▼

AND

**▼ Rule 2** Delete rule

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

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

Transcript sentiment match ▼

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

When the sentiment is  ▼

during the  ▼   ▼

when the speaker was  ▼

AND

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](#) 단원을 참조하세요.

다음 예에서는 규칙을 사용하여 범주를 생성합니다.

- 고객은 처음 60,000밀리초 내에 방해를 받았습니다. 이러한 중단의 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.

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](#) 메서드의 `CategoryName` 및 `Rules` 인수를 사용하여 범주를 생성합니다. 자세한 내용은 [CreateCallAnalyticsCategory](#), [CategoryProperties](#), [Rule](#) 단원을 참조하세요.

기능별, 시나리오 및 크로스 서비스 예제를 포함하여 AWS SDK를 사용하는 추가 예제는 이 장을 참조하십시오. [SDK를 사용한 Amazon Transcribe의 코드 예제 AWS](#)

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 오디오와 Amazon Transcribe 일치시키려는 모든 [카테고리](#)를 만들어야 합니다.

### Note

Call Analytics 트랜스크립트를 새 범주에 소급 적용할 수는 없습니다. Call Analytics 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 통화 분석에서 Amazon Transcribe 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
<input type="radio"/>	CatchNegativeSentiment	POST_CALL	February 17 2023, 10:43 (UTC-08:00)	February 17 2023, 10:43 (UTC-08:00)

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

Cancel

Previous

Create job

3. 작업 세부 정보 지정 페이지에서는 입력 데이터의 위치를 포함하여 Call Analytics 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.

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.



원하는 출력 데이터 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

4. 다음을 선택합니다.
5. 작업 구성에서 Call Analytics 작업에 포함하려는 모든 옵션 기능을 켭니다. 이전에 생성한 범주는 범주 패널에 표시되며 Call Analytics 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
<input type="radio"/>	CatchNegativeSentiment	POST_CALL	February 17 2023, 10:43 (UTC-08:00)	February 17 2023, 10:43 (UTC-08:00)

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

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](#) 섹션을 참조하세요.

```
aws transcribe start-call-analytics-job \
--region us-west-2 \
--call-analytics-job-name my-first-call-analytics-job \
--media MediaFileUri=s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac \
--output-location s3://DOC-EXAMPLE-BUCKET/my-output-files/ \
--data-access-role-arn arn:aws:iam::111122223333:role/ExampleRole \
--channel-definitions ChannelId=0,ParticipantRole=AGENT
ChannelId=1,ParticipantRole=CUSTOMER
```

다음은 [start-call-analytics-job](#) 명령을 사용하는 또 다른 예 및 해당 작업의 Call Analytics를 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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
  },
  "OutputLocation": "s3://DOC-EXAMPLE-BUCKET/my-output-files/",
  "ChannelDefinitions": [
    {
      "ChannelId": 0,
      "ParticipantRole": "AGENT"
    },
    {
      "ChannelId": 1,
      "ParticipantRole": "CUSTOMER"
    }
  ]
}
```

```
]
}
```

## AWS SDK for Python (Boto3)

[이 예제에서는 AWS SDK for Python \(Boto3\) 를 사용하여 start\\_call\\_analytics\\_job 메서드를 사용하여 콜 애널리틱스 작업을 시작합니다.](#) 자세한 내용은 [StartCallAnalyticsJob](#) 및 [ChannelDefinition](#) 섹션을 참조하세요.

기능별, 시나리오 및 크로스 서비스 예제를 포함하여 AWS SDK를 사용하는 추가 예제는 이 장을 참조하십시오. [SDK를 사용한 Amazon Transcribe의 코드 예제 AWS](#)

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

while True:
    status = transcribe.get_call_analytics_job(CallAnalyticsJobName = job_name)
    if status['CallAnalyticsJob']['CallAnalyticsJobStatus'] in ['COMPLETED', 'FAILED']:
        break
```

```
print("Not ready yet...")
time.sleep(5)
print(status)
```

## 통화 후 분석 출력

통화 후 분석 기록은 세그먼트별 형식으로 표시됩니다. turn-by-turn 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 Call Analytics 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 경우에는 요약 생성 없이 분석 작업이 완료됩니다.

- 대화 내용이 충분하지 않음: 대화에는 상담원과 고객 모두의 차례가 한 번 이상 포함되어야 합니다. 대화 내용이 충분하지 않은 경우 서비스는 오류 코드 `INFFINCIENT_CONVERSATION_CONTENT` 를 반환합니다.
- 안전 가드레일: 적절한 요약이 생성되도록 하려면 대화가 안전 가드레일을 준수해야 합니다. 이러한 가드레일이 충족되지 않을 경우 서비스는 오류 코드 `FAILED_SAFETY_GIELIDEN`을 반환합니다.

오류 코드는 출력의 섹션에서 확인할 수 있습니다. [Skipped AnalyticsJobDetails](#) [GetCallAnalyticsJob](#) API 응답에서도 오류 원인을 찾을 수 있습니다.

`CallAnalyticsJobDetails`

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 turn-by-turn 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. 모든 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 기능인 생성 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 제공: [자동](#) 악용 탐지 AWS 기능을 구현합니다. 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
<input type="radio"/>	CatchNegativeSentiment	POST_CALL	February 17 2023, 10:43 (UTC-08:00)	February 17 2023, 10:43 (UTC-08:00)

## AWS CLI

이 예시에서는 [start-call-analytics-job](#) 명령과 Settings 파라미터 및 Summarization 하위 파라미터를 사용합니다. 자세한 정보는 [StartCallAnalyticsJob](#)을 참조하세요.

```
aws transcribe start-call-analytics-job \
--region us-west-2 \
--call-analytics-job-name my-first-call-analytics-job \
--media MediaFileUri=s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac \
--output-location s3://DOC-EXAMPLE-BUCKET/my-output-files/ \
--data-access-role-arn arn:aws:iam::111122223333:role/ExampleRole \
--channel-definitions ChannelId=0,ParticipantRole=AGENT
ChannelId=1,ParticipantRole=CUSTOMER
--settings '{"Summarization":{"GenerateAbstractiveSummary":true}}'
```

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

```

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

```

## AWS SDK for Python (Boto3)

[이 예제에서는 `start\_call\_analytics\_job` 메서드를 사용하여 요약이 활성화된 상태에서 통화 분석을 시작하는 AWS SDK for Python \(Boto3\) 데 사용합니다.](#) 자세한 정보는 [StartCallAnalyticsJob](#)을 참조하세요.

기능별, 시나리오 및 크로스 서비스 예제를 포함하여 AWS SDK를 사용하는 추가 예제는 이 장을 참조하십시오. [SDK를 사용한 Amazon Transcribe의 코드 예제 AWS](#)

```

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

```

```

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

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

```

## 실시간 Call Analytics

실시간 Call Analytics는 문제를 해결하고 에스컬레이션 발생 시 이를 완화하는 데 사용할 수 있는 실시간 인사이트를 제공합니다.

실시간 Call Analytics를 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](#)

실시간 통화 분석 외에도 미디어 스트림에서 [통화 후](#) 분석을 수행할 Amazon Transcribe 수 있습니다. [PostCallAnalyticsSettings](#) 파라미터를 사용하여 실시간 Call Analytics 요청에 통화 후 분석을 포함할 수 있습니다.

## 실시간 인사이트

이 섹션에서는 실시간 Call Analytics 트랜스크립션에 사용할 수 있는 인사이트를 자세히 설명합니다.

### 범주 이벤트

범주 이벤트를 사용하면 정확한 키워드나 구절을 기반으로 트랜스크립션을 일치시킬 수 있습니다. 예를 들어 '매니저와 대화하고 싶습니다'라는 문구에 필터를 설정하면 해당 문구를 필터링하여 정확히 일치하는 문구를 Amazon Transcribe 필터링할 수 있습니다.

다음은 [출력 예](#)입니다.

실시간 Call Analytics 범주를 만드는 방법에 대한 자세한 내용은 [실시간 트랜스크립션의 범주 생성 섹션](#)을 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. out-of-the-box 작동하므로 모델 교육 또는 사용자 지정 범주와 같은 사용자 지정을 지원하지 않습니다.

실시간 Call Analytics를 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 식별은 호주 (en-AU), 영국 (), 미국 () 및 스페인어 방언 (en-GB) 과 같은 영어 방언에서 지원됩니다. en-US es-US

실시간 Call Analytics를 통한 PII 식별은 각각의 오디오 세그먼트 전체에서 이루어집니다.

이 기능을 사용하여 식별되는 PII 목록을 보거나 PII 식별에 대한 자세한 내용은 [을 참조하십시오.](#)

Amazon Transcribe [식별 정보](#)

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

호주 (en-AU), 영국 (), 미국 (), 스페인어 방언 () 등의 영어 방언에서는 실시간 PII 수정이 지원됩니다. en-GB en-US es-US

실시간 Call Analytics를 통한 PII 수정은 각각의 오디오 세그먼트 전체에서 이루어집니다.

이 기능을 사용하여 수정된 PII 목록을 보거나 Amazon Transcribe를 사용한 수정에 대해 자세히 알아보려면 [식별 정보](#) 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 Call Analytics를 통한 감정 분석은 각각의 오디오 세그먼트 전체에서 이루어집니다.

감정 분석은 작동하므로 모델 교육 또는 사용자 지정 out-of-the-box 카테고리나 같은 사용자 지정을 지원하지 않습니다.

다음은 [출력 예](#)입니다.

## 실시간 트랜스크립션의 범주 생성

실시간 Call Analytics는 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 Call Analytics 트랜스크립션은 [TranscriptFilter](#)(키워드 일치)를 사용하는 규칙만 지원합니다. [CreateCallAnalyticsCategory](#) 작업에 규칙을 사용하는 방법에 대한 자세한 내용은 [실시간 Call Analytics 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 Call Analytics 범주를 생성한 경우 이 실시간 범주 일치에 대한 [이벤트 알림](#)을 설정하여 듀티 매니저에게 알릴 수 있습니다.

## 통화 후 범주와 실시간 범주의 비교

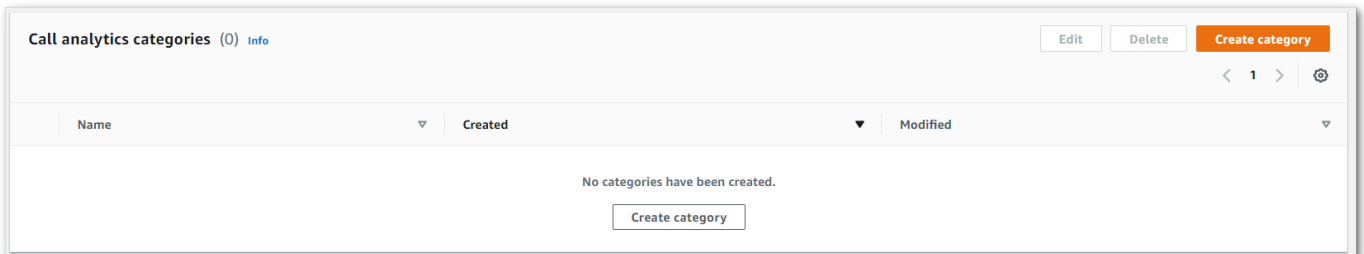
새 범주를 만들 때 해당 범주를 통화 후 범주(POST\_CALL)로 생성할지 아니면 실시간 범주(REAL\_TIME)로 생성할지 지정할 수 있습니다. 옵션을 지정하지 않으면 기본적으로 범주가 통화 후 범

주로 생성됩니다. 실시간 범주 일치를 사용하여 실시간 알림을 생성할 수 있습니다. 자세한 정보는 [범주 일치에 대한 실시간 알림 생성](#)을 참조하세요.

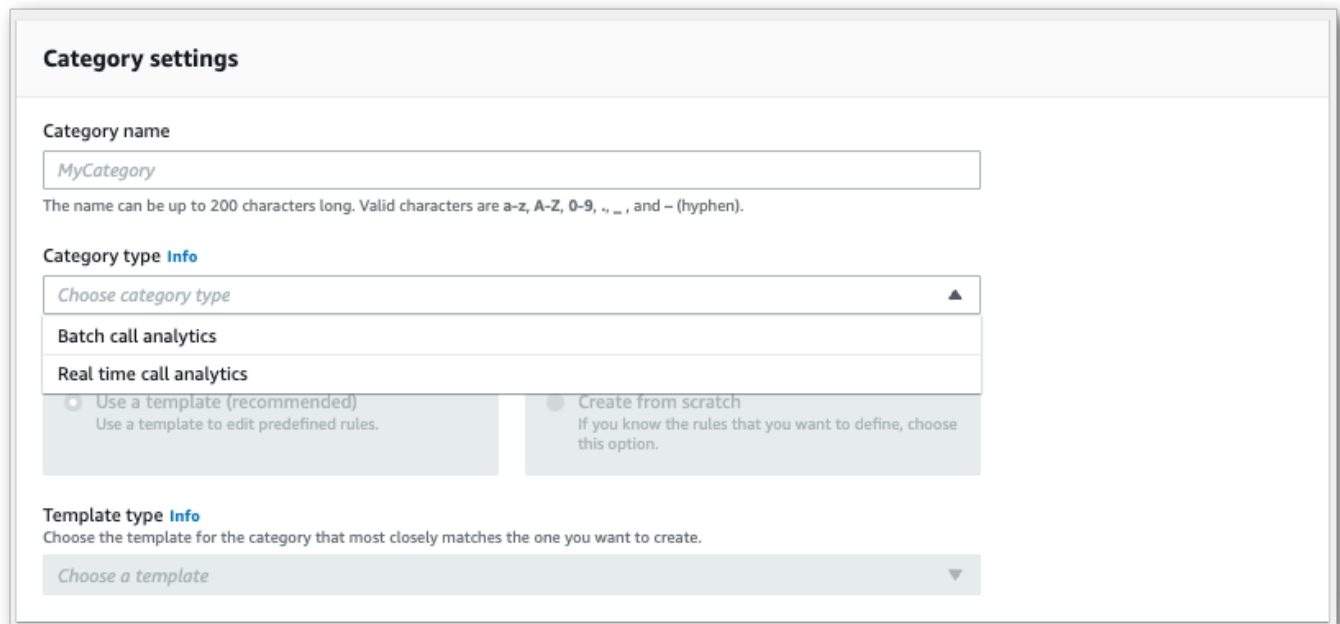
실시간 Call Analytics를 위한 새 범주를 생성하려면 AWS Management Console, AWS CLI 또는 AWS SDK를 사용할 수 있습니다. 예를 보려면 다음을 참조하세요.

## AWS Management Console

1. 탐색 창의 아래에서 Amazon Transcribe Amazon Transcribe 통화 분석을 선택합니다.
2. 통화 분석 범주를 선택하면 통화 분석 범주 페이지로 이동합니다. 범주 생성 버튼을 선택합니다.



3. 이제 범주 생성 페이지로 이동했습니다. 범주 이름을 입력한 다음 범주 유형 드롭다운 메뉴에서 '실시간 통화 분석'을 선택합니다.



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.

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.

6. 드롭다운 메뉴를 사용하여 범주에 규칙을 추가합니다. 범주당 최대 20개의 규칙을 추가할 수 있습니다. 실시간 Call Analytics 트랜스크립션에서는 트랜스크립트 콘텐츠 일치만 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.

7. 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](#) 단원을 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](#) 메서드의 `CategoryName` 및 `Rules` 인수를 사용하여 범주를 생성합니다. 자세한 내용은 [CreateCallAnalyticsCategory](#), [CategoryProperties](#), [Rule](#) 단원을 참조하세요.

기능별, 시나리오 및 크로스 서비스 예제를 포함하여 AWS SDK를 사용하는 추가 예제는 이 장을 참조하십시오. [SDK를 사용한 Amazon Transcribe의 코드 예제 AWS](#)

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 Call Analytics 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 Call Analytics 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 Call Analytics 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.

활성화된 경우 오디오 스트림의 통화 후 분석은 [오디오 파일에서 통화 후 분석](#)과 유사한 트랜스크립트를 생성하여 지정된 Amazon S3 버킷에 저장합니다. OutputLocation 또한 통화 후 분석은 오디오 스트림을 녹음하여 동일한 버킷에 오디오 파일(WAV형식)로 저장합니다. Amazon S3 수정을 활성화하면 수정된 대본과 수정된 오디오 파일도 지정된 버킷에 저장됩니다. Amazon S3 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-OGG 미디어를 사용하는 경우 트랜스크립트에 loudnessScore가 포함되지 않으며 스트림의 오디오 녹음도 생성되지 않습니다. 또한 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 Call Analytics 요청에 [PostCallAnalyticsSettings](#) 파라미터를 포함해야 합니다. PostCallAnalyticsSettings를 활성화할 때는 다음 파라미터를 포함해야 합니다.

- OutputLocation: 통화 후 Amazon S3 기록을 저장하려는 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 를 사용할 수 있습니다. WebSockets 예를 보려면 [실시간 Call Analytics 트랜스크립션 시작](#)을 참조하세요.

### Important

현재는 사전 로드된 오디오 예제가 포함된 실시간 통화 분석 데모를 제공하는 AWS Management Console 유일한 서비스입니다. 자체 오디오를 사용하려면 API (HTTP/2 또는 SDK) 를 사용해야 합니다. WebSockets

## 통화 후 분석 출력 예시

통화 후 기록은 세그먼트별 형식으로 표시됩니다. turn-by-turn 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 Call Analytics 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 Call Analytics 트랜스크립션 시작

실시간 통화 분석 트랜스크립션을 시작하기 전에 통화에서 Amazon Transcribe 매칭하려는 모든 [카테고리](#)를 만들어야 합니다.

### Note

Call Analytics 트랜스크립트를 새 범주에 소급 적용할 수는 없습니다. Call Analytics 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.

실시간 Call Analytics 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 Call Analytics 스트리밍 세션이 완료되면 별도의 트랜스크립트가 생성되어 지정된 버킷에 저장됩니다.

실시간 Call Analytics를 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 를 사용할 수 있습니다. 예를 보려면 다음을 참조하십시오 [WebSockets](#).

### ⚠ Important

현재는 사전 로드된 AWS Management Console 오디오 예제가 포함된 실시간 통화 분석 데모를 제공하는 유일한 서비스입니다. 자체 오디오를 사용하려면 API (HTTP/2 또는 SDK) 를 사용해야 합니다. [WebSockets](#)

## AWS Management Console

다음 절차에 따라 Call Analytics 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 통화 분석에서 Amazon Transcribe 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)	

3. 2단계: 통화 범주 검토에서는 이전에 만든 실시간 통화 분석 범주를 검토할 수 있습니다. 모든 실시간 Call Analytics 범주가 트랜스크립션에 적용됩니다.

범주 보기를 선택하면 새 창이 열리면서 기존 실시간 Call Analytics 범주를 표시하고 새 범주를 생성할 수 있는 링크를 제공합니다.

i Retry this demo after [creating your own custom categories](#).

### Call analytics categories (4) [Info](#)

0 matches < 1 > ⚙

**Type: Real time call analytics** ✕

Clear filters

Name	Type	Created	Modified
No matches We can't find a match. <div style="margin: 10px auto; border: 1px solid #ccc; padding: 5px 15px;">Clear filter</div>			

4. 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)**

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

**Personal (5 of 5 selected)**

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

i 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, 발연 속도, 발연 시간, 문제, 조치 항목 및 결과를 포함하여 통화 후 분석 트랜스크립션과 동일한 분석을 얻을 수 있습니다. 통화 후 분석 출력은 실시간 Call Analytics 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 대상 및 역할을 지정해야 합니다. IAM 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

Format: s3://bucket, s3://bucket/prefix/, or s3://bucket/prefix/object.

Encryption [Info](#)

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

[Create an IAM role](#) that grants access to the output bucket and KMS key (if specified) with the trust policy shown below

▶ Trust Policy

## 6. 스트리밍 시작을 선택합니다.

### HTTP/2 스트림

이 예시에서는 Call Analytics가 활성화된 상태에서 HTTP/2 요청을 생성합니다. HTTP/2 스트리밍을 사용하는 방법에 대한 자세한 내용은 [을 Amazon Transcribe참조하십시오.](#) [HTTP/2 스트림 설정](#) 관련 파라미터 및 헤더에 대한 자세한 내용은 [을 참조하십시오.](#) [Amazon TranscribeStartCallAnalyticsStreamTranscription](#)

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://DOC-EXAMPLE-BUCKET/my-output-files/",
        "DataAccessRoleArn": "arn:aws:iam::111122223333:role/ExampleRole"
      }
    }
  }
}

```

매개변수 정의는 [API 참조에서 찾을 수 있으며](#), 모든 AWS API 작업에 공통적인 매개변수는 [공통 매개변수](#) 섹션에 나열되어 있습니다.

## WebSocket 스트림

이 예제는 WebSocket 스트림에서 통화 분석을 사용하는 미리 서명된 URL을 만듭니다. 가독성을 높이기 위해 줄바꿈이 추가되었습니다. 에서 WebSocket 스트림을 사용하는 방법에 대한 자세한 내용은 Amazon Transcribe을 참조하십시오 [WebSocket 스트림 설정](#). 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://DOC-EXAMPLE-BUCKET/my-output-files/",
        "DataAccessRoleArn": "arn:aws:iam::111122223333:role/ExampleRole"
      }
    }
  }
}

```

매개변수 정의는 [API 참조에서 찾을 수 있으며](#), 모든 AWS API 작업에 공통적인 [매개변수는 공통 매개변수](#) 섹션에 나열되어 있습니다.

#### Tip

위의 HTTP/2 및 WebSocket 예에는 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 정보는 [스트리밍 트랜스크립션 설정](#)을 참조하세요.

## 범주 일치에 대한 실시간 알림 생성

실시간 알림을 설정하려면 먼저 REAL\_TIME 플래그가 있는 [TranscriptFilterType](#) 범주를 생성해야 합니다. 이 플래그를 사용하면 범주를 실시간 Call Analytics 트랜스크립션에 적용할 수 있습니다.

새 범주 생성에 대한 지침은 [실시간 트랜스크립션의 범주 생성](#) 섹션을 참조하세요.

실시간 Call Analytics 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 Call Analytics 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 Call Analytics 출력

실시간 통화 분석 기록은 세그먼트별 turn-by-turn 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 Call Analytics 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 Call Analytics 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
    }
  ]
},

```

## 컴파일된 실시간 Call Analytics 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
    }
  }
  ],
  "CategoryEvent": {
    "MatchedCategories": [
      "network-complaint"
    ],
    "MatchedDetails": {
      "network issues" : {
        "TimestampRanges": [
          {
            "BeginOffsetMillis": 9299375,
            "EndOffsetMillis": 7899375
          }
        ]
      }
    }
  }
}
```

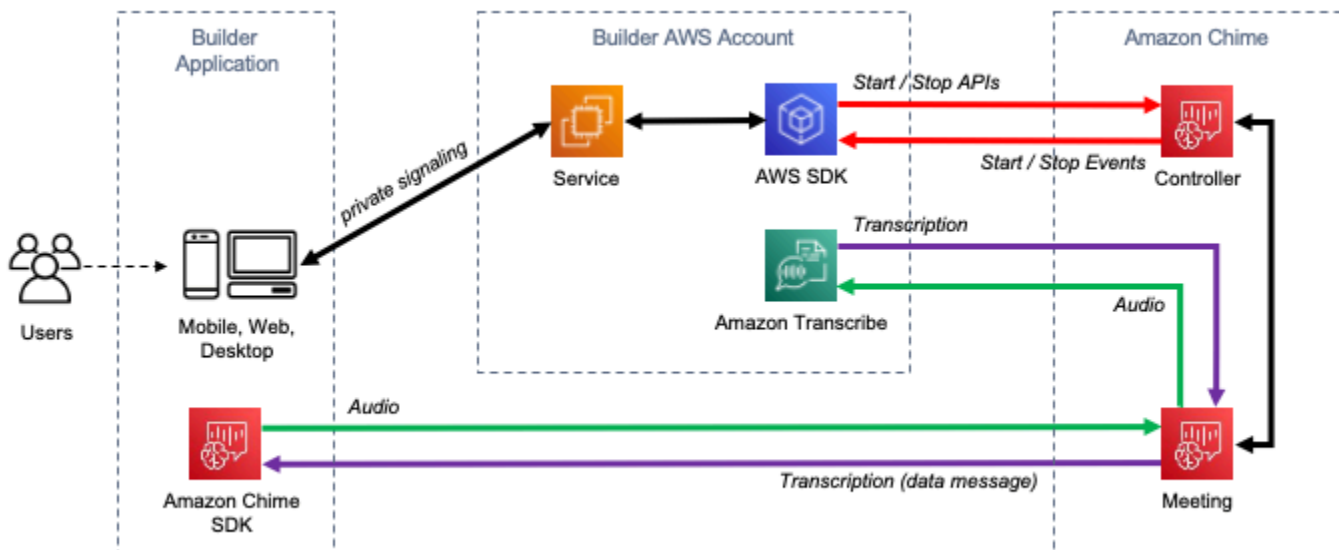
## Amazon Chime 통화 내용을 실시간으로 받아쓰기

Amazon Transcribe Amazon Chime SDK와 통합되어 Amazon Chime 통화를 실시간으로 기록할 수 있습니다.

Amazon Chime SDK API를 사용하여 트랜스크립션을 요청하면 는 오디오 스트리밍을 Amazon Chime 시작하고 통화 중에 계속 스트리밍합니다. Amazon Transcribe

Amazon Chime SDK는 'Active Talker' 알고리즘을 사용하여 상위 두 명의 활성 발화자를 선택한 다음 해당 오디오를 단일 스트림을 통해 두 개의 개별 Amazon Transcribe 채널로 전송합니다. 미팅 참가자는 Amazon Chime SDK 데이터 메시지를 통해 사용자 어트리뷰션 트랜스크립션을 받습니다. [Amazon Chime SDK 개발자 가이드에서](#) 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 SDK 개발자 가이드의 [Amazon Chime SDK 라이브 트랜스크립션 사용](#)을 참조하십시오. Amazon Chime API 작업에 대해서는 [Amazon Chime SDK API 참조](#)를 참조하십시오.

### 📌 AWS Machine Learning 블로그를 통해 더 자세히 알아보기

실시간 트랜스크립션을 통한 정확도 개선에 대한 자세한 내용은 다음을 참조하십시오.

- [Amazon Chime SDK 미팅은 이제 Amazon Transcribe 및 를 통한 실시간 트랜스크립션을 지원합니다. Amazon Transcribe Medical](#)

- [Amazon Chime원격 진료 솔루션을 위한 SDK](#)

## SDK를 사용한 Amazon Transcribe의 코드 예제 AWS

다음 코드 예제는 AWS 소프트웨어 개발 키트 (SDK) 와 함께 Amazon Transcribe를 사용하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 호출하는 방법을 보여 주며 관련 시나리오와 교차 서비스 예시에서 컨텍스트에 맞는 작업을 볼 수 있습니다.

시나리오는 동일한 서비스 내에서 여러 함수를 호출하여 특정 태스크를 수행하는 방법을 보여주는 코드 예시입니다.

교차 서비스 예시는 여러 AWS 서비스전반에서 작동하는 샘플 애플리케이션입니다.

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 을 참조하십시오. [SDK와 함께 이 서비스 사용 AWS](#) 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

### 코드 예시

- [SDK를 사용한 Amazon Transcribe 작업 AWS](#)
  - [AWS SDK 또는 CreateVocabulary CLI와 함께 사용](#)
  - [AWS SDK 또는 DeleteMedicalTranscriptionJob CLI와 함께 사용](#)
  - [AWS SDK 또는 DeleteTranscriptionJob CLI와 함께 사용](#)
  - [AWS SDK 또는 DeleteVocabulary CLI와 함께 사용](#)
  - [AWS SDK 또는 GetTranscriptionJob CLI와 함께 사용](#)
  - [AWS SDK 또는 GetVocabulary CLI와 함께 사용](#)
  - [AWS SDK 또는 ListMedicalTranscriptionJobs CLI와 함께 사용](#)
  - [AWS SDK 또는 ListTranscriptionJobs CLI와 함께 사용](#)
  - [AWS SDK 또는 ListVocabularies CLI와 함께 사용](#)
  - [AWS SDK 또는 StartMedicalTranscriptionJob CLI와 함께 사용](#)
  - [AWS SDK 또는 StartStreamTranscriptionAsync CLI와 함께 사용](#)
  - [AWS SDK 또는 StartTranscriptionJob CLI와 함께 사용](#)
  - [AWS SDK 또는 UpdateVocabulary CLI와 함께 사용](#)
- [SDK를 사용한 Amazon Transcribe 시나리오 AWS](#)
  - [SDK를 사용하여 Amazon Transcribe 사용자 지정 어휘를 만들고 수정하십시오. AWS](#)

- [SDK를 사용하여 Amazon Transcribe에서 오디오를 텍스트로 변환하고 작업 데이터를 가져올 수 있습니다. AWS](#)
- [SDK를 사용한 Amazon AWS Transcribe의 크로스 서비스 예제](#)
  - [Amazon Transcribe 앱 구축](#)
  - [Amazon Transcribe 스트리밍 앱 구축](#)
  - [SDK를 사용하여 텍스트를 음성으로 변환하고 다시 텍스트로 변환합니다. AWS](#)

## SDK를 사용한 Amazon Transcribe 작업 AWS

다음 코드 예제는 SDK를 사용하여 개별 Amazon Transcribe 작업을 수행하는 방법을 보여줍니다. AWS 이클 발체문은 Amazon Transcribe API를 호출하며, 컨텍스트에서 실행되어야 하는 더 큰 프로그램에서 발체한 코드입니다. 각 예제에는 코드 GitHub 설정 및 실행 지침을 찾을 수 있는 링크가 포함되어 있습니다.

다음 예제에는 가장 일반적으로 사용되는 작업만 포함되어 있습니다. 전체 목록은 [Amazon Transcribe API 참조](#)를 참조하세요.

### 예제

- [AWS SDK 또는 CreateVocabulary CLI와 함께 사용](#)
- [AWS SDK 또는 DeleteMedicalTranscriptionJob CLI와 함께 사용](#)
- [AWS SDK 또는 DeleteTranscriptionJob CLI와 함께 사용](#)
- [AWS SDK 또는 DeleteVocabulary CLI와 함께 사용](#)
- [AWS SDK 또는 GetTranscriptionJob CLI와 함께 사용](#)
- [AWS SDK 또는 GetVocabulary CLI와 함께 사용](#)
- [AWS SDK 또는 ListMedicalTranscriptionJobs CLI와 함께 사용](#)
- [AWS SDK 또는 ListTranscriptionJobs CLI와 함께 사용](#)
- [AWS SDK 또는 ListVocabularies CLI와 함께 사용](#)
- [AWS SDK 또는 StartMedicalTranscriptionJob CLI와 함께 사용](#)
- [AWS SDK 또는 StartStreamTranscriptionAsync CLI와 함께 사용](#)
- [AWS SDK 또는 StartTranscriptionJob CLI와 함께 사용](#)
- [AWS SDK 또는 UpdateVocabulary CLI와 함께 사용](#)



## AWS SDK 또는 **CreateVocabulary** CLI와 함께 사용

다음 코드 예제는 CreateVocabulary의 사용 방법을 보여줍니다.

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

#### AWS SDK for .NET

##### Note

더 많은 정보가 있습니다 GitHub. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배우보세요.

```
/// <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 [CreateVocabulary](#)참조를 참조하십시오.

## CLI

### AWS CLI

#### 사용자 지정 어휘를 생성하는 방법

다음 `create-vocabulary` 예시에서는 사용자 지정 어휘를 생성합니다. 사용자 지정 어휘를 생성하려면 더 정확하게 트랜스크립션하려는 모든 용어가 포함된 텍스트 파일을 만들어야 합니다. `vocabularly-file-uri` 경우 해당 텍스트 파일의 Amazon Simple Storage 서비스 (Amazon S3) URI를 지정하십시오. `language-code`의 경우 사용자 지정 어휘의 언어에 해당하는 언어 코드를 지정합니다. `vocabulary-name`의 경우 사용자 지정 어휘의 이름을 지정합니다.

```
aws transcribe create-vocabulary \  
  --language-code language-code \  
  --vocabulary-name cli-vocab-example \  
  --vocabulary-file-uri s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/the-text-file-  
for-the-custom-vocabulary.txt
```

#### 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 [CreateVocabulary](#)참조를 참조하십시오.

## Python

## SDK for Python(Boto3)

 Note

자세한 내용은 [에서 확인할 수 GitHub](#) 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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

```

- API에 대한 자세한 내용은 파이썬용AWS SDK (Boto3) API 레퍼런스를 참조하십시오 [CreateVocabulary](#).

AWS SDK 개발자 가이드 및 코드 예제의 전체 목록은 을 참조하십시오. [SDK와 함께 이 서비스 사용 AWS](#) 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

## AWS SDK 또는 **DeleteMedicalTranscriptionJob** CLI와 함께 사용

다음 코드 예제는 DeleteMedicalTranscriptionJob의 사용 방법을 보여줍니다.

.NET

AWS SDK for .NET

### Note

더 많은 정보가 있습니다 GitHub. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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 [DeleteMedicalTranscriptionJob](#)참조를 참조하십시오.

## CLI

### AWS CLI

의료 트랜스크립션 작업을 삭제하는 방법

다음 delete-medical-transcription-job 예시에서는 의료 트랜스크립션 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 안내서를 참조하십시오

[DeleteMedicalTranscriptionJob](#).

- API 세부 정보는 AWS CLI 명령 [DeleteMedicalTranscriptionJob](#)참조를 참조하십시오.

## JavaScript

### JavaScript (v3) 용 SDK

#### Note

더 많은 내용이 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배우보세요.

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 트랜스크립션 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 참조하십시오.
- API 세부 정보는 AWS SDK for JavaScript API [DeleteMedicalTranscriptionJob](#)참조를 참조하십시오.

AWS SDK 개발자 가이드 및 코드 예제의 전체 목록은 [을 참조하십시오](#)[SDK와 함께 이 서비스 사용 AWS](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

## AWS SDK 또는 **DeleteTranscriptionJob** CLI와 함께 사용

다음 코드 예제는 DeleteTranscriptionJob의 사용 방법을 보여줍니다.

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

AWS SDK for .NET

### Note

더 많은 정보가 있습니다 GitHub. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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 [DeleteTranscriptionJob](#)참조를 참조하십시오.

## CLI

### AWS CLI

트랜스크립션 작업 중 하나를 삭제하는 방법

다음 delete-transcription-job 예시에서는 트랜스크립션 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 안내서를 참조하십시오 [DeleteTranscriptionJob](#).

- API 세부 정보는 AWS CLI 명령 [DeleteTranscriptionJob](#)참조를 참조하십시오.

## JavaScript

### JavaScript (v3) 용 SDK

#### Note

더 많은 내용이 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배워보세요.

트랜스크립션 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 참조하십시오.
- API 세부 정보는 AWS SDK for JavaScript API [DeleteTranscriptionJob](#)참조를 참조하십시오.

## Python

### SDK for Python(Boto3)

#### Note

자세한 내용은 에서 확인할 수 GitHub 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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

```

- API에 대한 자세한 내용은 파이썬용 AWS SDK (Boto3) API 레퍼런스를 참조하십시오 [DeleteTranscriptionJob](#).

AWS SDK 개발자 가이드 및 코드 예제의 전체 목록은 을 참조하십시오. [SDK와 함께 이 서비스 사용 AWS](#) 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

## AWS SDK 또는 **DeleteVocabulary** CLI와 함께 사용

다음 코드 예제는 DeleteVocabulary의 사용 방법을 보여줍니다.

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

AWS SDK for .NET

### Note

더 많은 정보가 있습니다 GitHub. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배우보세요.

```

/// <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 [DeleteVocabulary](#)참조를 참조하십시오.

## 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 [DeleteVocabulary](#)참조를 참조하십시오.

## Python

## SDK for Python(Boto3)

**Note**

자세한 내용은 에서 확인할 수 GitHub 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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
```

- API에 대한 자세한 내용은 파이썬용AWS SDK (Boto3) API 레퍼런스를 참조하십시오 [DeleteVocabulary](#).

AWS SDK 개발자 가이드 및 코드 예제의 전체 목록은 을 참조하십시오. [SDK와 함께 이 서비스 사용 AWS](#) 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

## AWS SDK 또는 **GetTranscriptionJob** CLI와 함께 사용

다음 코드 예제는 GetTranscriptionJob의 사용 방법을 보여줍니다.


작업 예시는 대규모 프로그램에서 발체한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 다음 코드 예제에서는 컨텍스트 내에서 이 작업을 확인할 수 있습니다.

- [사용자 지정 어휘 생성 및 세부 조정](#)

- [오디오 트랜스크립션 및 작업 데이터 가져오기](#)

## .NET

## AWS SDK for .NET

 Note

더 많은 정보가 있습니다 GitHub. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배우보세요.

```

/// <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;
}

```

- API 세부 정보는 AWS SDK for .NET API [GetTranscriptionJob](#)참조를 참조하십시오.

## CLI

## AWS CLI

특정 트랜스크립션 작업에 대한 정보를 가져오려면

다음 `get-transcription-job` 예시에서는 특정 트랜스크립션 작업에 대한 정보를 가져옵니다. 트랜스크립션 결과에 액세스하려면 `TranscriptFileUri` 파라미터를 사용하세요. `MediaFileUri`

파라미터를 사용하여 이 작업에서 어떤 오디오 파일을 트랜스크립션했는지 확인할 수 있습니다. Settings 객체를 사용하여 트랜스크립션 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://DOC-EXAMPLE-BUCKET/your-audio-file.file-  
extension"  
    },  
    "Transcript": {  
      "TranscriptFileUri": "https://Amazon-S3-file-location-of-  
transcription-output"  
    },  
    "StartTime": "2020-09-18T22:27:23.970000+00:00",  
    "CreationTime": "2020-09-18T22:27:23.948000+00:00",  
    "CompletionTime": "2020-09-18T22:28:21.197000+00:00",  
    "Settings": {  
      "ChannelIdentification": false,  
      "ShowAlternatives": false  
    },  
    "IdentifyLanguage": true,  
    "IdentifiedLanguageScore": 0.8672199249267578  
  }  
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 [시작하기 \(AWS 명령줄 인터페이스\)](#) 를 참조하십시오.

- API 세부 정보는 AWS CLI 명령 [GetTranscriptionJob](#) 참조를 참조하십시오.

## Python

### SDK for Python(Boto3)

#### Note

자세한 내용은 에서 확인할 수 GitHub 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배우보세요.

```
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
```

- API에 대한 자세한 내용은 파이썬용AWS SDK (Boto3) API 레퍼런스를 참조하십시오 [GetTranscriptionJob](#).

AWS SDK 개발자 가이드 및 코드 예제의 전체 목록은 을 참조하십시오. [SDK와 함께 이 서비스 사용 AWS](#) 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

## AWS SDK 또는 **GetVocabulary** CLI와 함께 사용

다음 코드 예제는 GetVocabulary의 사용 방법을 보여줍니다.

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

#### AWS SDK for .NET

##### Note

더 많은 정보가 있습니다 GitHub. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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 [GetVocabulary](#)참조를 참조하십시오.



## CLI

### AWS CLI

사용자 지정 어휘에 대한 정보를 가져오려면

다음 `get-vocabulary` 예시에서는 이전에 생성한 사용자 지정 어휘에 대한 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 [GetVocabulary](#)참조를 참조하십시오.

## Python

### SDK for Python(Boto3)

#### Note

자세한 내용은 에서 확인할 수 GitHub 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배우보세요.

```
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

```

- API에 대한 자세한 내용은 파이썬용AWS SDK (Boto3) API 레퍼런스를 참조하십시오 [GetVocabulary](#).

AWS SDK 개발자 가이드 및 코드 예제의 전체 목록은 을 참조하십시오. [SDK와 함께 이 서비스 사용 AWS](#) 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

## AWS SDK 또는 **ListMedicalTranscriptionJobs** CLI와 함께 사용

다음 코드 예제는 ListMedicalTranscriptionJobs의 사용 방법을 보여줍니다.

.NET

AWS SDK for .NET

### Note

더 많은 정보가 있습니다 GitHub. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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 [ListMedicalTranscriptionJobs](#) 참조를 참조하십시오.

## CLI

### AWS CLI

의료 트랜스크립션 작업을 나열하는 방법

다음 `list-medical-transcription-jobs` 예시는 AWS 계정 및 지역과 관련된 의료 기록 작업을 나열합니다. 특정 트랜스크립션 작업에 대한 자세한 내용을 보려면 트랜스크립션 출력의 `MedicalTranscriptionJobName` 파라미터 값을 복사하고 명령 `MedicalTranscriptionJobName` 옵션에 해당 값을 지정하십시오. `get-medical-transcription-job` 더 많은 트랜스크립션 작업을 보려면 `NextToken` 매개 변수의 값을 복사하고 `list-medical-transcription-jobs` 명령을 다시 실행한 다음 옵션에 해당 값을 지정하십시오. `--next-token`

```
aws transcribe list-medical-transcription-jobs
```

출력:

```
{
  "NextToken": "3/PblzkiGhzjER3KHuQt2fmbPLF7cDYafjFMEoGn440N/
gsuUSTIkGyanvRE6WMXfd/ZTEc2EZj+P9eii/"
}
```

```

z102FDY1i6RLI0WoRX4RwMisVrh9G0Kie0Y8ikBCdtq1ZB10Wa9McC+eb01
+LaDtZPC4u6ttoHLRLEfzqstHXSgapXg3tEBtm9piIaPB6M0M5BB6t86+qtmocTR/
qrteHZBBudhTfbCwhsxaqujHiiUvFdm3BQbKKWIW06yV9b+4f38oD21VIan
+vfUs3gBYA15VTDmXXzQPbQ0HPjtwmFI+IWX15nSUjWuN3TUy1HgPWzDaYT8qBtu0Z+3UG4V6b
+K2CC0XszXg5rBq9hYgNzy4XoFh/6s5DoSnzq49Q9xHgHdT2yBADFmvFK7myZBs75+2vQZ0SVpWUPy3WT/32zFAc
+mFYfUjtTZ8n/jq7aQEjQ42A
+X/7K6Jg0cdVPtEg8P1Dr5kgYYG3q30mYXX37U3FZuJmnTI63VtIXsNn0U5eGoY0btpk00Nq9UkzgjSxqj84ZD5n
+S0EGy9ZUYBJRRcGeYUM3Q4DbSjFuwSaQcFdLIWZdp8qIREMQIBWY7BLwSdyqsQo2vRrd53hm5aWM7SVf6pPq6X/
IXR5+1eU00D8/coaTT4ES2DerbV6RkV4o0VT1d0SdVX/
MmtkNG8nYj8PqU07w7988quh1ZP6D80veJS1q73tUUR9MjnGernW2tAnvnLNhdefBcD
+sZVfYq3iBMFY7wTy1P1G6NqW9GrYDYox3tTPW1D7phpbVSyKrh/
PdYrps5UxnsGoA1b7L/FfAXDfUoGrGUB4N3JsPYXX9D++g+6gV1qBBs/
WfF934aKqfD6UTggm/zV3GA0WiBpfvAZRvEb924i6yGHYMC7y5401ZAwSBupmI
+FFd13CaP04kN1vJlth6aM5vUPXg4BpyUhtbRhwD/KxCvf9K0tLJGyL1A=="
  "MedicalTranscriptionJobSummaries": [
    {
      "MedicalTranscriptionJobName": "vocabulary-dictation-medical-
transcription-job",
      "CreationTime": "2020-09-21T21:17:27.016000+00:00",
      "StartTime": "2020-09-21T21:17:27.045000+00:00",
      "CompletionTime": "2020-09-21T21:17:59.561000+00:00",
      "LanguageCode": "en-US",
      "TranscriptionJobStatus": "COMPLETED",
      "OutputLocationType": "CUSTOMER_BUCKET",
      "Specialty": "PRIMARYCARE",
      "Type": "DICTATION"
    },
    {
      "MedicalTranscriptionJobName": "alternatives-dictation-medical-
transcription-job",
      "CreationTime": "2020-09-21T21:01:14.569000+00:00",
      "StartTime": "2020-09-21T21:01:14.592000+00:00",
      "CompletionTime": "2020-09-21T21:01:43.606000+00:00",
      "LanguageCode": "en-US",
      "TranscriptionJobStatus": "COMPLETED",
      "OutputLocationType": "CUSTOMER_BUCKET",
      "Specialty": "PRIMARYCARE",
      "Type": "DICTATION"
    },
    {
      "MedicalTranscriptionJobName": "alternatives-conversation-medical-
transcription-job",
      "CreationTime": "2020-09-21T19:09:18.171000+00:00",
      "StartTime": "2020-09-21T19:09:18.199000+00:00",

```

```

        "CompletionTime": "2020-09-21T19:10:22.516000+00:00",
        "LanguageCode": "en-US",
        "TranscriptionJobStatus": "COMPLETED",
        "OutputLocationType": "CUSTOMER_BUCKET",
        "Specialty": "PRIMARYCARE",
        "Type": "CONVERSATION"
    },
    {
        "MedicalTranscriptionJobName": "speaker-id-conversation-medical-
transcription-job",
        "CreationTime": "2020-09-21T18:43:37.157000+00:00",
        "StartTime": "2020-09-21T18:43:37.265000+00:00",
        "CompletionTime": "2020-09-21T18:44:21.192000+00:00",
        "LanguageCode": "en-US",
        "TranscriptionJobStatus": "COMPLETED",
        "OutputLocationType": "CUSTOMER_BUCKET",
        "Specialty": "PRIMARYCARE",
        "Type": "CONVERSATION"
    },
    {
        "MedicalTranscriptionJobName": "multichannel-conversation-medical-
transcription-job",
        "CreationTime": "2020-09-20T23:46:44.053000+00:00",
        "StartTime": "2020-09-20T23:46:44.081000+00:00",
        "CompletionTime": "2020-09-20T23:47:35.851000+00:00",
        "LanguageCode": "en-US",
        "TranscriptionJobStatus": "COMPLETED",
        "OutputLocationType": "CUSTOMER_BUCKET",
        "Specialty": "PRIMARYCARE",
        "Type": "CONVERSATION"
    }
]
}


```

자세한 내용은 Amazon Transcribe 개발자 가이드의 <https://docs.aws.amazon.com/transcribe/latest/dg/batch-med-transcription.html>을 참조하십시오.

- API 세부 정보는 명령 참조를 참조하십시오 [ListMedicalTranscriptionJobs](#).AWS CLI

## JavaScript

## JavaScript (v3) 용 SDK

 Note

더 많은 내용이 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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 트랜스크립션 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 참조하십시오.
- API 세부 정보는 AWS SDK for JavaScript API [ListMedicalTranscriptionJobs](#)참조를 참조하십시오.

AWS SDK 개발자 가이드 및 코드 예제의 전체 목록은 [을 참조하십시오](#)[SDK와 함께 이 서비스 사용 AWS](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

## AWS SDK 또는 **ListTranscriptionJobs** CLI와 함께 사용

다음 코드 예제는 ListTranscriptionJobs의 사용 방법을 보여줍니다.

.NET

AWS SDK for .NET

### Note

더 많은 정보가 있습니다 GitHub. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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 [ListTranscriptionJobs](#)참조를 참조하십시오.

## CLI

### AWS CLI

트랜스크립션 작업을 나열하는 방법

다음 `list-transcription-jobs` 예시는 AWS 계정 및 지역과 관련된 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 참조하십시오.

- API 세부 정보는 AWS CLI 명령 [ListTranscriptionJobs](#) 참조를 참조하십시오.

## Java

## SDK for Java 2.x

 Note

자세한 내용은 에서 확인할 수 GitHub 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배우보세요.

```
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 [ListTranscriptionJobs](#) 참조를 참조하십시오.

## JavaScript

### JavaScript (v3) 용 SDK

#### Note

더 많은 내용이 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배워보세요.

트랜스크립션 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 참조하십시오.
- API 세부 정보는 AWS SDK for JavaScript API [ListTranscriptionJobs](#)참조를 참조하십시오.

## Python

### SDK for Python(Boto3)

#### Note

자세한 내용은 에서 확인할 수 GitHub 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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

```

- API에 대한 자세한 내용은 파이썬용 AWS SDK (Boto3) API 레퍼런스를 참조하십시오 [ListTranscriptionJobs](#).

AWS SDK 개발자 가이드 및 코드 예제의 전체 목록은 을 참조하십시오. [SDK와 함께 이 서비스 사용 AWS](#) 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

## AWS SDK 또는 **ListVocabularies** CLI와 함께 사용

다음 코드 예제는 ListVocabularies의 사용 방법을 보여줍니다.

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

### AWS SDK for .NET

#### Note

더 많은 정보가 있습니다 GitHub. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배우보세요.

```

    /// <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 [ListVocabularies](#)참조를 참조하십시오.

## CLI

### AWS CLI

사용자 지정 어휘를 나열하는 방법

다음 `list-vocabularies` 예시는 AWS 계정 및 지역과 관련된 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 [ListVocabularies](#)참조를 참조하십시오.

## Python

### SDK for Python(Boto3)

#### Note

자세한 내용은 에서 확인할 수 GitHub 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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

```

- API에 대한 자세한 내용은 파이썬용 AWS SDK (Boto3) API 레퍼런스를 참조하십시오 [ListVocabularies](#).

AWS SDK 개발자 가이드 및 코드 예제의 전체 목록은 을 참조하십시오. [SDK와 함께 이 서비스 사용 AWS](#) 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

## AWS SDK 또는 **StartMedicalTranscriptionJob** CLI와 함께 사용

다음 코드 예제는 StartMedicalTranscriptionJob의 사용 방법을 보여줍니다.

.NET

AWS SDK for .NET

### Note

더 많은 정보가 있습니다 GitHub. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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 [StartMedicalTranscriptionJob](#)참조를 참조하십시오.

## CLI

### AWS CLI

예 1: 오디오 파일로 저장된 의료 구술을 트랜스크립션하는 방법

다음 `start-medical-transcription-job` 예시에서는 오디오 파일을 트랜스크립션합니다. `OutputBucketName` 파라미터에는 트랜스크립션 출력의 위치를 지정합니다.

```
aws transcribe start-medical-transcription-job \
  --cli-input-json file://myfile.json
```

`myfile.json`의 콘텐츠:

```
{
  "MedicalTranscriptionJobName": "simple-dictation-medical-transcription-job",
  "LanguageCode": "language-code",
  "Specialty": "PRIMARYCARE",
  "Type": "DICTATION",
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
  }
}
```

출력:

```
{
  "MedicalTranscriptionJob": {
    "MedicalTranscriptionJobName": "simple-dictation-medical-transcription-job",
    "TranscriptionJobStatus": "IN_PROGRESS",
    "LanguageCode": "language-code",
    "Media": {
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
    },
    "StartTime": "2020-09-20T00:35:22.256000+00:00",
    "CreationTime": "2020-09-20T00:35:22.218000+00:00",
    "Specialty": "PRIMARYCARE",
    "Type": "DICTATION"
  }
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 [배치 트랜스크립션 개요](#)를 참조하세요.

예 2: 오디오 파일로 저장된 의사와 환자 간 대화를 트랜스크립션하는 방법

다음 `start-medical-transcription-job` 예시에서는 의사와 환자 간 대화가 포함된 오디오 파일을 트랜스크립션합니다. `OutputBucketName` 파라미터에 트랜스크립션 출력 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": "DOC-EXAMPLE-BUCKET",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
  }
}
```

출력:

```
{
  "MedicalTranscriptionJob": {
    "MedicalTranscriptionJobName": "simple-conversation-medical-transcription-job",
    "TranscriptionJobStatus": "IN_PROGRESS",
    "LanguageCode": "language-code",
    "Media": {
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
    },
    "StartTime": "2020-09-20T23:19:49.965000+00:00",
    "CreationTime": "2020-09-20T23:19:49.941000+00:00",
    "Specialty": "PRIMARYCARE",
    "Type": "CONVERSATION"
  }
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 [배치 트랜스크립션 개요](#)를 참조하세요.

예 3: 의사와 환자 간 대화의 멀티채널 오디오 파일을 트랜스크립션하는 방법

다음 `start-medical-transcription-job` 예시에서는 오디오 파일에 있는 각 채널의 오디오를 트랜스크립션하고 각 채널의 개별 트랜스크립션을 단일 트랜스크립션 출력으로 병합합니다. `OutputBucketName` 파라미터에는 트랜스크립션 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": "DOC-EXAMPLE-BUCKET",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-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://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
    },
    "StartTime": "2020-09-20T23:46:44.081000+00:00",
    "CreationTime": "2020-09-20T23:46:44.053000+00:00",
    "Settings": {
      "ChannelIdentification": true
    },
    "Specialty": "PRIMARYCARE",
    "Type": "CONVERSATION"
  }
}
```

```
}
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 [채널 식별](#)을 참조하세요.

예 4: 의사와 환자 간 대화의 오디오 파일을 트랜스크립션하고 트랜스크립션 출력에서 화자를 식별하는 방법

다음 `start-medical-transcription-job` 예시에서는 오디오 파일을 트랜스크립션하고 트랜스크립션 출력에서 각 화자의 음성에 레이블을 지정합니다. `OutputBucketName` 파라미터에는 트랜스크립션 출력의 위치를 지정합니다.

```
aws transcribe start-medical-transcription-job \
  --cli-input-json file://myfourthfile.json
```

`myfourthfile.json`의 콘텐츠:

```
{
  "MedicalTranscriptionJobName": "speaker-id-conversation-medical-
transcription-job",
  "LanguageCode": "language-code",
  "Specialty": "PRIMARYCARE",
  "Type": "CONVERSATION",
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
  },
  "Settings": {
    "ShowSpeakerLabels": true,
    "MaxSpeakerLabels": 2
  }
}
```

출력:

```
{
  "MedicalTranscriptionJob": {
    "MedicalTranscriptionJobName": "speaker-id-conversation-medical-
transcription-job",
    "TranscriptionJobStatus": "IN_PROGRESS",
    "LanguageCode": "language-code",
    "Media": {
```

```

        "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
    },
    "StartTime": "2020-09-21T18:43:37.265000+00:00",
    "CreationTime": "2020-09-21T18:43:37.157000+00:00",
    "Settings": {
        "ShowSpeakerLabels": true,
        "MaxSpeakerLabels": 2
    },
    "Specialty": "PRIMARYCARE",
    "Type": "CONVERSATION"
}
}

```

자세한 내용은 Amazon Transcribe 개발자 안내서의 [화자 식별](#)을 참조하세요.

예 5: 최대 두 개의 대체 트랜스크립션을 사용하여 오디오 파일로 저장된 의료 대화를 트랜스크립션하는 방법

다음 `start-medical-transcription-job` 예시에서는 단일 오디오 파일에서 최대 두 개의 대체 트랜스크립션을 생성합니다. 모든 트랜스크립션에는 신뢰도가 있습니다. 기본적으로 Amazon Transcribe은 신뢰도가 가장 높은 트랜스크립션을 반환합니다. Amazon Transcribe에서 신뢰도가 낮은 추가 트랜스크립션을 반환하도록 지정할 수 있습니다. `OutputBucketName` 파라미터에는 트랜스크립션 출력의 위치를 지정합니다.

```

aws transcribe start-medical-transcription-job \
  --cli-input-json file://myfifthfile.json

```

`myfifthfile.json`의 콘텐츠:

```

{
  "MedicalTranscriptionJobName": "alternatives-conversation-medical-
transcription-job",
  "LanguageCode": "language-code",
  "Specialty": "PRIMARYCARE",
  "Type": "CONVERSATION",
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
  },
  "Settings": {
    "ShowAlternatives": true,
    "MaxAlternatives": 2
  }
}

```

```
}
}
```

출력:

```
{
  "MedicalTranscriptionJob": {
    "MedicalTranscriptionJobName": "alternatives-conversation-medical-
transcription-job",
    "TranscriptionJobStatus": "IN_PROGRESS",
    "LanguageCode": "language-code",
    "Media": {
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
    },
    "StartTime": "2020-09-21T19:09:18.199000+00:00",
    "CreationTime": "2020-09-21T19:09:18.171000+00:00",
    "Settings": {
      "ShowAlternatives": true,
      "MaxAlternatives": 2
    },
    "Specialty": "PRIMARYCARE",
    "Type": "CONVERSATION"
  }
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 [대체 트랜스크립션](#)을 참조하세요.

예 6: 최대 두 개의 대체 트랜스크립션을 사용하여 의료 구술 오디오 파일을 트랜스크립션하는 방법

다음 `start-medical-transcription-job` 예시에서는 오디오 파일을 트랜스크립션하고 어휘 필터를 사용하여 원하지 않는 단어를 마스킹합니다. 파라미터에 트랜스크립션 출력 위치를 지정합니다. `OutputBucketName`

```
aws transcribe start-medical-transcription-job \
  --cli-input-json file://mysixthfile.json
```

`mysixthfile.json`의 콘텐츠:

```
{
  "MedicalTranscriptionJobName": "alternatives-conversation-medical-
transcription-job",
```



```

"LanguageCode": "language-code",
"Specialty": "PRIMARYCARE",
"Type": "DICTATION",
"OutputBucketName": "DOC-EXAMPLE-BUCKET",
"Media": {
  "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
},
"Settings": {
  "ShowAlternatives": true,
  "MaxAlternatives": 2
}
}

```

출력:

```

{
  "MedicalTranscriptionJob": {
    "MedicalTranscriptionJobName": "alternatives-dictation-medical-
transcription-job",
    "TranscriptionJobStatus": "IN_PROGRESS",
    "LanguageCode": "language-code",
    "Media": {
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.extension"
    },
    "StartTime": "2020-09-21T21:01:14.592000+00:00",
    "CreationTime": "2020-09-21T21:01:14.569000+00:00",
    "Settings": {
      "ShowAlternatives": true,
      "MaxAlternatives": 2
    },
    "Specialty": "PRIMARYCARE",
    "Type": "DICTATION"
  }
}

```

자세한 내용은 Amazon Transcribe 개발자 안내서의 [대체 트랜스크립션](#)을 참조하세요.

예 7: 사용자 지정 어휘로 정확도를 높여 의료 구술 오디오 파일을 트랜스크립션하는 방법

다음 `start-medical-transcription-job` 예시에서는 오디오 파일을 트랜스크립션하고 이전에 생성한 의료 사용자 지정 어휘를 사용하여 트랜스크립션 정확도를 높입니다. `OutputBucketName` 파라미터에는 트랜스크립션 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": "DOC-EXAMPLE-BUCKET",  
  "Media": {  
    "MediaFileUri": "s3://DOC-EXAMPLE-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://DOC-EXAMPLE-BUCKET/your-audio-file.extension"  
    },  
    "StartTime": "2020-09-21T21:17:27.045000+00:00",  
    "CreationTime": "2020-09-21T21:17:27.016000+00:00",  
    "Settings": {  
      "VocabularyName": "cli-medical-vocab-1"  
    },  
    "Specialty": "PRIMARYCARE",  
    "Type": "DICTATION"  
  }  
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 [의료 사용자 지정 어휘](#)를 참조하세요.

- API 세부 정보는 AWS CLI 명령 [StartMedicalTranscriptionJob](#) 참조를 참조하십시오.

## JavaScript

### JavaScript (v3) 용 SDK

#### Note

더 많은 내용이 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배우보세요.

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 트랜스크립션 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 참조하십시오.
- API 세부 정보는 AWS SDK for JavaScript API [StartMedicalTranscriptionJob](#)참조를 참조하십시오.

AWS SDK 개발자 가이드 및 코드 예제의 전체 목록은 [을 참조하십시오](#)[SDK와 함께 이 서비스 사용 AWS](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

## AWS SDK 또는 **StartStreamTranscriptionAsync** CLI와 함께 사용

다음 코드 예시에서는 StartStreamTranscriptionAsync을 사용하는 방법을 보여 줍니다.

C++

SDK for C++

### Note

더 많은 정보가 있습니다 GitHub. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배우보세요.

```
int main() {
```

```
Aws::SDKOptions options;

Aws::InitAPI(options);
{
    //TODO(User): Set to the region of your AWS account.
    const Aws::String region = Aws::Region::US_WEST_2;

    //Load a profile that has been granted AmazonTranscribeFullAccess AWS
    managed permission policy.
    Aws::Client::ClientConfiguration config;
#ifdef _WIN32
    // ATTENTION: On Windows with the AWS C++ SDK, this example only runs if
    the SDK is built
    // with the curl library.
    // For more information, see the accompanying ReadMe.
    // For more information, see "Building the SDK for Windows with curl".
    // https://docs.aws.amazon.com/sdk-for-cpp/v1/developer-guide/setup-
    windows.html
    //TODO(User): Update to the location of your .crt file.
    config.caFile = "C:/curl/bin/curl-ca-bundle.crt";
#endif
    config.region = region;

    TranscribeStreamingServiceClient client(config);
    StartStreamTranscriptionHandler handler;
    handler.SetOnErrorCallback(
        [](const Aws::Client::AWSError<TranscribeStreamingServiceErrors>
&error) {
            std::cerr << "ERROR: " + error.GetMessage() << std::endl;
        });
    //SetTranscriptEventCallback called for every 'chunk' of file
    transcribed.
    // Partial results are returned in real time.
    handler.SetTranscriptEventCallback([](const TranscriptEvent &ev) {
        for (auto &&r: ev.GetTranscript().GetResults()) {
            if (r.GetIsPartial()) {
                std::cout << "[partial] ";
            }
            else {
                std::cout << "[Final] ";
            }
            for (auto &&alt: r.GetAlternatives()) {
                std::cout << alt.GetTranscript() << std::endl;
            }
        }
    });
}
```

```

    });

    StartStreamTranscriptionRequest request;
    request.SetMediaSampleRateHertz(SAMPLE_RATE);
    request.SetLanguageCode(LanguageCode::en_US);
    request.SetMediaEncoding(
        MediaEncoding::pcm); // wav and aiff files are PCM formats.
    request.SetEventStreamHandler(handler);

    auto OnStreamReady = [](AudioStream &stream) {
        Aws::FStream file(FILE_NAME, std::ios_base::in |
std::ios_base::binary);
        if (!file.is_open()) {
            std::cerr << "Failed to open " << FILE_NAME << '\n';
        }
        std::array<char, BUFFER_SIZE> buf;
        int i = 0;
        while (file) {
            file.read(&buf[0], buf.size());

            if (!file)
                std::cout << "File: only " << file.gcount() << " could be
read"
                    << std::endl;

            Aws::Vector<unsigned char> bits{buf.begin(), buf.end()};
            AudioEvent event(std::move(bits));
            if (!stream) {
                std::cerr << "Failed to create a stream" << std::endl;
                break;
            }
            //The std::basic_istream::gcount() is used to count the
characters in the given string. It returns
//the number of characters extracted by the last read()
operation.

            if (file.gcount() > 0) {
                if (!stream.WriteAudioEvent(event)) {
                    std::cerr << "Failed to write an audio event" <<
std::endl;

                    break;
                }
            }
            else {

```

```

        break;
    }
    std::this_thread::sleep_for(std::chrono::milliseconds(
        25)); // Slow down because we are streaming from a
file.
    }
    if (!stream.WriteAudioEvent(
        AudioEvent())) {
        // Per the spec, we have to send an empty event (an event
without a payload) at the end.
        std::cerr << "Failed to send an empty frame" << std::endl;
    }
    else {
        std::cout << "Successfully sent the empty frame" <<
std::endl;
    }
    stream.flush();
    stream.Close();
};

    Aws::Utils::Threading::Semaphore signaling(0 /*initialCount*/, 1 /
*maxCount*/);
    auto OnResponseCallback = [&signaling](
        const TranscribeStreamingServiceClient * /*unused*/,
        const Model::StartStreamTranscriptionRequest & /*unused*/,
        const Model::StartStreamTranscriptionOutcome &outcome,
        const std::shared_ptr<const Aws::Client::AsyncCallerContext> & /
*unused*/) {

        if (!outcome.IsSuccess()) {
            std::cerr << "Transcribe streaming error "
                << outcome.GetError().GetMessage() << std::endl;
        }

        signaling.Release();
    };

    std::cout << "Starting..." << std::endl;
    client.StartStreamTranscriptionAsync(request, OnStreamReady,
OnResponseCallback,
                                     nullptr /*context*/);
    signaling.WaitOne(); // Prevent the application from exiting until we're
done.
    std::cout << "Done" << std::endl;

```

```

    }

    Aws::ShutdownAPI(options);

    return 0;
}

```

- API 세부 정보는 AWS SDK for C++ API [StartStreamTranscriptionAsync](#) 참조를 참조하십시오.

AWS SDK 개발자 가이드 및 코드 예제의 전체 목록은 [을 참조하십시오](#) [SDK와 함께 이 서비스 사용 AWS](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

## AWS SDK 또는 **StartTranscriptionJob** CLI와 함께 사용

다음 코드 예제는 StartTranscriptionJob의 사용 방법을 보여줍니다.

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 트랜스크립션 및 작업 데이터 가져오기](#)

## .NET

### AWS SDK for .NET

#### Note

더 많은 정보가 있습니다 GitHub. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배우보세요.

```

/// <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 [StartTranscriptionJob](#)참조를 참조하십시오.

## CLI

### AWS CLI

예 1: 오디오 파일을 트랜스크립션하는 방법

다음 start-transcription-job 예시에서는 오디오 파일을 트랜스크립션합니다.

```
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://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-  
file-name.file-extension"  
  }  
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 [시작하기 \(AWS 명령줄 인터페이스\)](#) 를 참조하십시오.

예 2: 다중 채널 오디오 파일을 트랜스크립션하는 방법

다음 start-transcription-job 예시에서는 다중 채널 오디오 파일을 트랜스크립션합니다.

```
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://DOC-EXAMPLE-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://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-
file-name.file-extension"
    },
    "StartTime": "2020-09-17T16:07:56.817000+00:00",
    "CreationTime": "2020-09-17T16:07:56.784000+00:00",
    "Settings": {
      "ChannelIdentification": true
    }
  }
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 [다중 채널 오디오 트랜스크립션](#)을 참조하세요.

예 3: 오디오 파일을 트랜스크립션하고 다른 화자를 식별하는 방법

다음 start-transcription-job 예시에서는 오디오 파일을 트랜스크립션하고 트랜스크립션 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://DOC-EXAMPLE-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://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-
file-name.file-extension"
    },
    "StartTime": "2020-09-17T16:22:59.696000+00:00",
    "CreationTime": "2020-09-17T16:22:59.676000+00:00",
    "Settings": {
      "ShowSpeakerLabels": true,
      "MaxSpeakerLabels": 2
    }
  }
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 [화자 식별](#)을 참조하세요.

예 4: 오디오 파일을 트랜스크립션하고 트랜스크립션 출력에서 원하지 않는 단어를 마스킹하는 방법

다음 start-transcription-job 예시에서는 오디오 파일을 트랜스크립션하고 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://DOC-EXAMPLE-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 [트랜스크립션 필터링](#)을 참조하세요.

예 5: 오디오 파일을 트랜스크립션하고 트랜스크립션 출력에서 원하지 않는 단어를 제거하는 방법

다음 start-transcription-job 예시에서는 오디오 파일을 트랜스크립션하고 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://DOC-EXAMPLE-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://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-file-name.file-extension"
    },
    "StartTime": "2020-09-18T16:36:18.568000+00:00",
    "CreationTime": "2020-09-18T16:36:18.547000+00:00",
    "Settings": {
      "VocabularyFilterName": "your-vocabulary-filter",
      "VocabularyFilterMethod": "remove"
    }
  }
}

```

자세한 내용은 Amazon Transcribe 개발자 안내서의 [트랜스크립션 필터링](#)을 참조하세요.

예 6: 사용자 지정 어휘로 정확도를 높여 오디오 파일을 트랜스크립션하는 방법

다음 start-transcription-job 예시에서는 오디오 파일을 트랜스크립션하고 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://DOC-EXAMPLE-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://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-
file-name.file-extension"
    },
    "StartTime": "2020-09-18T16:36:18.568000+00:00",
    "CreationTime": "2020-09-18T16:36:18.547000+00:00",
    "Settings": {
      "VocabularyName": "your-vocabulary"
    }
  }
}

```

자세한 내용은 Amazon Transcribe 개발자 안내서의 [트랜스크립션 필터링](#)을 참조하세요.

예 7: 오디오 파일의 언어를 식별하고 트랜스크립션하는 방법

다음 start-transcription-job 예시에서는 오디오 파일을 트랜스크립션하고 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://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-
file-name.file-extension"
  }
}

```

출력:

```

{
  "TranscriptionJob": {
    "TranscriptionJobName": "cli-identify-language-transcription-job",
    "TranscriptionJobStatus": "IN_PROGRESS",
    "Media": {
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-
file-name.file-extension"
    },
    "StartTime": "2020-09-18T22:27:23.970000+00:00",
    "CreationTime": "2020-09-18T22:27:23.948000+00:00",
    "IdentifyLanguage": true
  }
}

```

자세한 내용은 Amazon Transcribe 개발자 안내서의 [언어 식별](#)을 참조하세요.

예 8: 개인 식별 정보를 수정하여 오디오 파일을 트랜스크립션하는 방법

다음 start-transcription-job 예시에서는 오디오 파일을 트랜스크립션하고 트랜스크립션 출력에서 개인 식별 정보를 수정합니다.

```

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.

예 9: 개인 식별 정보(PII)를 수정한 트랜스크립트와 수정하지 않은 트랜스크립트를 생성하는 방법

다음 start-transcription-job 예시에서는 오디오 파일의 트랜스크립션 두 개를 생성합니다. 하나는 개인 식별 정보를 수정한 것이고 다른 하나는 수정하지 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.

예 10: 이전에 생성한 사용자 지정 언어 모델을 사용하여 오디오 파일을 트랜스크립션하는 방법  
다음 start-transcription-job 예시에서는 이전에 생성한 사용자 지정 언어 모델을 사용하여 오디오 파일을 트랜스크립션합니다.

```

aws transcribe start-transcription-job \
  --cli-input-json file://mytenthfile.json

```

mytenthfile.json의 콘텐츠:

```

{
  "TranscriptionJobName": "cli-clm-2-job-1",
  "LanguageCode": "language-code",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.file-extension"
  },
}

```

```

    "ModelSettings": {
      "LanguageModelName": "cli-clm-2"
    }
  }
}

```

출력:

```

{
  "TranscriptionJob": {
    "TranscriptionJobName": "cli-clm-2-job-1",
    "TranscriptionJobStatus": "IN_PROGRESS",
    "LanguageCode": "language-code",
    "Media": {
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/your-audio-file.file-extension"
    },
    "StartTime": "2020-09-28T17:56:01.835000+00:00",
    "CreationTime": "2020-09-28T17:56:01.801000+00:00",
    "ModelSettings": {
      "LanguageModelName": "cli-clm-2"
    }
  }
}

```

자세한 내용은 Amazon Transcribe 개발자 안내서의 [사용자 지정 언어 모델을 사용한 도메인별 트랜스크립션 정확도 개선](#)을 참조하세요.

- API 세부 정보는 AWS CLI 명령 [StartTranscriptionJob](#)참조를 참조하십시오.

## Java

### SDK for Java 2.x

#### Note

자세한 내용은 에서 확인할 수 GitHub 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배우보세요.

```

public class TranscribeStreamingDemoApp {
    private static final Region REGION = Region.US_EAST_1;

```

```
private static TranscribeStreamingAsyncClient client;

public static void main(String args[])
    throws URISyntaxException, ExecutionException, InterruptedException,
LineUnavailableException {

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

    CompletableFuture<Void> result =
client.startStreamTranscription(getRequest(16_000),
    new AudioStreamPublisher(getStreamFromMic()),
    getResponseHandler());

    result.get();
    client.close();
}

private static InputStream getStreamFromMic() throws LineUnavailableException
{

    // Signed PCM AudioFormat with 16kHz, 16 bit sample size, mono
    int sampleRate = 16000;
    AudioFormat format = new AudioFormat(sampleRate, 16, 1, true, false);
    DataLine.Info info = new DataLine.Info(TargetDataLine.class, format);

    if (!AudioSystem.isLineSupported(info)) {
        System.out.println("Line not supported");
        System.exit(0);
    }

    TargetDataLine line = (TargetDataLine) AudioSystem.getLine(info);
    line.open(format);
    line.start();

    InputStream audioStream = new AudioInputStream(line);
    return audioStream;
}

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

```
private static StartStreamTranscriptionRequest getRequest(Integer
mediaSampleRateHertz) {
    return StartStreamTranscriptionRequest.builder()
        .languageCode(LanguageCode.EN_US.toString())
        .mediaEncoding(MediaEncoding.PCM)
        .mediaSampleRateHertz(mediaSampleRateHertz)
        .build();
}

private static StartStreamTranscriptionResponseHandler getResponseHandler() {
    return StartStreamTranscriptionResponseHandler.builder()
        .onResponse(r -> {
            System.out.println("Received Initial response");
        })
        .onError(e -> {
            System.out.println(e.getMessage());
            StringWriter sw = new StringWriter();
            e.printStackTrace(new PrintWriter(sw));
            System.out.println("Error Occurred: " + sw.toString());
        })
        .onComplete(() -> {
            System.out.println("=== All records stream successfully
===");
        })
        .subscriber(event -> {
            List<Result> results = ((TranscriptEvent)
event).transcript().results();
            if (results.size() > 0) {
                if (!
results.get(0).alternatives().get(0).transcript().isEmpty()) {
                    System.out.println(results.get(0).alternatives().get(0).transcript());
                }
            }
        })
        .build();
}

private InputStream getStreamFromFile(String audioFileName) {
    try {
        File inputFile = new
File(getClass().getClassLoader().getResource(audioFileName).getFile());
        InputStream audioStream = new FileInputStream(inputFile);
    }
}
```

```
        return audioStream;
    } catch (FileNotFoundException e) {
        throw new RuntimeException(e);
    }
}

private static class AudioStreamPublisher implements Publisher<AudioStream> {
    private final InputStream inputStream;
    private static Subscription currentSubscription;

    private AudioStreamPublisher(InputStream inputStream) {
        this.inputStream = inputStream;
    }

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

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

public static class SubscriptionImpl implements Subscription {
    private static final int CHUNK_SIZE_IN_BYTES = 1024 * 1;
    private final Subscriber<? super AudioStream> subscriber;
    private final InputStream inputStream;
    private ExecutorService executor = Executors.newFixedThreadPool(1);
    private AtomicLong demand = new AtomicLong(0);

    SubscriptionImpl(Subscriber<? super AudioStream> s, InputStream
inputStream) {
        this.subscriber = s;
        this.inputStream = inputStream;
    }

    @Override
    public void request(long n) {
        if (n <= 0) {
```

```
        subscriber.onError(new IllegalArgumentException("Demand must be
positive"));
    }

    demand.getAndAdd(n);

    executor.submit(() -> {
        try {
            do {
                ByteBuffer audioBuffer = getNextEvent();
                if (audioBuffer.remaining() > 0) {
                    AudioEvent audioEvent =
audioEventFromBuffer(audioBuffer);
                    subscriber.onNext(audioEvent);
                } else {
                    subscriber.onComplete();
                    break;
                }
            } while (demand.decrementAndGet() > 0);
        } catch (Exception e) {
            subscriber.onError(e);
        }
    });
}

@Override
public void cancel() {
    executor.shutdown();
}

private ByteBuffer getNextEvent() {
    ByteBuffer audioBuffer = null;
    byte[] audioBytes = new byte[CHUNK_SIZE_IN_BYTES];

    int len = 0;
    try {
        len = inputStream.read(audioBytes);

        if (len <= 0) {
            audioBuffer = ByteBuffer.allocate(0);
        } else {
            audioBuffer = ByteBuffer.wrap(audioBytes, 0, len);
        }
    } catch (IOException e) {
```

```

        throw new UncheckedIOException(e);
    }

    return audioBuffer;
}

private AudioEvent audioEventFromBuffer(ByteBuffer bb) {
    return AudioEvent.builder()
        .audioChunk(SdkBytes.fromByteBuffer(bb))
        .build();
}
}
}

```

- API 세부 정보는 AWS SDK for Java 2.x API [StartTranscriptionJob](#)참조를 참조하십시오.

## JavaScript

### JavaScript (v3) 용 SDK

#### Note

더 많은 내용이 있습니다. GitHub [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배우보세요.

트랜스크립션 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 참조하십시오.
- API 세부 정보는 AWS SDK for JavaScript API [StartTranscriptionJob](#)참조를 참조하십시오.

## Python

### SDK for Python(Boto3)

#### Note

자세한 내용은 에서 확인할 수 GitHub 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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
```

- API에 대한 자세한 내용은 파이썬용 AWS SDK (Boto3) API 레퍼런스를 참조하십시오 [StartTranscriptionJob](#).

AWS SDK 개발자 가이드 및 코드 예제의 전체 목록은 을 참조하십시오. [SDK와 함께 이 서비스 사용 AWS](#) 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

## AWS SDK 또는 **UpdateVocabulary** CLI와 함께 사용

다음 코드 예제는 UpdateVocabulary의 사용 방법을 보여줍니다.

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

AWS SDK for .NET

### Note

더 많은 정보가 있습니다 GitHub. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배우보세요.

```
/// <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 [UpdateVocabulary](#) 참조를 참조하십시오.

## 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://DOC-EXAMPLE-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 [UpdateVocabulary](#)참조를 참조하십시오.

## Python

### SDK for Python(Boto3)

#### Note

자세한 내용은 에서 확인할 수 GitHub 있습니다. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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
```

- API에 대한 자세한 내용은 파이썬용 AWS SDK (Boto3) API 레퍼런스를 참조하십시오 [UpdateVocabulary](#).

AWS SDK 개발자 가이드 및 코드 예제의 전체 목록은 을 참조하십시오. [SDK와 함께 이 서비스 사용 AWS](#) 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

## SDK를 사용한 Amazon Transcribe 시나리오 AWS

다음 코드 예제는 SDK를 사용하여 Amazon Transcribe에서 일반적인 시나리오를 구현하는 방법을 보여줍니다. AWS 이러한 시나리오에서는 Amazon Transcribe 내에서 여러 함수를 호출하여 특정 태스크를 수행하는 방법을 보여줍니다. 각 시나리오에는 코드 설정 및 실행 방법에 대한 지침을 찾을 수 있는 링크가 포함되어 있습니다. GitHub

### 예제

- [SDK를 사용하여 Amazon Transcribe 사용자 지정 어휘를 만들고 수정하십시오. AWS](#)
- [SDK를 사용하여 Amazon Transcribe에서 오디오를 텍스트로 변환하고 작업 데이터를 가져올 수 있습니다. AWS](#)

## SDK를 사용하여 Amazon Transcribe 사용자 지정 어휘를 만들고 수정하십시오. AWS

다음 코드 예시는 다음과 같은 작업을 수행하는 방법을 보여줍니다.

- Amazon S3에 오디오 파일을 업로드합니다.
- Amazon Transcribe 작업을 실행하여 파일을 트랜스크립션하고 결과를 얻습니다.
- 사용자 지정 어휘를 생성하고 세부 조정하여 트랜스크립션 정확도를 향상시킵니다.
- 사용자 지정 어휘와 함께 작업을 실행하고 결과를 얻습니다.

## Python

## SDK for Python(Boto3)

 Note

더 많은 정보가 있습니다 GitHub. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배우보세요.

Lewis Carroll의 Jabberwocky 낭독이 포함된 오디오 파일을 트랜스크립션바니다. 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

```

랩퍼 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 참조하십시오.
  - [CreateVocabulary](#)
  - [DeleteTranscriptionJob](#)
  - [DeleteVocabulary](#)
  - [GetTranscriptionJob](#)
  - [GetVocabulary](#)
  - [ListVocabularies](#)
  - [StartTranscriptionJob](#)
  - [UpdateVocabulary](#)

AWS SDK 개발자 가이드 및 코드 예제의 전체 목록은 [여기](#)를 참조하십시오. [SDK와 함께 이 서비스 사용 AWS](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

## SDK를 사용하여 Amazon Transcribe에서 오디오를 텍스트로 변환하고 작업 데이터를 가져올 수 있습니다. AWS

다음 코드 예제는 다음과 같은 작업을 수행하는 방법을 보여줍니다.

- Amazon Transcribe를 통해 트랜스크립션 작업을 시작합니다.
- 작업이 완료될 때까지 기다립니다.
- 트랜스크립트가 저장되는 URI를 가져옵니다.



자세한 내용은 [Amazon Transcribe 시작하기](#)를 참조하세요.

## Java

### SDK for Java 2.x

#### Note

더 많은 정보가 있습니다 GitHub. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 찾고 설정 및 실행하는 방법을 배우보세요.

PCM 파일을 트랜스크립션합니다.

```
/**
 * 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 트랜스크립션합니다.

```

public class TranscribeStreamingDemoApp {
    private static final Region REGION = Region.US_EAST_1;
    private static TranscribeStreamingAsyncClient client;

    public static void main(String args[])
        throws URISyntaxException, ExecutionException, InterruptedException,
        LineUnavailableException {

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

```

```
        CompletableFuture<Void> result =
client.startStreamTranscription(getRequest(16_000),
                                new AudioStreamPublisher(getStreamFromMic()),
                                getResponseHandler());

        result.get();
        client.close();
    }

    private static InputStream getStreamFromMic() throws LineUnavailableException
    {

        // Signed PCM AudioFormat with 16kHz, 16 bit sample size, mono
        int sampleRate = 16000;
        AudioFormat format = new AudioFormat(sampleRate, 16, 1, true, false);
        DataLine.Info info = new DataLine.Info(TargetDataLine.class, format);

        if (!AudioSystem.isLineSupported(info)) {
            System.out.println("Line not supported");
            System.exit(0);
        }

        TargetDataLine line = (TargetDataLine) AudioSystem.getLine(info);
        line.open(format);
        line.start();

        InputStream audioStream = new AudioInputStream(line);
        return audioStream;
    }

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

    private static StartStreamTranscriptionRequest getRequest(Integer
mediaSampleRateHertz) {
        return StartStreamTranscriptionRequest.builder()
            .languageCode(LanguageCode.EN_US.toString())
            .mediaEncoding(MediaEncoding.PCM)
            .mediaSampleRateHertz(mediaSampleRateHertz)
            .build();
    }

    private static StartStreamTranscriptionResponseHandler getResponseHandler() {
```

```

return StartStreamTranscriptionResponseHandler.builder()
    .onResponse(r -> {
        System.out.println("Received Initial response");
    })
    .onError(e -> {
        System.out.println(e.getMessage());
        StringWriter sw = new StringWriter();
        e.printStackTrace(new PrintWriter(sw));
        System.out.println("Error Occurred: " + sw.toString());
    })
    .onComplete(() -> {
        System.out.println("=== All records stream successfully
===");
    })
    .subscriber(event -> {
        List<Result> results = ((TranscriptEvent)
event).transcript().results();
        if (results.size() > 0) {
            if (!
results.get(0).alternatives().get(0).transcript().isEmpty()) {

System.out.println(results.get(0).alternatives().get(0).transcript());
            }
        }
    })
    .build();
}

private InputStream getStreamFromFile(String audioFileName) {
    try {
        File inputFile = new
File(getClass().getClassLoader().getResource(audioFileName).getFile());
        InputStream audioStream = new FileInputStream(inputFile);
        return audioStream;
    } catch (FileNotFoundException e) {
        throw new RuntimeException(e);
    }
}

private static class AudioStreamPublisher implements Publisher<AudioStream> {
    private final InputStream inputStream;
    private static Subscription currentSubscription;

    private AudioStreamPublisher(InputStream inputStream) {

```

```
        this.inputStream = inputStream;
    }

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

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

public static class SubscriptionImpl implements Subscription {
    private static final int CHUNK_SIZE_IN_BYTES = 1024 * 1;
    private final Subscriber<? super AudioStream> subscriber;
    private final InputStream inputStream;
    private ExecutorService executor = Executors.newFixedThreadPool(1);
    private AtomicLong demand = new AtomicLong(0);

    SubscriptionImpl(Subscriber<? super AudioStream> s, InputStream
inputStream) {
        this.subscriber = s;
        this.inputStream = inputStream;
    }

    @Override
    public void request(long n) {
        if (n <= 0) {
            subscriber.onError(new IllegalArgumentException("Demand must be
positive"));
        }

        demand.getAndAdd(n);

        executor.submit(() -> {
            try {
                do {
                    ByteBuffer audioBuffer = getNextEvent();
                    if (audioBuffer.remaining() > 0) {
```



```
        AudioEvent audioEvent =
audioEventFromBuffer(audioBuffer);
        subscriber.onNext(audioEvent);
    } else {
        subscriber.onComplete();
        break;
    }
    } while (demand.decrementAndGet() > 0);
} catch (Exception e) {
    subscriber.onError(e);
}
});
}

@Override
public void cancel() {
    executor.shutdown();
}

private ByteBuffer getNextEvent() {
    ByteBuffer audioBuffer = null;
    byte[] audioBytes = new byte[CHUNK_SIZE_IN_BYTES];

    int len = 0;
    try {
        len = inputStream.read(audioBytes);

        if (len <= 0) {
            audioBuffer = ByteBuffer.allocate(0);
        } else {
            audioBuffer = ByteBuffer.wrap(audioBytes, 0, len);
        }
    } catch (IOException e) {
        throw new UncheckedIOException(e);
    }

    return audioBuffer;
}

private AudioEvent audioEventFromBuffer(ByteBuffer bb) {
    return AudioEvent.builder()
        .audioChunk(SdkBytes.fromByteBuffer(bb))
        .build();
}
```

```
    }
}
```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 다음 주제를 참조하십시오.
  - [GetTranscriptionJob](#)
  - [StartTranscriptionJob](#)

## Python

### SDK for Python(Boto3)

#### Note

더 많은 것이 있어요 GitHub. [AWS 코드 예제 리포지토리](#)에서 전체 예제를 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 참조하십시오.
  - [GetTranscriptionJob](#)
  - [StartTranscriptionJob](#)

AWS SDK 개발자 가이드 및 코드 예제의 전체 목록은 [여기](#)를 참조하십시오. [SDK와 함께 이 서비스 사용 AWS](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

## SDK를 사용한 Amazon AWS Transcribe의 크로스 서비스 예제

다음 샘플 애플리케이션은 AWS SDK를 사용하여 Amazon Transcribe를 다른 애플리케이션과 결합합니다. AWS 서비스각 예제에는 애플리케이션 GitHub 설정 및 실행 방법에 대한 지침을 찾을 수 있는 링크가 포함되어 있습니다.

### 예제

- [Amazon Transcribe 앱 구축](#)
- [Amazon Transcribe 스트리밍 앱 구축](#)
- [SDK를 사용하여 텍스트를 음성으로 변환하고 다시 텍스트로 변환합니다. AWS](#)

## Amazon Transcribe 앱 구축

다음 코드 예제에서는 Amazon Transcribe를 사용하여 브라우저에서 음성 녹음을 텍스트로 기록하고 표시하는 방법을 보여줍니다.

### JavaScript

#### JavaScript (v3) 용 SDK

Amazon Transcribe를 사용하여 브라우저에서 음성 녹음을 텍스트로 기록하고 표시하는 앱을 만듭니다. 이 앱은 두 개의 Amazon Simple Storage Service (Amazon S3) 버킷을 사용합니다. 하나는 애플리케이션 코드를 호스팅하고 다른 하나는 트랜스크립션을 저장합니다. 이 앱은 Amazon Cognito 사용자 풀을 사용하여 사용자를 인증합니다. 인증된 사용자는 필요한 서비스에 액세스할 수 있는 AWS Identity and Access Management (IAM) 권한을 가집니다. AWS

전체 소스 코드와 설정 및 실행 방법에 대한 지침은 전체 예제를 참조하십시오. [GitHub](#)

이 예시는 [AWS SDK for JavaScript v3 개발자 안내서](#)에서도 확인할 수 있습니다.

이 예시에서 사용되는 서비스

- Amazon Cognito 자격 증명
- Amazon S3
- Amazon Transcribe

AWS SDK 개발자 가이드 및 코드 예제의 전체 목록은 [여기](#)를 참조하십시오. [SDK와 함께 이 서비스 사용 AWS](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

## Amazon Transcribe 스트리밍 앱 구축

다음 코드 예제에서는 라이브 오디오를 실시간으로 기록, 변환 및 번역하고 결과를 이메일로 보내는 앱을 구축하는 방법을 보여줍니다.

### JavaScript

#### JavaScript (v3) 용 SDK

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 의 전체 예제를 참조하십시오. [GitHub](#)

이 예시에서 사용되는 서비스

- Amazon Comprehend
- Amazon SES
- Amazon Transcribe
- Amazon Translate

AWS SDK 개발자 가이드 및 코드 예제의 전체 목록은 을 참조하십시오 [SDK와 함께 이 서비스 사용 AWS](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

## SDK를 사용하여 텍스트를 음성으로 변환하고 다시 텍스트로 변환합니다. AWS

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 의 전체 예제를 참조하십시오 [GitHub](#).

이 예시에서 사용되는 서비스

- Amazon Polly
- Amazon S3
- Amazon Transcribe

AWS SDK 개발자 가이드 및 코드 예제의 전체 목록은 [을 참조하십시오](#)[SDK와 함께 이 서비스 사용 AWS](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

# Amazon Transcribe의 보안

AWS에서 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 귀하의 공동 책임입니다. [공동 책임 모델](#)은 이 사항을 클라우드의 보안 및 클라우드 내 보안으로 설명합니다.

- 클라우드의 보안: AWS는 AWS 서비스를 실행하는 인프라를 보호해야 합니다. AWS 클라우드, AWS 또한 안전하게 사용할 수 있는 서비스를 제공합니다. 타사 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 데이터의 민감도, 회사 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 Transcribe를 구성하는 방법을 보여줍니다. 또한 다른 AWS 서비스를 사용하여 Amazon Transcribe 리소스를 모니터링하고 보호하는 방법에 대해 알아봅니다.

## 주제

- [Identity 및 Access Management에 대한 Amazon Transcribe](#)
- [Amazon Transcribe의 데이터 보호](#)
- [모니터링 Amazon Transcribe](#)
- [에 대한 규정 준수 검증 Amazon Transcribe](#)
- [Amazon Transcribe의 복원성](#)
- [Amazon Transcribe의 인프라 보안](#)
- [Amazon Transcribe에서 취약점 분석 및 관리](#)
- [Amazon Transcribe의 보안 모범 사례](#)

## Identity 및 Access Management에 대한 Amazon Transcribe

AWS Identity and Access Management (IAM)은 관리자가 리소스에 대한 액세스를 안전하게 제어할 수 있도록 AWS 서비스 있도록 도와줍니다. IAM 관리자는 리소스를 사용할 수 있는 인증 (로그인) 및 권

한 부여 (권한 보유) 를 받을 수 있는 사용자를 제어합니다. Amazon Transcribe IAM은 추가 AWS 서비스 비용 없이 사용할 수 있습니다.

## 주제

- [고객](#)
- [ID를 통한 인증](#)
- [정책을 사용한 액세스 관리](#)
- [IAM의 Amazon Transcribe 작동 방식](#)
- [교차 서비스 혼동된 대리자 예방](#)
- [Amazon Transcribe 자격 증명 기반 정책 예제](#)
- [Amazon Transcribe ID 및 액세스 문제 해결](#)

## 고객

사용하는 방식 AWS Identity and Access Management (IAM) 은 수행하는 작업에 따라 다릅니다. Amazon Transcribe

서비스 사용자 - Amazon Transcribe 서비스를 사용하여 작업을 수행하는 경우 관리자가 필요한 자격 증명과 권한을 제공합니다. 더 많은 Amazon Transcribe 기능을 사용하여 작업을 수행함에 따라 추가 권한이 필요할 수 있습니다. 액세스 권한 관리 방식을 이해하면 적절한 권한을 관리자에게 요청할 수 있습니다. Amazon Transcribe의 기능에 액세스할 수 없는 경우 [Amazon Transcribe ID 및 액세스 문제 해결](#)을 참조하세요.

서비스 관리자 — 회사에서 Amazon Transcribe 리소스를 담당하는 경우 전체 액세스 권한이 있을 수 Amazon Transcribe 있습니다. 서비스 사용자가 액세스해야 하는 Amazon Transcribe 기능과 리소스를 결정하는 것은 여러분의 몫입니다. 그런 다음, IAM 관리자에게 요청을 제출하여 서비스 사용자의 권한을 변경해야 합니다. 이 페이지의 정보를 검토하여 IAM의 기본 개념을 이해하십시오. 회사에서 IAM을 어떻게 사용할 수 있는지 자세히 Amazon Transcribe알아보려면 을 참조하십시오 [IAM의 Amazon Transcribe 작동 방식](#).

IAM 관리자 - IAM 관리자라면 Amazon Transcribe에 대한 액세스 권한 관리 정책 작성 방법을 자세히 알고 싶을 것입니다. IAM에서 사용할 수 있는 Amazon Transcribe ID 기반 정책의 예를 보려면 을 참조하십시오. [Amazon Transcribe 자격 증명 기반 정책 예제](#)



## ID를 통한 인증

인증은 자격 증명 자격 증명을 AWS 사용하여 로그인하는 방법입니다. IAM 사용자로 인증 (로그인 AWS) 하거나 IAM 역할을 맡아 인증 (로그인) 해야 합니다. AWS 계정 루트 사용자

ID 소스를 통해 제공된 자격 증명을 사용하여 페더레이션 ID로 로그인할 수 있습니다. AWS IAM Identity Center (IAM ID 센터) 사용자, 회사의 싱글 사인온 인증, Google 또는 Facebook 자격 증명이 페더레이션 ID의 예입니다. 페더레이션 ID로 로그인할 때 관리자가 이전에 IAM 역할을 사용하여 ID 페더레이션을 설정했습니다. 페더레이션을 사용하여 액세스하는 경우 AWS 간접적으로 역할을 맡게 됩니다.

사용자 유형에 따라 AWS Management Console 또는 AWS 액세스 포털에 로그인할 수 있습니다. 로그인에 대한 자세한 내용은 AWS 로그인 사용 설명서의 [내 로그인 방법을](#) 참조하십시오. AWS 계정

AWS 프로그래밍 방식으로 액세스하는 경우 자격 증명을 사용하여 요청에 암호화 방식으로 서명할 수 있는 소프트웨어 개발 키트 (SDK) 와 명령줄 인터페이스 (CLI) 를 AWS 제공합니다. AWS 도구를 사용하지 않는 경우 요청에 직접 서명해야 합니다. 권장 방법을 사용하여 직접 요청에 서명하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 AWS [API 요청 서명을](#) 참조하십시오.

사용하는 인증 방법에 상관없이 추가 보안 정보를 제공해야 할 수도 있습니다. 예를 들어, AWS 계정의 보안을 강화하기 위해 다단계 인증 (MFA) 을 사용할 것을 권장합니다. 자세한 내용은 AWS IAM Identity Center 사용 설명서의 [다중 인증](#) 및 IAM 사용 설명서의 [AWS에서 다중 인증\(MFA\) 사용](#)을 참조하십시오.

### AWS 계정 루트 사용자

계정을 AWS 계정만들 때는 먼저 계정의 모든 AWS 서비스 리소스에 대한 완전한 액세스 권한을 가진 하나의 로그인 ID로 시작합니다. 이 ID를 AWS 계정 루트 사용자라고 하며, 계정을 만들 때 사용한 이메일 주소와 비밀번호로 로그인하여 액세스할 수 있습니다. 일상적인 태스크에 루트 사용자를 사용하지 않을 것을 강력히 권장합니다. 루트 사용자 보안 인증 정보를 보호하고 루트 사용자만 수행할 수 있는 태스크를 수행하는 데 사용하세요. 루트 사용자로 로그인해야 하는 전체 작업 목록은 IAM 사용 설명서의 [루트 사용자 보안 인증이 필요한 작업](#)을 참조하십시오.

### 페더레이션 자격 증명

가장 좋은 방법은 관리자 액세스가 필요한 사용자를 비롯한 수동 AWS 서비스 사용자가 ID 공급자와의 페더레이션을 사용하여 임시 자격 증명을 사용하여 액세스하도록 하는 것입니다.

페더레이션 ID는 기업 사용자 디렉토리, 웹 ID 공급자, Identity Center 디렉터리의 사용자 또는 ID 소스를 통해 제공된 자격 증명을 사용하여 액세스하는 AWS 서비스 모든 사용자를 말합니다. AWS

Directory Service 페더레이션 ID에 AWS 계정 액세스하면 이들이 역할을 맡고 역할은 임시 자격 증명을 제공합니다.

중앙 집중식 액세스 관리를 위해 AWS IAM Identity Center(을)를 사용하는 것이 좋습니다. IAM Identity Center에서 사용자 및 그룹을 생성하거나 자체 ID 소스의 사용자 및 그룹 집합에 연결하고 동기화하여 모든 사용자 및 애플리케이션에서 사용할 수 있습니다. AWS 계정 IAM Identity Center에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서에서 [IAM Identity Center란 무엇입니까?](#)를 참조하십시오.

## IAM 사용자 및 그룹

[IAM 사용자는 단일 사용자](#) 또는 애플리케이션에 대한 특정 권한을 AWS 계정 가진 사용자 내 자격 증명입니다. 가능하면 암호 및 액세스 키와 같은 장기 보안 인증이 있는 IAM 사용자를 생성하는 대신 임시 보안 인증을 사용하는 것이 좋습니다. 하지만 IAM 사용자의 장기 보안 인증이 필요한 특정 사용 사례가 있는 경우, 액세스 키를 교체하는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 [장기 보안 인증이 필요한 사용 사례의 경우 정기적으로 액세스 키 교체](#)를 참조하십시오.

[IAM 그룹](#)은 IAM 사용자 컬렉션을 지정하는 자격 증명입니다. 사용자는 그룹으로 로그인할 수 없습니다. 그룹을 사용하여 여러 사용자의 권한을 한 번에 지정할 수 있습니다. 그룹을 사용하면 대규모 사용자 집합의 권한을 더 쉽게 관리할 수 있습니다. 예를 들어, IAMAdmins라는 그룹이 있고 이 그룹에 IAM 리소스를 관리할 권한을 부여할 수 있습니다.

사용자는 역할과 다릅니다. 사용자는 한 사람 또는 애플리케이션과 고유하게 연결되지만, 역할은 해당 역할이 필요한 사람이라면 누구나 수입할 수 있습니다. 사용자는 영구적인 장기 보안 인증 정보를 가지고 있지만, 역할은 임시 보안 인증만 제공합니다. 자세한 내용은 IAM 사용 설명서의 [IAM 사용자를 만들어야 하는 경우\(역할이 아님\)](#)를 참조하십시오.

## IAM 역할

[IAM 역할](#)은 특정 권한을 가진 사용자 AWS 계정 내의 자격 증명입니다. IAM 사용자와 유사하지만, 특정 개인과 연결되지 않습니다. 역할을 AWS Management Console [전환하여](#) 에서 일시적으로 IAM 역할을 맡을 수 있습니다. AWS CLI 또는 AWS API 작업을 호출하거나 사용자 지정 URL을 사용하여 역할을 수입할 수 있습니다. 역할 사용 방법에 대한 자세한 내용은 IAM 사용 설명서의 [IAM 역할 사용](#)을 참조하십시오.

임시 보안 인증이 있는 IAM 역할은 다음과 같은 상황에서 유용합니다.

- 페더레이션 사용자 액세스 - 페더레이션 ID에 권한을 부여하려면 역할을 생성하고 해당 역할의 권한을 정의합니다. 페더레이션 ID가 인증되면 역할이 연결되고 역할에 정의된 권한이 부여됩니다. 페더레이션 역할에 대한 자세한 내용은 IAM 사용 설명서의 [서드 파티 ID 공급자의 역할 생성](#) 단원을 참조

하십시오. IAM Identity Center를 사용하는 경우, 권한 집합을 구성합니다. 인증 후 ID가 액세스할 수 있는 항목을 제어하기 위해 IAM Identity Center는 권한 세트를 IAM의 역할과 연관짓습니다. 권한 세트에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서의 [권한 세트](#)를 참조하십시오.

- 임시 IAM 사용자 권한 - IAM 사용자 또는 역할은 IAM 역할을 수임하여 특정 태스크에 대한 다양한 권한을 임시로 받을 수 있습니다.
- 크로스 계정 액세스 - IAM 역할을 사용하여 다른 계정의 사용자(신뢰할 수 있는 보안 주체)가 내 계정의 리소스에 액세스하도록 허용할 수 있습니다. 역할은 계정 간 액세스를 부여하는 기본적인 방법입니다. 그러나 일부 AWS 서비스 경우에는 역할을 프록시로 사용하는 대신 정책을 리소스에 직접 연결할 수 있습니다. 계정 간 액세스에 대한 역할과 리소스 기반 정책의 차이점을 알아보려면 [IAM 사용 설명서의 IAM의 교차 계정 리소스 액세스](#)를 참조하십시오.
- 서비스 간 액세스 — 일부는 다른 기능을 사용합니다. AWS 서비스 AWS 서비스 예를 들어 서비스에서 직접적 호출을 수행하면 일반적으로 해당 서비스는 Amazon EC2에서 애플리케이션을 실행하거나 Amazon S3에 객체를 저장합니다. 서비스는 직접적으로 호출하는 보안 주체의 권한을 사용하거나, 서비스 역할을 사용하거나, 또는 서비스 연결 역할을 사용하여 이 태스크를 수행할 수 있습니다.
- 순방향 액세스 세션 (FAS) — IAM 사용자 또는 역할을 사용하여 작업을 수행하는 경우 보안 AWS 주체로 간주됩니다. 일부 서비스를 사용하는 경우 다른 서비스에서 다른 작업을 시작하는 작업을 수행할 수 있습니다. FAS는 전화를 거는 주체의 권한을 다운스트림 AWS 서비스 서비스에 AWS 서비스 요청하기 위한 요청과 결합하여 사용합니다. FAS 요청은 다른 서비스 AWS 서비스 또는 리소스와 상호 작용이 필요한 요청을 서비스가 수신한 경우에만 이루어집니다. 이 경우 두 작업을 모두 수행할 수 있는 권한이 있어야 합니다. FAS 요청 시 정책 세부 정보는 [전달 액세스 세션](#)을 참조하세요.
- 서비스 역할 - 서비스 역할은 서비스가 사용자를 대신하여 태스크를 수행하기 위해 맡는 [IAM 역할](#)입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 [AWS 서비스에 대한 권한을 위임할 역할 생성](#)을 참조하십시오.
- 서비스 연결 역할 — 서비스 연결 역할은 연결된 서비스 역할의 한 유형입니다. AWS 서비스 서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은 사용자에게 AWS 계정 표시되며 해당 서비스가 소유합니다. IAM 관리자는 서비스 링크 역할의 권한을 볼 수 있지만 편집은 할 수 없습니다.
- Amazon EC2에서 실행되는 애플리케이션 — IAM 역할을 사용하여 EC2 인스턴스에서 실행되고 API 요청을 AWS CLI 하는 애플리케이션의 임시 자격 증명을 관리할 수 있습니다. AWS 이는 EC2 인스턴스 내에 액세스 키를 저장할 때 권장되는 방법입니다. EC2 인스턴스에 AWS 역할을 할당하고 모든 애플리케이션에서 사용할 수 있게 하려면 인스턴스에 연결된 인스턴스 프로파일을 생성합니다. 인스턴스 프로파일에는 역할이 포함되어 있으며 EC2 인스턴스에서 실행되는 프로그램이 임시 보안 인증을 얻을 수 있습니다. 자세한 내용은 IAM 사용 설명서의 [IAM 역할을 사용하여 Amazon EC2 인스턴스에서 실행되는 애플리케이션에 권한 부여](#)를 참조하십시오.

IAM 역할을 사용할지 또는 IAM 사용자를 사용할지를 알아보려면 [IAM 사용 설명서](#)의 IAM 역할(사용자 대신)을 생성하는 경우를 참조하십시오.

## 정책을 사용한 액세스 관리

정책을 생성하고 이를 AWS ID 또는 리소스에 AWS 연결하여 액세스를 제어할 수 있습니다. 정책은 ID 또는 리소스와 연결될 때 AWS 해당 권한을 정의하는 객체입니다. AWS 주도자 (사용자, 루트 사용자 또는 역할 세션)가 요청할 때 이러한 정책을 평가합니다. 정책에서 권한은 요청이 허용되거나 거부되는 지를 결정합니다. 대부분의 정책은 JSON 문서로 AWS 저장됩니다. JSON 정책 문서의 구조와 콘텐츠에 대한 자세한 내용은 IAM 사용 설명서의 [JSON 정책 개요](#)를 참조하십시오.

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.

기본적으로, 사용자와 역할에는 어떠한 권한도 없습니다. 사용자에게 사용자가 필요한 리소스에서 작업을 수행할 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성하면 됩니다. 그런 다음 관리자가 IAM 정책을 역할에 추가하고, 사용자가 역할을 수임할 수 있습니다.

IAM 정책은 작업을 수행하기 위해 사용하는 방법과 상관없이 작업에 대한 권한을 정의합니다. 예를 들어, iam:GetRole 작업을 허용하는 정책이 있다고 가정합니다. 해당 정책을 사용하는 사용자는 AWS Management Console, AWS CLI, 또는 AWS API에서 역할 정보를 가져올 수 있습니다.

## 보안 인증 기반 정책

ID 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 ID에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자와 역할이 어떤 리소스와 어떤 조건에서 어떤 태스크를 수행할 수 있는지를 제어합니다. ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 [IAM 정책 생성](#)을 참조하십시오.

보안 인증 기반 정책은 인라인 정책 또는 관리형 정책으로 한층 더 분류할 수 있습니다. 인라인 정책은 단일 사용자, 그룹 또는 역할에 직접 포함됩니다. 관리형 정책은 내 여러 사용자, 그룹 및 역할에 연결할 수 있는 독립형 정책입니다. AWS 계정관리형 정책에는 AWS 관리형 정책과 고객 관리형 정책이 포함됩니다. 관리형 정책 또는 인라인 정책을 선택하는 방법을 알아보려면 IAM 사용 설명서의 [관리형 정책과 인라인 정책의 선택](#)을 참조하십시오.

## 리소스 기반 정책

리소스 기반 정책은 리소스에 연결하는 JSON 정책 설명서입니다. 리소스 기반 정책의 예는 IAM 역할 신뢰 정책과 Amazon S3 버킷 정책입니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는

이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 정책이 연결된 리소스의 경우, 정책은 지정된 보안 주체가 해당 리소스와 어떤 조건에서 어떤 태스크를 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 [보안 주체를 지정](#)해야 합니다. 보안 주체에는 계정, 사용자, 역할, 연동 사용자 등이 포함될 수 있습니다. AWS 서비스

리소스 기반 정책은 해당 서비스에 있는 인라인 정책입니다. IAM의 AWS 관리형 정책은 리소스 기반 정책에 사용할 수 없습니다.

## 액세스 제어 목록(ACL)

액세스 제어 목록(ACL)은 어떤 보안 주체(계정 멤버, 사용자 또는 역할)가 리소스에 액세스할 수 있는 권한을 가지고 있는지를 제어합니다. ACLs는 JSON 정책 문서 형식을 사용하지 않지만 리소스 기반 정책과 유사합니다.

ACL을 지원하는 서비스의 예로는 아마존 S3와 아마존 VPC가 있습니다. AWS WAF ACL에 대해 자세히 알아보려면 Amazon Simple Storage Service 개발자 가이드의 [ACL\(액세스 제어 목록\) 개요](#)를 참조하십시오.

## 기타 정책 타입

AWS 일반적이지 않은 추가 정책 유형을 지원합니다. 이러한 정책 타입은 더 일반적인 정책 타입에 따라 사용자에게 부여되는 최대 권한을 설정할 수 있습니다.

- 권한 경계 - 권한 경계는 자격 증명 기반 정책에 따라 IAM 엔티티(IAM 사용자 또는 역할)에 부여할 수 있는 최대 권한을 설정하는 고급 기능입니다. 개체에 대한 권한 경계를 설정할 수 있습니다. 그 결과로 얻는 권한은 개체의 보안 인증 기반 정책과 그 권한 경계의 교집합입니다. Principal 필드에서 사용자나 역할을 지정하는 리소스 기반 정책은 권한 경계를 통해 제한되지 않습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 권한 경계에 대한 자세한 내용은 IAM 사용 설명서의 [IAM 엔티티에 대한 권한 경계](#)를 참조하십시오.
- 서비스 제어 정책 (SCP) - SCP는 조직 또는 조직 단위 (OU)에 대한 최대 권한을 지정하는 JSON 정책입니다. AWS Organizations AWS Organizations 사업체가 소유한 여러 AWS 계정 개를 그룹화하고 중앙에서 관리하는 서비스입니다. 조직에서 모든 기능을 활성화할 경우, 서비스 제어 정책 (SCP)을 임의의 또는 모든 계정에 적용할 수 있습니다. SCP는 구성원 계정의 엔티티 (각 엔티티 포함)에 대한 권한을 제한합니다. AWS 계정 루트 사용자조직 및 SCP에 대한 자세한 내용은 AWS Organizations 사용 설명서의 [SCP 작동 방식](#)을 참조하십시오.
- 세션 정책 - 세션 정책은 역할 또는 페더레이션 사용자에게 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 결과적으로 얻는 세션의 권한은 사용자 또는 역할의 보안 인증 기반 정책의 교차와 세션 정책입니다. 또한 권한을 리소스 기반 정책에서 가져올 수도 있

습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 자세한 내용은 IAM 사용 설명서의 [세션 정책](#)을 참조하십시오.

## 여러 정책 타입

여러 정책 유형이 요청에 적용되는 경우, 결과 권한은 이해하기가 더 복잡합니다. 여러 정책 유형이 관련되어 있을 때 요청을 허용할지 여부를 AWS 결정하는 방법을 알아보려면 IAM 사용 설명서의 [정책 평가 로직](#)을 참조하십시오.

## IAM의 Amazon Transcribe 작동 방식

IAM을 사용하여 액세스를 Amazon Transcribe 관리하기 전에 어떤 IAM 기능과 함께 사용할 수 있는지 알아보세요. Amazon Transcribe

IAM 함께 사용할 수 있는 기능 Amazon Transcribe

IAM 기능	Amazon Transcribe 지원
<a href="#">ID 기반 정책</a>	예
<a href="#">리소스 기반 정책</a>	아니요
<a href="#">정책 작업</a>	예
<a href="#">정책 리소스</a>	예
<a href="#">정책 조건 키(서비스별)</a>	예
<a href="#">ACLs</a>	아니요
<a href="#">ABAC(정책 내 태그)</a>	부분
<a href="#">임시 보안 인증</a>	예
<a href="#">보안 주체 권한</a>	예
<a href="#">서비스 역할</a>	예
<a href="#">서비스 연결 역할</a>	아니요

Amazon Transcribe 및 기타 AWS 서비스가 대부분의 IAM 기능과 함께 작동하는 방식을 자세히 알아보려면 IAM 사용 IAM설명서에서 [함께 작동하는AWS 서비스를](#) 참조하십시오.

## Amazon Transcribe에 대한 ID 기반 정책

보안 인증 기반 정책 지원 예

자격 증명 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 자격 증명에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자와 역할이 어떤 리소스와 어떤 조건에서 어떤 태스크를 수행할 수 있는지를 제어합니다. ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 [IAM 정책 생성](#)을 참조하십시오.

IAM ID 기반 정책을 사용하면 허용되거나 거부되는 작업과 리소스뿐 아니라 작업이 허용되거나 거부되는 조건을 지정할 수 있습니다. 보안 인증 기반 정책에서는 보안 주체가 연결된 사용자 또는 역할에 적용되므로 보안 주체를 지정할 수 없습니다. JSON 정책에서 사용하는 모든 요소에 대해 알아보려면 IAM 사용 설명서의 [IAM JSON 정책 요소 참조](#)를 참조하십시오.

아이덴티티 기반 정책 예시 Amazon Transcribe

Amazon Transcribe ID 기반 정책의 예를 보려면 [을](#) 참조하십시오. [Amazon Transcribe 자격 증명 기반 정책 예제](#)

## Amazon Transcribe내 리소스 기반 정책

리소스 기반 정책 지원 아니요

리소스 기반 정책은 리소스에 연결하는 JSON 정책 문서입니다. 리소스 기반 정책의 예는 IAM 역할 신뢰 정책과 Amazon S3 버킷 정책입니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 정책이 연결된 리소스의 경우, 정책은 지정된 보안 주체가 해당 리소스와 어떤 조건에서 어떤 태스크를 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 [보안 주체를 지정](#)해야 합니다. 보안 주체에는 계정, 사용자, 역할, 연동 사용자 등이 포함될 수 있습니다. AWS 서비스

교차 계정 액세스를 활성화하려는 경우, 전체 계정이나 다른 계정의 IAM 개체를 리소스 기반 정책의 보안 주체로 지정할 수 있습니다. 리소스 기반 정책에 크로스 계정 보안 주체를 추가하는 것은 트러스트 관계 설정의 절반밖에 되지 않는다는 것을 유념하십시오. 보안 주체와 리소스가 다른 AWS 계정경우

신뢰할 수 있는 계정의 IAM 관리자는 보안 주체 개체 (사용자 또는 역할) 에게 리소스에 액세스할 수 있는 권한도 부여해야 합니다. 엔터티에 ID 기반 정책을 연결하여 권한을 부여합니다. 하지만 리소스 기반 정책이 동일 계정의 보안 주체에 액세스를 부여하는 경우, 추가 자격 증명 기반 정책이 필요하지 않습니다. 자세한 내용은 IAM 사용 설명서의 [IAM의 교차 계정 리소스 액세스](#)를 참조하십시오.

## Amazon Transcribe 정책 작업

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 태스크를 설명합니다. 정책 작업은 일반적으로 관련 AWS API 작업과 이름이 같습니다. 일치하는 API 작업이 없는 권한 전용 작업 같은 몇 가지 예외도 있습니다. 정책에서 여러 작업이 필요한 몇 가지 작업도 있습니다. 이러한 추가 작업을 일컬어 종속 작업이라고 합니다.

연결된 작업을 수행할 수 있는 권한을 부여하기 위한 정책에 작업을 포함하십시오.

Amazon Transcribe 작업 목록을 보려면 서비스 권한 부여 Amazon Transcribe참조에 [정의된 작업을](#) 참조하십시오.

정책 액션은 액션 앞에 transcribe 접두사를 Amazon Transcribe 사용합니다. 단일 문에서 여러 작업을 지정하려면 다음과 같이 쉼표로 구분합니다.

```
"Action": [
  "transcribe:action1",
  "transcribe:action2"
]
```

와일드카드(\*)를 사용하여 여러 작업을 지정할 수 있습니다. 예를 들어, List라는 단어로 시작하는 모든 태스크를 지정하려면 다음 태스크를 포함합니다.

```
"Action": "transcribe:List*"
```

Amazon Transcribe ID 기반 정책의 예를 보려면 을 참조하십시오. [Amazon Transcribe 자격 증명 기반 정책 예제](#)



## Amazon Transcribe 정책 리소스

### 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 개체를 지정합니다. 문장에는 Resource 또는 NotResource 요소가 반드시 추가되어야 합니다. 모범 사례에 따라 [Amazon 리소스 이름\(ARN\)](#)을 사용하여 리소스를 지정합니다. 리소스 수준 권한이라고 하는 특정 리소스 유형을 지원하는 작업에 대해 이 태스크를 수행할 수 있습니다.

작업 나열과 같이 리소스 수준 권한을 지원하지 않는 작업의 경우, 와일드카드(\*)를 사용하여 해당 문이 모든 리소스에 적용됨을 나타냅니다.

```
"Resource": "*"

```

Amazon Transcribe 리소스 유형 및 해당 ARN 목록을 보려면 서비스 권한 부여 Amazon Transcribe [참조에 정의된 리소스](#)를 참조하십시오. 각 리소스의 ARN을 지정할 수 있는 작업을 알아보려면 [Amazon Transcribe가 정의한 작업](#)을 참조하십시오.

Amazon Transcribe ID 기반 정책의 예를 보려면 을 참조하십시오. [Amazon Transcribe 자격 증명 기반 정책 예제](#)

## 예에 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 요소(또는 Condition 블록)를 사용하면 정책이 발효되는 조건을 지정할 수 있습니다. Condition 요소는 옵션입니다. 같거나 작음과 같은 [조건 연산자](#)를 사용하여 정책의 조건을 요청의 값과 일치시키는 조건식을 생성할 수 있습니다.

한 문에서 여러 Condition 요소를 지정하거나 단일 Condition 요소에서 여러 키를 지정하는 경우, AWS 는 논리적 AND 태스크를 사용하여 평가합니다. 단일 조건 키에 여러 값을 지정하는 경우 는 논리

적 OR 연산을 사용하여 조건을 AWS 평가합니다. 명문의 권한을 부여하기 전에 모든 조건을 충족해야 합니다.

조건을 지정할 때 자리 표시자 변수를 사용할 수도 있습니다. 예컨대, IAM 사용자에게 IAM 사용자 이름으로 태그가 지정된 경우에만 리소스에 액세스할 수 있는 권한을 부여할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 [IAM 정책 요소: 변수 및 태그](#)를 참조하십시오.

AWS 글로벌 조건 키 및 서비스별 조건 키를 지원합니다. 모든 AWS 글로벌 조건 키를 보려면 IAM 사용 [AWS 설명서의 글로벌 조건 컨텍스트 키](#)를 참조하십시오.

Amazon Transcribe 조건 키 목록을 보려면 서비스 권한 부여 참조의 [조건 키를 참조하십시오 Amazon Transcribe](#). 조건 키를 사용할 수 있는 작업 및 리소스에 대해 알아보려면 [작업 정의 기준](#)을 참조하십시오 Amazon Transcribe.

Amazon Transcribe ID 기반 정책의 예를 보려면 을 참조하십시오. [Amazon Transcribe 자격 증명 기반 정책 예제](#)

## 내 ACL Amazon Transcribe

ACL 지원 아니요

ACL(액세스 통제 목록)은 어떤 보안 주체(계정 멤버, 사용자 또는 역할)가 리소스에 액세스할 수 있는 권한을 가지고 있는지를 제어합니다. ACLs는 JSON 정책 문서 형식을 사용하지 않지만 리소스 기반 정책과 유사합니다.

## ABAC 포함 Amazon Transcribe

ABAC(정책 내 태그) 지원 부분

ABAC(속성 기반 액세스 통제)는 속성에 근거하여 권한을 정의하는 권한 부여 전략입니다. AWS에서는 이러한 속성을 태그라고 합니다. IAM 개체 (사용자 또는 역할) 및 여러 AWS 리소스에 태그를 첨부할 수 있습니다. ABAC의 첫 번째 단계로 개체 및 리소스에 태그를 지정합니다. 그런 다음 보안 주체의 태그가 액세스하려는 리소스의 태그와 일치할 때 작업을 허용하도록 ABAC 정책을 설계합니다.

ABAC는 빠르게 성장하는 환경에서 유용하며 정책 관리가 번거로운 상황에 도움이 됩니다.

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 정보는 IAM 사용 설명서의 [ABAC란 무엇입니까?](#)를 참조하십시오. ABAC 설정 단계가 포함된 자습서를 보려면 IAM 사용 설명서의 [속성 기반 액세스 제어\(ABAC\) 사용](#)을 참조하십시오.

Amazon Transcribe 리소스 태깅에 대한 자세한 내용은 [을 참조하십시오. 리소스에 태그 지정](#) 태그 기반 액세스 제어에 대한 자세한 내용은 [태그를 사용한 AWS 리소스 액세스 제어](#)를 참조하세요.

## Amazon Transcribe에서 임시 보안 인증 정보 사용

### 임시 보안 인증 지원

### 예

임시 자격 증명을 사용하여 로그인하면 작동하지 AWS 서비스 않는 것도 있습니다. 임시 자격 증명을 사용하는 방법을 AWS 서비스 비롯한 추가 정보는 [IAM 사용 설명서의 IAM과AWS 서비스 연동되는](#) 내용을 참조하십시오.

사용자 이름과 암호를 제외한 다른 방법을 AWS Management Console 사용하여 로그인하면 임시 자격 증명을 사용하는 것입니다. 예를 들어 회사의 SSO (Single Sign-On) 링크를 AWS 사용하여 액세스 하는 경우 이 프로세스에서 자동으로 임시 자격 증명을 생성합니다. 또한 콘솔에 사용자로 로그인한 다음 역할을 전환할 때 임시 보안 인증을 자동으로 생성합니다. 역할 전환에 대한 자세한 내용은 IAM 사용 설명서의 [역할로 전환\(콘솔\)](#)을 참조하십시오.

또는 API를 사용하여 임시 자격 증명을 수동으로 생성할 수 있습니다 AWS CLI . AWS 그런 다음 해당 임시 자격 증명을 사용하여 액세스할 수 AWS있습니다. AWS 장기 액세스 키를 사용하는 대신 임시 자격 증명을 동적으로 생성할 것을 권장합니다. 자세한 정보는 [IAM의 임시 보안 자격 증명](#) 단원을 참조하세요.

## Amazon Transcribe의 서비스 간 보안 주체 권한

### 전달 액세스 세션(FAS) 지원

### 예

IAM 사용자 또는 역할을 사용하여 작업을 수행하는 AWS경우 보안 주체로 간주됩니다. 일부 서비스를 사용하는 경우 다른 서비스에서 다른 작업을 시작하는 작업을 수행할 수 있습니다. FAS는 전화를 거는 주체의 권한을 다운스트림 서비스에 AWS 서비스 요청하라는 요청과 결합하여 사용합니다. AWS 서비스 FAS 요청은 다른 서비스 AWS 서비스 또는 리소스와의 상호 작용이 필요한 요청을 서비스가 수신

한 경우에만 이루어집니다. 이 경우 두 작업을 모두 수행할 수 있는 권한이 있어야 합니다. FAS 요청 시 정책 세부 정보는 [전달 액세스 세션](#)을 참조하세요.

## 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 서비스에 대한 권한을 위임할 역할 생성](#)을 참조하십시오.

### Warning

서비스 역할의 권한을 변경하면 Amazon Transcribe 기능이 중단될 수 있습니다. 서비스 역할을 편집하기 위한 지침이 Amazon Transcribe 제공되는 경우에만 서비스 역할을 편집하십시오.

## Amazon Transcribe에 대한 서비스 연결 역할

서비스 연결 역할 지원

아니요

서비스 연결 역할은 [예](#) 연결된 서비스 역할의 한 유형입니다. AWS 서비스서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은 사용자에게 AWS 계정 표시되며 해당 서비스가 소유합니다. IAM 관리자는 서비스 링크 역할의 권한을 볼 수 있지만 편집은 할 수 없습니다.

Amazon Transcribe 서비스 연결 역할을 지원하지 않습니다.

다른 서비스의 서비스 연결 역할을 만들거나 관리하는 방법에 대한 자세한 내용은 함께 작동하는 [AWS 서비스를](#) 참조하십시오. IAM 서비스 연결 역할 열에서 Yes(이)가 포함된 서비스를 테이블에서 찾습니다. 해당 서비스에 대한 서비스 연결 역할 설명서를 보려면 Yes(네) 링크를 선택합니다.

## 교차 서비스 혼동된 대리자 예방

혼란스러운 대리인은 다른 주체에 의해 조치를 취하도록 강요받는 엔티티 (서비스 또는 계정)입니다. 이러한 유형의 사칭은 계정 간 및 서비스 간에 발생할 수 있습니다.

대리자 혼동된 대리자 방지 차원에서 AWS에서는 리소스에 대한 액세스 권한이 부여된 모든 서비스에 대한 액세스 권한이 부여된 서비스 보안 주체를 사용하여 모든 서비스에 대한 액세스 권한이 부여된 서비스 보안 주체를 사용하여 모든 서비스에 대한 액세스 권한이 부여된 서비스 보안 주체를 사용하여 모든 서비스에 대한 액세스 권한이 부여된 서비스 보안 주체를 사용하여 AWS 계정 이 섹션에서는 특정 서비스 간 혼동 문제 예방에 중점을 두고 있지만 이 항목에 대한 자세한 내용은 IAM 사용 설명서의 [혼동 보조 문제](#) 섹션에서 확인할 수 있습니다. Amazon Transcribe

리소스에 액세스할 수 있는 권한을 IAM Amazon Transcribe 제한하려면 글로벌 조건 컨텍스트 [aws:SourceArn](#) 키와 [aws:SourceAccount](#) 리소스 정책을 사용하는 것이 좋습니다.

두 전역 조건 컨텍스트 키를 모두 사용하는 경우, `aws:SourceArn`과 `aws:SourceAccount` 값과 `in`이 동일한 정책 문에서 사용할 경우 동일한 AWS 계정 ID를 사용해야 합니다. AWS 계정 ID를 `aws:SourceArn`에 포함시키십시오.

하나의 리소스만 교차 서비스 액세스와 연결되게 하려는 경우 `aws:SourceArn`을 사용합니다. 해당 리소스에 포함된 리소스를 서비스 간 AWS 계정 액세스와 연결하려면 `aws:SourceAccount`를 사용하십시오.

#### Note

혼동된 대리자 문제로부터 보호하는 가장 효과적인 방법은 리소스의 전체 ARN이 포함된 `aws:SourceArn` 글로벌 조건 컨텍스트 키를 사용하는 것입니다. 전체 ARN을 모를 경우 또는 여러 리소스를 지정하는 경우, ARN의 알 수 없는 부분에 대해 와일드카드 (\*)를 포함한 `aws:SourceArn` 글로벌 조건 컨텍스트 키를 사용합니다. 예: `arn:aws:transcribe::123456789012:*`.

혼란스러운 대리자 문제를 방지하는 방법을 보여주는 역할 수입 정책의 예는 [이 섹션](#)을 참조하십시오. [혼동된 대리자 방지 정책](#).

## Amazon Transcribe 자격 증명 기반 정책 예제

기본적으로 사용자 및 역할에는 Amazon Transcribe 리소스를 생성하거나 수정할 수 있는 권한이 없습니다. 또한 AWS Management Console, AWS Command Line Interface(AWS CLI) 또는 AWS API를 사용하여 작업을 수행할 수 없습니다. 사용자에게 사용자가 필요한 리소스에서 작업을 수행할 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성하면 됩니다. 그런 다음 관리자가 IAM 정책을 역할에 추가하고, 사용자가 역할을 맡을 수 있습니다.

이러한 예제 JSON 정책 문서를 사용하여 IAM ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 [IAM 정책 생성](#)을 참조하세요.

각 리소스 유형에 대한 ARN 형식을 포함하여 Amazon Transcribe에서 정의한 작업 및 리소스 유형에 대한 자세한 내용은 서비스 승인 참조의 [Amazon Transcribe에 사용되는 작업, 리소스 및 조건 키](#)를 참조하세요.

## 주제

- [정책 모범 사례](#)
- [AWS Management Console 사용](#)
- [IAM역할에 필요한 권한](#)
- [Amazon S3암호화 키에 필요한 권한](#)
- [사용자가 자신이 권한을 볼 수 있도록 허용](#)
- [AWS KMS암호화 컨텍스트 정책](#)
- [혼동된 대리자 방지 정책](#)
- [태그를 기준으로 트랜스크립션 작업 보기](#)

## 정책 모범 사례

ID 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. 자격 증명 기반 정책을 생성하거나 편집할 때는 다음 지침과 권장 사항을 따르세요.

- AWS 관리형 정책으로 시작하고 최소 권한을 향해 나아가기 - 사용자 및 워크로드에 권한 부여를 시작하려면 많은 일반 사용 사례에 대한 권한을 부여하는 AWS 관리형 정책을 사용합니다. 관리형 정책은 AWS 계정에서 사용할 수 있습니다. 사용 사례에 고유한 AWS 고객 관리형 정책을 정의하여 권한을 줄이는 것이 좋습니다. 자세한 정보는 IAM 사용 설명서의 [AWS managed policies](#)(관리형 정책) 또는 [AWS managed policies for job functions](#)(직무에 대한 관리형 정책)를 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 [Policies and permissions in IAM](#)(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. 특정 AWS 서비스(예: AWS CloudFormation)를 통해 사용되는 경우에

만 서비스 작업에 대한 액세스 권한을 부여할 수도 있습니다. 자세한 정보는 IAM 사용 설명서의 [IAM JSON 정책 요소: 조건](#)을 참조하세요.

- IAM Access Analyzer를 통해 IAM 정책을 검증하여 안전하고 기능적인 권한 보장 – IAM Access Analyzer에서는 IAM 정책 언어(JSON)와 모범 사례가 정책에서 준수되도록 신규 및 기존 정책을 검증합니다. IAM Access Analyzer는 100개 이상의 정책 확인 항목과 실행 가능한 권장 사항을 제공하여 안전하고 기능적인 정책을 작성하도록 돕습니다. 자세한 정보는 IAM 사용 설명서의 [IAM Access Analyzer policy validation](#)(IAM Access Analyzer 정책 검증)을 참조하세요.
- 다중 인증(MFA) 필요 – AWS 계정 계정에 IAM 사용자 또는 루트 사용자가 필요한 시나리오가 있는 경우 추가 보안을 위해 MFA를 설정합니다. API 작업을 호출할 때 MFA가 필요하다면 정책에 MFA 조건을 추가합니다. 자세한 정보는 IAM 사용 설명서의 [Configuring MFA-protected API access](#)(MFA 보호 API 액세스 구성)를 참조하세요.

IAM의 모범 사례에 대한 자세한 내용은 IAM 사용 설명서의 [IAM의 보안 모범 사례](#)를 참조하세요.

## AWS Management Console 사용

Amazon Transcribe 콘솔에 액세스하려면 최소한의 권한 집합이 있어야 합니다. 이러한 권한은 AWS 계정에서 Amazon Transcribe 리소스에 대한 세부 정보를 나열하고 볼 수 있도록 허용해야 합니다. 최소 필수 권한보다 더 제한적인 자격 증명 기반 정책을 만들면 콘솔이 해당 정책에 연결된 개체(사용자 또는 역할)에 대해 의도대로 작동하지 않습니다.

AWS CLI 또는 AWS API만 호출하는 사용자에게 최소 콘솔 권한을 허용할 필요가 없습니다. 그 대신, 수행하려는 API 작업과 일치하는 작업에만 액세스할 수 있도록 합니다.

엔티티(사용자 및 역할)가 를 사용할 수 있도록 하려면 다음AWS 관리형 정책 중 하나를 해당 엔티티에 연결하십시오. [AWS Management Console](#)

- AmazonTranscribeFullAccess: 모든Amazon Transcribe 리소스를 만들고, 읽고, 업데이트하고, 삭제하고, 실행할 수 있는 전체 권한을 부여합니다. 또한 버킷 이름이 포함된Amazon S3transcribe 버킷에 액세스할 수 있습니다.
- AmazonTranscribeReadOnlyAccess:Amazon Transcribe 리소스에 대한 읽기 전용 액세스 권한을 부여하여 트랜스크립션 작업과 사용자 지정 어휘를 가져오고 나열할 수 있습니다.

**Note**

예) IAM AWS Management Console 로그인하고 정책 이름으로 검색하여 관리형 권한 정책을 검토할 수 있습니다. "Transcribe"를 검색하면 위에 나열된 정책 (AmazonTranscribeReadOnly 및 AmazonTranscribeFullAccess) 이 모두 반환됩니다.

Amazon Transcribe API 작업에 대한 권한을 허용하는 고유의 사용자 지정 IAM 정책을 생성할 수도 있습니다. 해당 권한이 필요한 엔터티에 이러한 사용자 지정 정책을 연결할 수 있습니다.

**IAM 역할에 필요한 권한**

Amazon Transcribe 호출할 IAM 역할을 생성하는 경우 해당 역할에는 Amazon S3 버킷에 액세스할 수 있는 권한이 있어야 합니다. 해당하는 경우 를 사용하여 버킷의 콘텐츠를 KMS key 암호화해야 합니다. 정책 예시는 다음 섹션을 참조하세요.

**신뢰 정책**

트랜스크립션 요청을 할 때 사용하는 IAM 주체에는 해당 역할을 Amazon Transcribe 맡을 수 있는 신뢰 정책이 있어야 합니다. 다음 Amazon Transcribe 신뢰 정책을 사용하십시오. 참고로 통화 후 분석을 활성화한 상태에서 실시간 통화 분석을 요청하는 경우 '실시간 통화 분석을 위한 신뢰 정책'을 사용해야 합니다.

**에 대한 신뢰 정책 Amazon Transcribe**

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "transcribe.amazonaws.com"
        ]
      },
      "Action": [
        "sts:AssumeRole"
      ],
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "111122223333"
        }
      }
    }
  ]
}
```



```

    },
    "StringLike": {
      "aws:SourceArn": "arn:aws:transcribe:us-west-2:111122223333:*"
    }
  }
}
]
}

```

## 실시간 통화 분석을 위한 신뢰 정책

```

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

```

## Amazon S3입력 버킷 정책

다음 정책은 지정된 입력 버킷의 파일에 액세스할 수 있는 IAM 역할 권한을 부여합니다.

```

{
  "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 S3출력 버킷 정책

다음 정책은 지정된 출력 버킷에 파일을 쓸 수 있는IAM 역할 권한을 부여합니다.

```

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

```

## Amazon S3암호화 키에 필요한 권한

를 사용하여Amazon S3 버킷을 암호화하는 경우KMS key 정책에 다음을 포함하세요.KMS key 이렇게 해야 Amazon Transcribe가 버킷의 콘텐츠에 액세스할 수 있습니다. 액세스 허용에 대한 자세한 내용은 AWS KMS개발자 안내서의 [외부AWS 계정KMS key 액세스 허용](#)을 참조하십시오.KMS keys

```

{
  "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 자신이 권한을 볼 수 있도록 허용

이 예제는 IAM 사용자가 자신의 사용자 자격 증명에 연결된 인라인 및 관리형 정책을 볼 수 있도록 허용하는 정책을 생성하는 방법을 보여줍니다. 이 정책에는 콘솔에서 또는 AWS CLI나 AWS API를 사용하여 프로그래밍 방식으로 이 태스크를 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 KMS암호화 컨텍스트 정책

다음 정책은 IAM 역할 "ExampleRole" 에게 이 특정 작업에 대한 AWS KMS 암호 해독 및 암호화 작업을 사용할 수 있는 권한을 KMS key 부여합니다. 이 정책은 하나 이상의 암호화 컨텍스트 쌍 (이 경우 "color:indigoBlue") 이 있는 요청에만 적용됩니다. AWS KMS 암호화 컨텍스트에 대한 자세한 내용은 [참조하십시오 AWS KMS 암호화 컨텍스트](#).

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
      },
      "Action": [
        "kms:Decrypt",
        "kms:DescribeKey",
        "kms:Encrypt",
        "kms:GenerateDataKey*",
        "kms:ReEncrypt*"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:EncryptionContext:color": "indigoBlue"
        }
      }
    }
  ]
}

```

## 혼동된 대리자 방지 정책

다음은 혼란스러운 대리인 문제를 방지하는 `aws:SourceArn` 방법과 `aws:SourceAccount` 이를 사용하는 방법을 보여주는 역할 수임 정책의 예시입니다. Amazon Transcribe 혼동성 대리 예방에 대한 자세한 내용은 [참조하십시오 교차 서비스 혼동된 대리자 예방](#).

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

## 태그를 기준으로 트랜스크립션 작업 보기

자격 증명 기반 정책의 조건을 사용하여 태그를 기반으로 Amazon Transcribe 리소스에 대한 액세스를 제어할 수 있습니다. 이 예제에서는 트랜스크립션 작업을 볼 수 있도록 허용하는 정책을 생성하는 방법을 보여줍니다. 하지만 트랜스크립션 작업 태그owner가 사용자의 사용자 이름의 값과 같은 경우에만 권한이 부여됩니다. 이 정책을 사용하여 이 작업을 완료하는 데 필요한 권한도 부여합니다AWS Management Console.

이 정책을 계정의IAM 엔티티에 연결할 수 있습니다. 지정된 역할이 트랜스크립션 작업을test-role 보려고 하는 경우 트랜스크립션 작업에 태그를owner=test-role 지정하거나owner=test-role (조건 키 이름은 대소문자를 구분하지 않음) 해야 합니다. 그렇지 않으면 액세스가 거부됩니다. 자세한 내용은 IAM사용 설명서의 [IAMJSON 정책 요소: 조건](#)을 참조하십시오.

태깅에 대한 자세한 내용은Amazon Transcribe 을 참조하십시오[리소스에 태그 지정](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```

    {
      "Sid": "ListTranscriptionJobsInConsole",
      "Effect": "Allow",
      "Action": "transcribe:ListTranscriptionJobs",
      "Resource": "*"
    },
    {
      "Sid": "ViewTranscriptionJobsIfOwner",
      "Effect": "Allow",
      "Action": "transcribe:GetTranscriptionJobs",
      "Resource": "arn:aws:transcribe:*:*:transcription-job/*",
      "Condition": {
        "StringEquals": {"aws:ResourceTag/Owner": "${aws:username}"}
      }
    }
  ]
}

```

## Amazon Transcribe ID 및 액세스 문제 해결

다음 정보를 사용하여 및 AWS Identity and Access Management (IAM) 작업 시 발생할 수 있는 일반적인 문제를 Amazon Transcribe 진단하고 수정하십시오.

### 주제

- [Amazon Transcribe에서 작업을 수행할 권한이 없음](#)
- [저는 IAM을 수행할 권한이 없습니다. PassRole](#)
- [외부 사용자가 내 Amazon Transcribe 리소스에 액세스할 수 있도록 AWS 계정 허용하고 싶습니다.](#)

### Amazon Transcribe에서 작업을 수행할 권한이 없음

작업을 수행할 권한이 없다는 오류가 수신되면, 작업을 수행할 수 있도록 정책을 업데이트해야 합니다.

다음 예제 오류는 mateojacksonIAM 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 필요하면 AWS 관리자에게 문의하세요. 관리자는 로그인 자격 증명을 제공한 사람입니다.

## 저는 IAM을 수행할 권한이 없습니다. PassRole

iam:PassRole 작업을 수행할 수 있는 권한이 없다는 오류가 수신되면 Amazon Transcribe에 역할을 전달할 수 있도록 정책을 업데이트해야 합니다.

일부 AWS 서비스 서비스에서는 새 서비스 역할 또는 서비스 연결 역할을 생성하는 대신 기존 역할을 해당 서비스에 전달할 수 있습니다. 이렇게 하려면 사용자가 서비스에 역할을 전달할 수 있는 권한을 가지고 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 수 있는 권한을 가지고 있지 않습니다.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

이 경우, Mary가 iam:PassRole 작업을 수행할 수 있도록 Mary의 정책을 업데이트해야 합니다.

도움이 필요하면 관리자에게 문의하세요. AWS 관리자는 로그인 자격 증명을 제공한 사람입니다.

외부 사용자가 내 Amazon Transcribe 리소스에 액세스할 수 있도록 AWS 계정 허용하고 싶습니다.

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.

자세히 알아보려면 다음을 참조하십시오.

- 이러한 기능의 Amazon Transcribe 지원 여부를 알아보려면 [IAM의 Amazon Transcribe 작동 방식](#).
- 소유한 리소스에 대한 액세스 권한을 AWS 계정 부여하는 방법을 알아보려면 IAM 사용 [설명서에서 자신이 소유한 다른 AWS 계정 IAM 사용자에게 액세스 권한 제공](#)을 참조하십시오.
- [제3자에게 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 타사 AWS 계정 AWS 계정 소유에 대한 액세스 제공](#)을 참조하십시오.

- ID 페더레이션을 통해 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 [외부에서 인증된 사용자에게 액세스 권한 제공\(자격 증명 페더레이션\)](#)을 참조하십시오.
- 교차 계정 액세스에 대한 역할 사용과 리소스 기반 정책의 차이점을 알아보려면 [IAM 사용 설명서의 IAM의 교차 계정 리소스 액세스](#)를 참조하십시오.

## Amazon Transcribe의 데이터 보호

AWS [공동 책임 모델](#)은 Amazon Transcribe의 데이터 보호에 적용됩니다. 이 모델에서 설명하는 것처럼 AWS은(는) 모든 AWS 클라우드을(를) 실행하는 글로벌 인프라를 보호할 책임이 있습니다. 사용자는 인프라에서 호스팅되는 콘텐츠를 관리해야 합니다. 사용하는 AWS 서비스의 보안 구성과 관리 작업에 대한 책임도 사용자에게 있습니다. 데이터 프라이버시에 대한 자세한 내용은 [Data Privacy FAQ](#)(데이터 프라이버시 FAQ)를 참조하세요. 유럽의 데이터 보호에 대한 자세한 내용은 AWS 보안 블로그의 [AWS Shared Responsibility Model and GDPR](#) 블로그 게시물을 참조하세요.

데이터를 보호하려면 AWS 계정보안 인증 정보를 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 멀티 팩터 인증 설정(MFA)을 사용하세요.
- SSL/TLS를 사용하여 AWS 리소스와 통신하세요. TLS 1.2는 필수이며 TLS 1.3를 권장합니다.
- AWS CloudTrail로 API 및 사용자 활동 로깅을 설정하세요.
- 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 140-2 인증 암호화 모듈이 필요한 경우 FIPS 엔드포인트를 사용하세요. 사용 가능한 FIPS 엔드포인트에 대한 자세한 내용은 [FIPS\(Federal Information Processing Standard\) 140-2](#)를 참조하세요.

고객의 이메일 주소와 같은 기밀 정보나 중요한 정보는 태그나 Name 필드와 같은 자유 양식 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 보안 인증을 URL에 포함시켜서는 안 됩니다.



## 인터넷워크 트래픽 개인 정보 보호

Amazon Transcribe에 대한 Amazon Virtual Private Cloud(Amazon VPC) 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 암호화는 전송 중인 데이터와 저장된 데이터를 보호하는 것을 말합니다. 전송 중에 표준 TLS (전송 계층 보안) 와 함께 Amazon S3 관리형 키 또는 KMS keys 저장된 데이터를 사용하여 데이터를 보호할 수 있습니다.

### 유휴 시 암호화

Amazon Transcribe Amazon S3 버킷에 있는 트랜스크립트의 서버 측 암호화를 위해 기본 Amazon S3 키 (SSE-S3) 를 사용합니다.

작업을 사용할 때 사용자 고유의 [StartTranscriptionJob](#) KMS key 작업을 지정하여 트랜스크립션 작업의 출력을 암호화할 수 있습니다.

Amazon Transcribe은 기본 키로 암호화된 Amazon EBS 볼륨을 사용합니다.

### 전송 중 데이터 암호화

Amazon Transcribe에서는 TLS 1.2를 AWS 인증서와 함께 사용하여 전송 중인 데이터를 암호화합니다. 여기에는 스트리밍 트랜스크립션이 포함됩니다.

### 키 관리

Amazon Transcribe와 함께 KMS keys 작동하여 데이터에 대한 향상된 암호화를 제공합니다. 를 사용하면 트랜스크립션 작업을 만들 때 입력 미디어를 암호화할 수 있습니다. Amazon S3 와 AWS KMS 통합 하면 [StartTranscriptionJob](#) 요청의 출력을 암호화할 수 있습니다.

KMS key를 지정하지 않으면 트랜스크립션 작업의 출력이 기본 Amazon S3 키 (SSE-S3) 로 암호화됩니다.

에 대한 AWS KMS 자세한 내용은 [AWS Key Management Service](#) 개발자 안내서를 참조하십시오.

## 를 사용한 키 관리AWS Management Console

트랜스크립션 작업의 출력을 암호화하려면 요청을AWS 계정 하는 용도를 사용하거나 다른KMS key 용도의 a를 사용할지 선택할 수AWS 계정 있습니다.KMS key

KMS keya를 지정하지 않으면 트랜스크립션 작업의 출력이 기본Amazon S3 키 (SSE-S3) 로 암호화됩니다.

출력 암호화를 활성화하려면:

1. 출력 데이터에서 암호화를 선택합니다.

2. 현재 사용 중인 제품인지 아니면 다른 제품인지를 선택합니다AWS 계정.KMS keyAWS 계정 현재 AWS 계정 키 (Enter) 의 키를 KMS key선택합니다. 다른AWS 계정 키에서 가져온 키를 사용하는 경우 해당 키의 ARN을 입력해야 합니다. 다른AWS 계정 사람의 키를 사용하려면 호출자에게 에 대한kms:Encrypt 권한이 있어야 합니다KMS key. 자세한 내용은 [키 정책 생성을](#) 참조하십시오.

## API를 사용하여 키 키 관리

API에서 출력 암호화를 사용하려면

[StartCallAnalyticsJobStartMedicalTranscriptionJob](#), 또는

[StartTranscriptionJob](#)작업의OutputEncryptionKMSKeyId 매개 변수를KMS key 사용하여 지정해야 합니다.

현재AWS 계정 키에 있는 키를 사용하는 경우 다음 네 가지 방법 중 하나로 키를 지정할 수 있습니다.KMS key

1. KMS keyID 자체를 사용하세요. 예: 1234abcd-12ab-34cd-56ef-1234567890ab.
2. KMS keyID에는 별칭을 사용하십시오. 예: alias/ExampleAlias.
3. KMS keyID에 Amazon 리소스 이름 (Amazon) 을 선택합니다. 예:  
arn:aws:kms:region:account-ID:key/1234abcd-12ab-34cd-56ef-1234567890ab.

4. KMS key별칭에는 ARN을 사용합니다. 예: `arn:aws:kms:region:account-ID:alias/ExampleAlias`.

현재AWS 계정 키와 AWS 계정다른 위치에 있는 키를 사용하는 경우 다음 두 가지 방법 중 하나로 키를 지정할 수KMS key 있습니다.

1. KMS keyID에는 ARN을 사용하십시오. 예: `arn:aws:kms:region:account-ID:key/1234abcd-12ab-34cd-56ef-1234567890ab`.
2. KMS key별칭에는 ARN을 사용합니다. 예: `arn:aws:kms:region:account-ID:alias/ExampleAlias`.

요청을 하는 주체에는 지정된 에 대한 권한이 있어야 합니다KMS key.

## AWS KMS 암호화 컨텍스트

AWS KMS암호화 컨텍스트는 일반 텍스트, 비밀번호 키:값 쌍의 맵입니다. 이 맵은 암호화 컨텍스트 쌍이라고 하는 인증된 추가 데이터를 나타내며, 이는 데이터에 대한 추가 보안 계층을 제공합니다. Amazon Transcribe트랜스크립션 출력을 고객 지정Amazon S3 버킷으로 암호화하려면 대칭 암호화 키가 필요합니다. 자세히 알아보려면 [의 비대칭 키를](#) 참조하십시오AWS KMS.

암호화 컨텍스트 쌍을 만들 때는 민감한 정보를 포함하지 마십시오. 암호화 컨텍스트는 비밀이 아닙니다. CloudTrail 로그에 일반 텍스트로 표시되므로 암호화 작업을 식별하고 분류하는 데 사용할 수 있습니다.

암호화 컨텍스트 쌍에는 밑줄 (`_`), 대시 (`-`), 슬래시 (`/`), 콜론 (`:`) 과 같은 특수 문자가 포함될 수 있습니다.

### Tip

암호화 컨텍스트 쌍의 값을 암호화되는 데이터와 연관시키는 것이 유용할 수 있습니다. 필수는 아니지만 파일 이름, 헤더 값 또는 암호화되지 않은 데이터베이스 필드 등 암호화된 콘텐츠와 관련된 민감하지 않은 메타데이터를 사용하는 것이 좋습니다.

API에서 출력 암호화를 사용하려면 [StartTranscriptionJob](#)작업에서KMSEncryptionContext 파라미터를 설정합니다. 출력 암호화 작업에 대한 암호화 컨텍스트를 제공하려면OutputEncryptionKMSKeyId 파라미터가 대칭KMS key ID를 참조해야 합니다.

IAM정책과 함께 [AWS KMS조건 키](#)를 사용하여 암호화 [작업](#) 요청에 사용된 암호화 컨텍스트를 KMS key 기반으로 대칭 암호화에 대한 액세스를 제어할 수 있습니다. 암호화 컨텍스트 정책의 예는 을 참조 하십시오 [AWS KMS 암호화 컨텍스트 정책](#).

암호화 컨텍스트는 선택 사항이며 암호화 (암호화) 을 선택합니다. 자세한 내용은 [암호화 컨텍스트를](#) 참조하세요.

## 서비스 개선을 위한 데이터 사용 거부

기본적으로, 는 서비스를 개발하고 사용자 경험을 지속적으로 개선하기 위해 처리한 음성 입력을 Amazon Transcribe 저장하고 사용합니다. 옵트아웃 정책을 사용하여 콘텐츠를 개발 및 개선하는 Amazon Transcribe 데 사용하는 것을 AWS Organizations 거부할 수 있습니다. 옵트아웃 방법에 대한 자세한 내용은 [단원을 참조하세요](#).

## 모니터링 Amazon Transcribe

모니터링은 및 기타 AWS 솔루션의 신뢰성, 가용성 및 성능을 유지하는 데 Amazon Transcribe 있어 중요한 부분입니다. AWS 문제 발생 시 이를 확인하고 보고하고 적절한 Amazon Transcribe 경우 자동 조치를 취할 수 있는 다음과 같은 모니터링 도구를 제공합니다.

- Amazon CloudWatch 실행 중인 AWS 리소스와 애플리케이션을 AWS 실시간으로 모니터링합니다. 지표를 수집 및 추적하고, 맞춤 대시보드를 생성할 수 있으며, 지정된 지표가 지정한 임계값에 도달 하면 사용자에게 알리거나 조치를 취하도록 경보를 설정할 수 있습니다. 예를 들어 Amazon EC2 인스턴스의 CPU 사용량 또는 기타 지표를 CloudWatch 추적하고 필요할 때 새 인스턴스를 자동으로 시작할 수 있습니다.
- Amazon CloudWatch Logs Amazon EC2 인스턴스 및 기타 소스에서 로그 파일을 모니터링 CloudTrail, 저장 및 액세스할 수 있습니다. CloudWatch Logs 로그 파일의 정보를 모니터링하고 특정 임계값이 충족되면 알려줄 수 있습니다. 또한 매우 내구력 있는 스토리지에 로그 데이터를 저장할 수 있습니다.
- AWS CloudTrail 사용자가 또는 사용자를 대신하여 수행한 API 호출 AWS 계정 및 관련 이벤트를 캡처하고 지정한 Amazon S3 버킷에 로그 파일을 전송합니다. 어떤 사용자와 계정이 전화를 걸었는지 AWS, 어떤 소스 IP 주소에서 호출이 이루어졌는지, 언제 호출이 발생했는지 확인할 수 있습니다.

자세한 정보는 [Amazon CloudWatch 사용 설명서](#)를 참조하세요.

Amazon EventBridge 이벤트를 사용하여 애플리케이션 구성 요소를 서로 연결하여 확장 가능한 이벤트 기반 애플리케이션을 더 쉽게 구축할 수 있도록 하는 서버리스 서비스입니다. EventBridge 자체 애플리케이션, SaaS (Software as a Service) 애플리케이션 및 AWS 서비스에서 실시간 데이터 스트림을

제공하고 해당 데이터를 다음과 같은 대상으로 라우팅합니다. Lambda서비스에서 발생하는 이벤트를 모니터링하고 이벤트 기반 아키텍처를 구축할 수 있습니다. 자세한 정보는 [Amazon EventBridge 사용 설명서](#)를 참조하세요.

주제

- [를 통한 모니터링 Amazon TranscribeAmazon CloudWatch](#)
- [Amazon Transcribe 를 통한 모니터링 AWS CloudTrail](#)
- [Amazon EventBridge 와 함께 사용 Amazon Transcribe](#)

## 를 통한 모니터링 Amazon TranscribeAmazon CloudWatch

원시 데이터를 수집하여 읽을 수 있는 거의 실시간 지표로 처리하는 를 Amazon Transcribe 사용하여 CloudWatch모니터링할 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 정보는 CloudWatch 사용 설명서를 참조하세요. <https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/WhatIsCloudWatch.html>

다음과 같은 Amazon CloudWatch 측정항목 및 측정기준 사용 Amazon Transcribe

Amazon Transcribe 성과 모니터링에 도움이 되는 데이터인 CloudWatch 측정항목 및 측정기준을 지원합니다. 지원되는 지표 범주에는 트랜스크립션 작업과 관련된 트래픽, 오류, 데이터 전송, 지연 시간이 포함됩니다. 지원되는 지표는 AWS/Transcribe CloudWatch 네임스페이스를 통해 찾을 수 있습니다.

### Note

CloudWatch 모니터링 지표는 무료이며 서비스 할당량에 포함되지 않습니다. CloudWatch

CloudWatch 지표에 대한 자세한 내용은 지표 [사용을 Amazon CloudWatch](#) 참조하십시오.

## Amazon Transcribe 를 통한 모니터링 AWS CloudTrail

Amazon Transcribe AWS Identity and Access Management (IAM) 사용자나 역할 또는 서비스가 수행한 작업의 기록을 제공하는 AWS 서비스와 AWS CloudTrail통합됩니다. Amazon Transcribe CloudTrail 에 대한 모든 API 호출을 Amazon Transcribe캡처합니다. 여기에는 Amazon Transcribe API 에 대한 호출 AWS Management Console 및 코드 호출이 이벤트로 포함됩니다. 트레일을 생성하면 이벤트 (에 대한 CloudTrail Amazon Transcribe이벤트 포함) 를 Amazon S3 버킷에 지속적으로 전달할

수 있습니다. 추적을 생성하지 않은 경우 이벤트 기록에서 CloudTrail AWS Management Console 콘솔의 최신 이벤트를 볼 수도 있습니다. 에서 수집한 CloudTrail정보를 사용하여 각 요청을 받은 사람 Amazon Transcribe, 요청한 IP 주소, 요청한 사람, 요청 시기 및 추가 세부 정보를 볼 수 있습니다.

자세한 CloudTrail내용은 [AWS CloudTrail 사용 설명서](#)를 참조하십시오.

## Amazon Transcribe 그리고 CloudTrail

CloudTrail 계정을 만들 AWS 계정 때 활성화됩니다. 에서 활동이 Amazon Transcribe발생하면 해당 활동이 CloudTrail 이벤트 기록의 다른 이벤트와 함께 AWS 서비스 CloudTrail 이벤트에 기록됩니다. 내 사이트에서 최근 이벤트를 보고, 검색하고, 다운로드할 수 있습니다 AWS 계정. 자세한 내용은 [CloudTrail 이벤트 기록에서 이벤트 보기](#)를 참조하세요.

에 대한 Amazon Transcribe이벤트를 포함하여 내 이벤트의 진행 중인 기록을 보려면 트레일을 만드세요 AWS 계정. 트레일은 이벤트를 지정된 CloudTrail Amazon S3 버킷에 로그 파일로 전송할 수 있는 구성입니다. CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함되어 있습니다. 이벤트는 원본의 단일 요청을 나타냅니다. 여기에는 요청된 작업, 작업 날짜 및 시간, 요청 매개 변수 등에 대한 정보가 포함됩니다. CloudTrail 로그 파일은 공개 API 호출의 정렬된 스택 추적이 아니므로 특정 순서로 표시되지 않습니다.

기본적으로 에서 트레일을 만들면 트레일이 모두에 적용됩니다 AWS 리전. AWS Management Console트레일은 AWS 파티션에 AWS 리전 있는 모든 사람의 이벤트를 기록하고 지정한 Amazon S3 버킷에 로그 파일을 전달합니다. 또한 CloudTrail 로그에서 수집된 이벤트 데이터를 추가로 분석하고 이에 따라 조치를 AWS 서비스 취하도록 기타를 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 [API 참조](#)에 문서화된 모든 Amazon Transcribe 작업을 기록합니다. 예를 들어, [CreateVocabularyGetTranscriptionJob](#), 및 [StartTranscriptionJob](#)작업은 CloudTrail 로그 파일에 항목을 생성합니다.

모든 이벤트 및 로그 항목에는 요청을 생성한 사용자에게 대한 정보가 들어 있습니다. 이 정보를 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청이 루트 자격 증명으로 이루어 졌는지 IAM 사용자 자격 증명으로 이루어 졌는지 여부

- IAM 역할 또는 페더레이션 사용자에게 대한 임시 보안 자격 증명을 통해 요청한 것인지 여부
- 다른 사람이 요청했는지 여부 AWS 서비스

자세한 내용은 [CloudTrail userIdentity 요소](#)를 참조하세요.

여러 개 AWS 리전 및 여러 개의 Amazon Transcribe 로그 파일을 단일 AWS 계정 Amazon S3 버킷으로 집계할 수도 있습니다. 자세한 내용은 [여러 지역에서 CloudTrail 로그 파일 받기 및 여러 계정에서 CloudTrail 로그 파일 받기](#)를 참조하십시오.

예: Amazon Transcribe 로그 파일 항목

트레일은 이벤트를 로그 파일로 지정된 Amazon S3 버킷에 전달할 수 있는 구성입니다. CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함되어 있습니다. 이벤트는 원본의 단일 요청을 나타냅니다. 여기에는 작업 날짜 및 시간, 요청 매개 변수 등 요청된 작업에 대한 정보가 포함됩니다. CloudTrail 로그 파일은 공개 API 호출의 정렬된 스택 추적이 아니므로 특정 순서로 표시되지 않습니다.

[StartTranscriptionJob](#) 및 [GetTranscriptionJob](#) API 작업을 호출하면 다음 항목이 생성됩니다.

```
{
  "Records": [
    {
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "111122223333",
        "arn": "arn:aws:iam:us-west-2:111122223333:user/my-user-name",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "my-user-name"
      },
      "eventTime": "2022-03-07T15:03:45Z",
      "eventSource": "transcribe.amazonaws.com",
      "eventName": "StartTranscriptionJob",
      "awsRegion": "us-west-2",
      "sourceIPAddress": "127.0.0.1",
      "userAgent": "[]",
      "requestParameters": {
        "mediaFormat": "flac",
        "languageCode": "en-US",
        "transcriptionJobName": "my-first-transcription-job",
        "media": {
```

```

        "mediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-media-file.flac"
    }
},
"responseElements": {
    "transcriptionJob": {
        "transcriptionJobStatus": "IN_PROGRESS",
        "mediaFormat": "flac",
        "creationTime": "2022-03-07T15:03:44.229000-08:00",
        "transcriptionJobName": "my-first-transcription-job",
        "languageCode": "en-US",
        "media": {
            "mediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-media-file.flac"
        }
    }
},
"requestID": "47B8E8D397DCE7A6",
"eventID": "cdc4b7ed-e171-4cef-975a-ad829d4123e8",
"eventType": "AwsApiCall",
"recipientAccountId": "111122223333"
},
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "111122223333",
        "arn": "arn:aws:iam:us-west-2:111122223333:user/my-user-name",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "my-user-name"
    },
    "eventTime": "2022-03-07T15:07:11Z",
    "eventSource": "transcribe.amazonaws.com",
    "eventName": "GetTranscriptionJob",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "[]",
    "requestParameters": {
        "transcriptionJobName": "my-first-transcription-job"
    },
    "responseElements": {
        "transcriptionJob": {
            "settings": {
                },
            },
        }
    }
}

```



```

        "transcriptionJobStatus": "COMPLETED",
        "mediaFormat": "flac",
        "creationTime": "2022-03-07T15:03:44.229000-08:00",
        "transcriptionJobName": "my-first-transcription-job",
        "languageCode": "en-US",
        "media": {
            "mediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-media-file.flac"
        },
        "transcript": {
            "transcriptFileUri": "s3://DOC-EXAMPLE-BUCKET/my-first-
transcription-job.json"
        }
    },
    "requestID": "BD8798EACDD16751",
    "eventID": "607b9532-1423-41c7-b048-ec2641693c47",
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
]
}

```

## Amazon EventBridge 와 함께 사용 Amazon Transcribe

를 사용하면 다른 작업에서 이벤트를 시작하여 Amazon Transcribe 작업의 상태 변화에 대응할 수 있습니다. Amazon EventBridge AWS 서비스트랜스크립션 작업의 상태가 변경되면 EventBridge 자동으로 이벤트를 이벤트 스트림으로 보냅니다. 이벤트 스트림에서 모니터링하고자 하는 이벤트를 정의하고, 이벤트 발생 시 EventBridge 에서 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 완료되었을 때 이벤트를 AWS Lambda 함수로 라우팅하도록 규칙을 구성할 수 있습니다. [EventBridge 규칙](#)을 정의하려면 다음 섹션을 참조하세요.

이메일, [AWS Chatbot](#) 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 사용할 [AWS 사용자 알림](#) 수 있습니다. AWS 사용자 알림 집계를 지원하므로 특정 이벤트 중에 받는 알림 수를 줄일 수 있습니다.

## 규칙 정의 EventBridge

EventBridge 규칙을 정의하려면 를 사용하십시오 [AWS Management Console](#). 규칙을 정의할 때 Amazon Transcribe 를 서비스 이름으로 사용하세요. 규칙을 만드는 방법에 대한 예는 [Amazon EventBridge 규칙을](#) 참조하십시오. EventBridge

사용하기 EventBridge전에 다음 정의를 참고하십시오.

- 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. 예: AWS Lambda 또는 Amazon Simple Notification Service (Amazon SNS). 대상은 JSON 형식으로 이벤트를 수신합니다.
- 규칙 - 규칙은 EventBridge 감시하려는 수신 이벤트를 매칭하고 처리 대상으로 라우팅합니다. 규칙 이 하나의 이벤트를 여러 대상으로 라우팅하는 경우, 모든 대상은 이벤트를 병렬 방식으로 처리합니다. 규칙은 대상으로 전송된 JSON을 사용자 지정할 수 있습니다.

Amazon EventBridge 이벤트는 최선의 노력을 기울여 발생합니다. 에서 EventBridge이벤트를 만들고 관리하는 방법에 대한 자세한 내용은 Amazon EventBridge 사용 설명서의Amazon EventBridge [이벤트를](#) 참조하십시오.

다음은 트랜스크립션 작업의 상태가 또는 로 COMPLETED 변경될 때 시작되는 EventBridge 규칙의 예입니다. Amazon Transcribe FAILED

```
{
  "source": [
    "aws.transcribe"
  ],
  "detail-type": [
    "Transcribe Job State Change"
  ],
  "detail": {
    "TranscriptionJobStatus": [
      "COMPLETED",
      "FAILED"
    ]
  }
}
```

규칙에는 다음 필드가 포함됩니다.

- `source`—이벤트의 원본. 왜냐하면 이는 Amazon Transcribe 항상 그렇기 때문입니다.  
`aws.transcribe`
- `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 `FAILED`로 변경될 때 이벤트를 시작합니다.

## Amazon Transcribe 이벤트

Amazon EventBridge 여러 Amazon Transcribe 이벤트를 기록합니다.

- [트랜스크립션 작업 이벤트](#)
- [언어 식별 이벤트](#)
- [Call Analytics 이벤트](#)
- [Call Analytics 통화 후 이벤트](#)
- [어휘 이벤트](#)

이러한 이벤트는 모두 다음과 같은 공유 필드를 포함합니다.

- `version`: 이벤트 데이터의 버전. 이 값은 항상 0입니다.
- `id`: 이벤트에 EventBridge 대해 생성한 고유 식별자입니다.
- `detail-type`: 이벤트 세부 정보에 대한 식별자. 예를 들어 `Transcribe Job State Change`입니다.
- `source`: 이벤트의 원본. Amazon Transcribe 왜냐하면 항상 그렇기 `aws.transcribe` 때문입니다.
- `account`: API 호출을 생성한 계정의 AWS 계정 ID.
- `time`: 이벤트가 전달된 날짜 및 시간.
- `region`: 요청이 이루어진 위치. AWS 리전
- `resources`: API 호출에서 사용한 리소스. 왜냐하면 Amazon Transcribe이 필드는 항상 비어 있기 때문입니다.
- `detail`: 이벤트에 대한 추가 세부 정보.
  - `FailureReason`: 이 필드는 상태 또는 상태가 `FAILED`로 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

작업 상태가 에서 COMPLETED 또는 IN\_PROGRESS FAILED 로 변경되면 이벤트가 Amazon Transcribe 생성됩니다. 상태가 바뀐 작업을 식별하고 대상에서 이벤트를 시작하려면 이벤트의 TranscriptionJobName 필드를 사용합니다. Amazon Transcribe 이벤트에는 다음 정보가 포함됩니다. 트랜스크립션 작업 상태가 detail인 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 Transcribe 에서 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.

## Call Analytics 이벤트

[Call Analytics](#) 작업의 상태가 IN\_PROGRESS에서 COMPLETED 또는 FAILED로 바뀌면 Amazon Transcribe 에서 이벤트를 생성합니다. 상태가 바뀐 Call Analytics 작업을 식별하고 대상에서 이벤트를 시작하려면 이벤트의 JobName 필드를 사용합니다. Amazon Transcribe 이벤트에는 다음 정보가 포함됩니다. Call Analytics 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"),
      "AnalyticsJobDetails": { // only when you enable optional features such as
Generative Call Summarization
        "Skipped": []
      }
    }
  }
}

```

- JobName: Call Analytics 트랜스크립션 작업의 고유한 이름.
- JobStatus: Call Analytics 트랜스크립션 작업의 상태. 이는 COMPLETED 또는 FAILED일 수 있습니다.
- AnalyticsJobDetails: 건너뛰는 분석 기능에 대한 정보를 포함한 통화 분석 기록 작업의 세부 정보.

## Call Analytics 통화 후 이벤트

[통화 후 분석](#) 트랜스크립션의 상태가 IN\_PROGRESS에서 COMPLETED 또는 FAILED로 바뀌면 Amazon Transcribe 에서 이벤트를 생성합니다. 상태가 바뀐 Call Analytics 통화 후 작업을 식별하고 대 상에서 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://DOC-EXAMPLE-BUCKET/my-output-files/my-redacted-file.JSON",
      "TranscriptFileUri": "s3://DOC-EXAMPLE-BUCKET/my-output-files/my-file.JSON"
    },
    "Media": {
      "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-output-files/my-redacted-file.WAV",
      "RedactedMediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-output-files/my-redacted-file.WAV"
    }
  }
}

```

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 Call Analytics 트랜스크립션 요청에 할당된 식별 번호.
- PostCallStatus: 통화 후 Call Analytics 트랜스크립션의 상태. 이는 COMPLETED 또는 FAILED일 수 있습니다.
- Transcript: 수정된 트랜스크립트과 수정되지 않은 트랜스크립트의 URI.
- Media: 수정된 오디오 파일과 수정되지 않은 오디오 파일의 URI.

## 어휘 이벤트

[사용자 지정 어휘](#)의 상태가 에서 READY 또는 PENDING 로 변경되면 FAILED 이벤트가 Amazon Transcribe 생성됩니다. 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.

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.

## 에 대한 규정 준수 검증 Amazon Transcribe

특정 규정 준수 프로그램의 범위 내에 AWS 서비스 있는지 알아보려면 AWS 서비스 규정 준수 [프로그램의 AWS 서비스 범위별, 규정](#) 참조하여 관심 있는 규정 준수 프로그램을 선택하십시오. 일반 정보는 [AWS 규정 준수 프로그램 AWS 보증 프로그램 규정 AWS](#) 참조하십시오.

를 사용하여 AWS Artifact 타사 감사 보고서를 다운로드할 수 있습니다. 자세한 내용은 의 보고서 <https://docs.aws.amazon.com/artifact/latest/ug/downloading-documents.html> 참조하십시오 AWS Artifact.

사용 시 규정 준수 AWS 서비스 책임은 데이터의 민감도, 회사의 규정 준수 목표, 관련 법률 및 규정에 따라 결정됩니다. AWS 규정 준수에 도움이 되는 다음 리소스를 제공합니다.



- [보안 및 규정 준수 킷스타트 가이드](#) - 이 배포 가이드에서는 아키텍처 고려 사항을 설명하고 보안 및 규정 준수에 AWS 중점을 둔 기본 환경을 배포하기 위한 단계를 제공합니다.
- [Amazon Web Services의 HIPAA 보안 및 규정 준수를 위한 설계 — 이 백서에서는 기업이 HIPAA 적격 애플리케이션을 만드는 AWS 데 사용할 수 있는 방법을 설명합니다.](#)

#### Note

모든 AWS 서비스 사람이 HIPAA 자격을 갖춘 것은 아닙니다. 자세한 내용은 [HIPAA 적격 서비스 참조](#)를 참조하십시오.

- [AWS 규정 준수 리소스AWS](#) — 이 워크북 및 가이드 모음은 해당 산업 및 지역에 적용될 수 있습니다.
- [AWS 고객 규정 준수 가이드](#) — 규정 준수의 관점에서 공동 책임 모델을 이해하십시오. 이 가이드에서는 보안을 유지하기 위한 모범 사례를 AWS 서비스 요약하고 여러 프레임워크 (미국 표준 기술 연구소 (NIST), 결제 카드 산업 보안 표준 위원회 (PCI), 국제 표준화기구 (ISO) 등) 에서 보안 제어에 대한 지침을 매핑합니다.
- AWS Config 개발자 안내서의 [규칙을 사용하여 리소스 평가](#) — 이 AWS Config 서비스는 리소스 구성이 내부 관행, 업계 지침 및 규정을 얼마나 잘 준수하는지 평가합니다.
- [AWS Security Hub](#) — 이를 AWS 서비스 통해 내부 AWS보안 상태를 포괄적으로 파악할 수 있습니다. Security Hub는 보안 제어를 사용하여 AWS 리소스를 평가하고 보안 업계 표준 및 모범 사례에 대한 규정 준수를 확인합니다. 지원되는 서비스 및 제어 목록은 [Security Hub 제어 참조](#)를 참조하십시오.
- [Amazon GuardDuty](#) — 환경에 의심스럽고 악의적인 활동이 있는지 AWS 계정모니터링하여 워크로드, 컨테이너 및 데이터에 대한 잠재적 위협을 AWS 서비스 탐지합니다. GuardDuty 특정 규정 준수 프레임워크에서 요구하는 침입 탐지 요구 사항을 충족하여 PCI DSS와 같은 다양한 규정 준수 요구 사항을 해결하는 데 도움이 될 수 있습니다.
- [AWS Audit Manager](#) — 이를 AWS 서비스 통해 AWS 사용량을 지속적으로 감사하여 위험을 관리하고 규정 및 업계 표준을 준수하는 방법을 단순화할 수 있습니다.

## 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 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](#)를 참조하세요.

## Amazon Transcribe의 인프라 보안

매니지드 서비스로서 AWS 글로벌 네트워크 보안으로 Amazon Transcribe 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 호출을 사용하여 네트워크를 통해 Amazon Transcribe에 액세스합니다. 고객은 다음을 지원해야 합니다.

- 전송 계층 보안(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.

또한 요청은 액세스 키 ID 및 IAM 주체와 관련된 보안 액세스 키를 사용하여 서명해야 합니다. 또는 [AWS Security Token Service](#)(AWS STS)를 사용하여 임시 보안 자격 증명을 생성하여 요청에 서명할 수 있습니다.

## Amazon Transcribe에서 취약점 분석 및 관리

구성 및 IT 제어는 AWS와 고객 간의 공동 책임입니다. 자세한 내용은 AWS [공동 책임 모델](#)을 참조하세요.

## Amazon Transcribe 및 인터페이스 VPC 엔드포인트(AWS PrivateLink)

인터페이스 VPC 종단점을 생성하여 VPC와 Amazon Transcribe 간에 프라이빗 연결을 설정할 수 있습니다. 인터페이스 엔드포인트는 다음에 의해 구동됩니다. [AWS PrivateLink](#), 비공개로 액세스하는 데 사용할 수 있는 기술 Amazon Transcribe 인터넷 게이트웨이, NAT 디바이스, VPN 연결이 없는 API 또는 AWS Direct Connect 연결. VPC의 인스턴스는 Amazon Transcribe API와 통신하는 데 퍼블릭 IP 주소를 필요로 하지 않습니다. VPC와 Amazon Transcribe 간의 트래픽은 Amazon 네트워크를 벗어나지 않습니다.

각 인터페이스 엔드포인트는 서브넷에서 하나 이상의 [탄력적 네트워크 인터페이스](#)로 표현됩니다.

자세한 내용은 Amazon VPC 사용 설명서의 [인터페이스 VPC 엔드포인트\(AWS PrivateLink\)](#)를 참조하세요.

## Amazon Transcribe VPC 엔드포인트 고려 사항

인터페이스 VPC 엔드포인트를 설정하기 전 Amazon Transcribe, 반드시 검토하십시오 [인터페이스 엔드포인트 속성 및 제한](#)에서 Amazon VPC 사용자 가이드.

Amazon Transcribe은 VPC에서 모든 API 작업에 대한 호출 수행을 지원합니다.

## Amazon Transcribe에 대한 인터페이스 VPC 엔드포인트 생성

에 대한 VPC 엔드포인트를 생성할 수 있습니다. Amazon Transcribe를 사용한 서비스 Amazon VPC AWS Management Console 또는 AWS CLI. 자세한 내용은 을 참조하십시오. [인터페이스 엔드포인트 생성](#)에서 Amazon VPC 사용자 가이드.

에서 일괄 트랜스크립션용 Amazon Transcribe 다음 서비스 이름을 사용하여 VPC 엔드포인트를 생성하십시오.

- `com.amazonaws.## ## 2`. 고쳐 쓰다

내 스트리밍 트랜스크립트의 경우 Amazon Transcribe 다음 서비스 이름을 사용하여 VPC 엔드포인트를 생성하십시오.

- `com.amazonaws.## ## 2`. 트랜스크립트 스트리밍

엔드포인트에 프라이빗 DNS를 활성화하면 다음 주소로 API 요청을 보낼 수 있습니다. Amazon Transcribe에 대한 기본 DNS 이름 사용 AWS 리전, 예를 들면, `transcribestreaming.us-east-2.amazonaws.com`.

자세한 내용은 을 참조하십시오. [인터페이스 엔드포인트를 통한 서비스 액세스](#)에서 Amazon VPC 사용자 가이드.

## Amazon Transcribe에 대한 VPC 엔드포인트 정책 생성

의 스트리밍 서비스 또는 일괄 트랜스크립션 서비스에 대한 액세스를 제어하는 엔드포인트 정책을 VPC 엔드포인트에 연결할 수 있습니다. Amazon Transcribe. 이 정책은 다음 정보를 지정합니다.

- 태스크를 수행할 수 있는 보안 주체.
- 수행할 수 있는 작업입니다.
- 태스크를 수행할 있는 리소스.

자세한 내용은 을 참조하십시오. [VPC 엔드포인트로 서비스에 대한 액세스 제어](#)에서 Amazon VPC 사용자 가이드.

예: 다음에 대한 VPC 엔드포인트 정책 Amazon Transcribe 일괄 트랜스크립션 작업

다음은 에서 일괄 트랜스크립션을 위한 엔드포인트 정책의 예입니다. Amazon Transcribe. 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: 다음에 대한 VPC 엔드포인트 정책 Amazon Transcribe 스트리밍 트랜스크립션 액션

다음은 스트리밍 트랜스크립션에 대한 엔드포인트 정책의 예입니다. Amazon Transcribe. 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

공유되는 서브넷에서는 VPC 엔드포인트를 생성, 설명, 수정 또는 삭제할 수 없습니다. 그러나 공유하는 서브넷의 VPC 엔드포인트를 사용할 수는 있습니다. VPC 공유에 대한 자세한 내용은 [공유 VPC를 다른 계정과 공유](#)에서 Amazon Virtual Private Cloud 안내서.

## Amazon Transcribe의 보안 모범 사례

다음 모범 사례는 일반적인 지침이며 완벽한 보안 솔루션을 나타내지는 않습니다. 이러한 모범 사례는 환경에 적절하지 않거나 충분하지 않을 수 있으므로 참고용으로만 사용해 주십시오.

- 암호화 컨텍스트와 같은 데이터 AWS KMS 암호화 사용

AWS KMS 암호화 컨텍스트는 일반 텍스트, 비밀번호 키 값 쌍의 맵입니다. 이 맵은 암호화 컨텍스트 쌍이라고 하는 인증된 추가 데이터를 나타내며, 이는 데이터에 대한 추가 보안 계층을 제공합니다.

자세한 정보는 [AWS KMS 암호화 컨텍스트](#) 단원을 참조하십시오.

- 가능하면 임시 자격 증명 사용

가능하면 액세스 키와 같은 장기 자격 증명 대신 임시 보안 인증 정보를 사용하십시오. 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 액세스 키를 안전하게 교체 및 제거할 때 마지막으로 사용한 액세스 정보를 사용하는 것이 좋습니다.

자세한 내용은 [액세스 키 교체](#) 및 [보안 모범 사례](#)를 참조하십시오 IAM.

- Amazon Transcribe 액세스가 필요한 애플리케이션 및 AWS 서비스에 IAM 역할 사용

IAM 역할을 사용하여 액세스해야 하는 애플리케이션이나 서비스의 임시 자격 증명을 Amazon Transcribe 관리하세요. 역할을 사용할 때 Amazon EC2 인스턴스나 제품 (예:) 에 장기 자격 증명 (예: 암호 또는 액세스 키) 을 배포하지 않아도 됩니다. AWS IAM 역할은 애플리케이션에서 AWS 리소스를 요청할 때 사용할 수 있는 임시 권한을 제공할 수 있습니다.

자세한 내용은 [IAM 역할 및 역할에 대한 일반적인 시나리오 \(예: 사용자, 애플리케이션 및 서비스\)](#) 를 참조하세요.

- 태그 기반 액세스 제어 사용

태그를 사용하여 내 액세스를 제어할 수 있습니다AWS 계정. 입력Amazon Transcribe 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 참조하십시오.

- AWS모니터링 도구 사용

모니터링은 Amazon Transcribe와 사용자 AWS 솔루션의 안정성, 보안, 가용성 및 성능을 유지하는 중요한 역할을 합니다. 를Amazon Transcribe 사용하여 모니터링할 수CloudTrail 있습니다.

자세한 정보는 [Amazon Transcribe 를 통한 모니터링 AWS CloudTrail](#) 단원을 참조하십시오.

- 사용 설정AWS Config

AWS ConfigAWS리소스 구성을 측정, 감사 및 평가할 수 있습니다. 를 사용하여AWS Config 구성 및 AWS 리소스 간 관계의 변경 사항을 검토할 수 있습니다. 또한 자세한 리소스 구성 기록을 조사하고 내부 지침에 지정되어 있는 구성을 기준으로 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 What [IsAWS Config?](#) 를 참조하십시오.

# Amazon Transcribe 의료

Amazon TranscribeMedical은 의사가 작성한 메모, 약물 안전 모니터링, 원격 진료 예약 또는 의사-환자 대화와 같은 의료 관련 음성을 필사하려는 의료 전문가를 위해 설계된 자동 음성 인식 (ASR) 서비스입니다. Amazon Transcribe 의료 서비스는 실시간 스트리밍 (마이크 사용) 또는 업로드된 파일의 트랜스크립션 (배치) 을 통해 이용할 수 있습니다.

## Important

Amazon Transcribe의료는 전문적인 의학적 조언, 진단 또는 치료를 대체하지 않습니다. 사용 사례에 적합한 신뢰 임계값을 식별하고, 높은 정확도가 필요한 상황에서는 높은 신뢰 임계값을 사용합니다. 특정 사용 사례의 경우 적절한 교육을 받은 전문 검토자가 결과를 검토하고 검증해야 합니다. Amazon Transcribe 의료 기록은 숙련된 의료 전문가가 정확성과 건전한 의학적 판단을 검토한 후 환자 치료 시나리오에서만 사용해야 합니다.

Amazon TranscribeAWSMedical은 공동 책임 모델에 따라 운영되므로Amazon Transcribe Medical을 운영하는 인프라를 보호할 책임이 있으며 사용자는 데이터를 관리할 책임이 있습니다. 자세한 내용은 [공동 책임 모델](#)을 참조하십시오.

Amazon Transcribe의료 서비스는 미국 영어 (en-US) 로 제공됩니다.

최상의 결과를 얻으려면 PCM 16비트 인코딩과 함께 FLAC 또는 WAV와 같은 무손실 오디오 형식을 사용하십시오. Amazon Transcribe Medical16,000Hz 이상의 샘플 레이트를 지원합니다.

성적 증명서 분석을 위해 다음과 같은 다른AWS 서비스 것을 사용할 수 [Amazon Comprehend Medical](#)있습니다.

지원되는 전문 분야

스페셜티	하위 전문 분야	오디오 입력
심장학	none	스트리밍 전용
신경학	none	스트리밍 전용
종양학	none	스트리밍 전용
기본 진료	가정의학	배치, 스트림

스페셜티	하위 전문 분야	오디오 입력
기본 진료	내과학	배치, 스트림
기본 진료	산부인과학	배치, 스트림
기본 진료	소아과학	배치, 스트림
방사선학	none	스트리밍 전용
비뇨기과	none	스트리밍 전용

## 지역 가용성 및 할당량

통화 분석은 다음에서 지원됩니다AWS 리전.

리전(Region)	트랜스크립션 유형
af-south-1 (케이프타운)	배치
ap-east-1 (홍콩)	배치
ap-northeast-1(도쿄)	배치, 스트림
ap-northeast-2(서울)	배치, 스트림
ap-south-1 (뭄바이)	배치
ap-southeast-1(싱가포르)	배치
ap-southeast-2(시드니)	배치, 스트림
ca-central-1 (캐나다, 중부)	배치, 스트림
eu-central-1(프랑크푸르트)	배치, 스트림
eu-north-1	배치
eu-west-1(아일랜드)	배치, 스트림



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

참고로 [Amazon Transcribe](#) Amazon Transcribe Medical, 및 [통화 분석의 지역 지원은 다릅니다.](#) .

지원되는 각 지역의 엔드포인트를 가져오려면 AWS일반 참조의 [서비스 엔드포인트를](#) 참조하십시오.

트랜스크립션과 관련된 할당량 목록은 AWS일반 참조의 [서비스 할당량을](#) 참조하십시오. 일부 할당량은 요청 시 변경할 수 있습니다. 조정 가능 열에 '예'가 포함된 경우 증가를 요청할 수 있습니다. 이를 위해 제공된 링크를 선택합니다.

## Medical

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 [시작하기](#)를 참조하십시오 Amazon Simple Storage Service.

다음은 샘플 JSON에 입력할 최소 요청 파라미터 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 TranscribeMedical을 사용하면 대체 트랜스크립션을 생성할 수 있습니다. 자세한 정보는 [대체 트랜스크립션 생성](#)을 참조하세요.

스피커 파티셔닝을 활성화하거나 오디오의 채널을 식별할 수도 있습니다. 자세한 내용은 [스피커 분할 사용 설정 설정 설정 설정 설정 설정](#) 및 [멀티 채널 오디오 트랜스크립팅](#) 단원을 참조하세요.

## 의료 용어 및 측정값 필사

Amazon Transcribe의 의료 용어 및 측정치를 기록할 수 있습니다. Amazon Transcribe 음성 용어에 대한 의료 출력 약어. 예를 들어 “blood pressure”는 BP로 트랜스크립션됩니다. 이 페이지의 표에서 Amazon Transcribe 의료 용어 및 측정에 대해 의료에서 사용하는 규칙 목록을 찾을 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 사용되는 용어	출력에 사용되는 약어	출력 예
Centigrade	C	The patient's temperature is 37.4 C.(환자의 체온은 37.4C입니다.)
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.)

소스 오디오에서 사용되는 용어	출력에 사용되는 약어	출력 예
milliliter	mL	Patient excreted 100 mL urine. (환자는 100mL의 소변을 배설했습니다.)
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 심방세동이 있었습니다.)
beats per minute	BPM	Patient had atrial fibrillation with heart rate of 160 BPM.(환자는 160BPM의 심박수로 심방세동이 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.)

소스 오디오에서 사용되는 용어	출력에 사용되는 약어	출력 예
post op	POSTOP	Patient came for POSTOP evaluation.(환자가 POSTOP 평가를 위해 방문했습니다.)
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.)

## 숫자 표기하기

Amazon Transcribe의학은 숫자를 단어 대신 숫자로 기록합니다. 예를 들어 숫자 "천 이백 사십 이"를 말하면 1242로 트랜스크립션됩니다.

번호는 다음 규칙에 따라 기록됩니다.

규칙	설명
10보다 큰 기수를 숫자로 변환합니다.	<ul style="list-style-type: none"> <li>• "Fifty five" &gt; 55</li> <li>• "a hundred" &gt; 100</li> <li>• "One thousand and thirty one" &gt; 1031</li> <li>• "One hundred twenty-three million four hundred fifty six thousand seven hundred eight nine" &gt; 123,456,789</li> </ul>
"million" 또는 "billion" 뒤에 숫자가 오지 않는 경우 뒤에 "million" 또는 "billion"이 있는 기수를 숫자로 변환합니다.	<ul style="list-style-type: none"> <li>• "one hundred million" &gt; 100,000,000</li> <li>• "one billion" &gt; 1,000,000,000</li> <li>• "two point three million" &gt; 2,300,000</li> </ul>
10보다 큰 서수를 숫자로 변환합니다.	<ul style="list-style-type: none"> <li>• "Forty third" &gt; 43rd</li> <li>• "twenty sixth avenue" &gt; 26th avenue</li> </ul>
분수를 숫자 형식으로 변환합니다.	<ul style="list-style-type: none"> <li>• "a quarter" &gt; 1/4</li> <li>• "three sixteenths" &gt; 3/16</li> <li>• "a half" &gt; 1/2</li> <li>• "a hundredth" &gt; 1/100</li> </ul>
행에 하나 이상의 자릿수가 10 미만으로 숫자로 변환합니다.	<ul style="list-style-type: none"> <li>• "three four five" &gt; 345</li> <li>• "My phone number is four two five five five five one two one two" &gt; 4255551212</li> </ul>
소수점은 "점"으로 표시됩니다.	<ul style="list-style-type: none"> <li>• "three hundred and three dot five" &gt; 303.5</li> <li>• "three point twenty three" &gt; 3.23</li> <li>• "zero point four" &gt; 0.4</li> <li>• "point three" &gt; 0.3</li> </ul>
숫자 뒤의 "percent" 단어를 백분율 기호(%)로 변환합니다.	<ul style="list-style-type: none"> <li>• "twenty three percent" &gt; 23%</li> <li>• "twenty three point four five percent" &gt; 23.45%</li> </ul>

규칙	설명
숫자 뒤의 "dollar," "US dollar," "Australian dollar," "AUD" 또는 "USD" 단어를 숫자 앞에 달러 기호로 변환합니다.	<ul style="list-style-type: none"> <li>"one dollar and fifteen cents" &gt; \$1.15</li> <li>"twenty three USD" &gt; \$23</li> <li>"twenty three Australian dollars" &gt; \$23</li> </ul>
"pounds" 또는 "milligrams" 단어를 각각 "lbs" 또는 "mg"로 변환합니다.	<ul style="list-style-type: none"> <li>"twenty three pounds" &gt; 23 lbs</li> <li>"forty-five milligrams" &gt; 45 mg</li> </ul>
숫자 뒤의 "rupees," "Indian rupees" 또는 "INR" 단어를 숫자 앞의 루피 기호(#)로 변환합니다.	<ul style="list-style-type: none"> <li>"twenty three rupees" &gt; #23</li> <li>"fifty rupees thirty paise" &gt; #50.30</li> </ul>
시간을 숫자로 변환합니다.	<ul style="list-style-type: none"> <li>"seven a m eastern standard time" &gt; 7 a.m. eastern standard time</li> <li>"twelve thirty p m" &gt; 12:30 p.m.</li> </ul>
연도를 두 자리에서 네 자리로 표현합니다. 20세기, 21세기, 22세기에 한해 유효합니다.	<ul style="list-style-type: none"> <li>"nineteen sixty two" &gt; 1962</li> <li>"the year is twenty twelve" &gt; the year is 2012</li> <li>"twenty nineteen" &gt; 2019</li> <li>"twenty one thirty" &gt; 2130</li> </ul>
날짜를 숫자로 변환합니다.	<ul style="list-style-type: none"> <li>"May fifth twenty twelve" &gt; May 5th 2012</li> <li>"May five twenty twelve" &gt; May 5 2012</li> <li>"five May twenty twelve" &gt; 5 May 2012</li> </ul>
숫자의 범위를 "to" 단어로 구분합니다.	<ul style="list-style-type: none"> <li>"twenty three to thirty seven" &gt; 23 to 37</li> </ul>

## 의료 대화 내용 필사하기

Amazon TranscribeMedical을 사용하면 일괄 기록 작업이나 실시간 스트림을 사용하여 임상 의사와 환자 간의 의료 대화를 기록할 수 있습니다. Batch 트랜스크립션 작업을 통해 오디오 파일을 트랜스크립션할 수 있습니다. Amazon TranscribeMedical에서 가능한 가장 높은 정확도로 전사 결과를 생성하려면 필사 작업 또는 스트림에서 해당 임상 의사의 전문 의료 분야를 지정해야 합니다.

다음 의료 전문 분야의 임상 의-환자 방문 기록을 기록할 수 있습니다.

- 심장학 — 스트리밍 트랜스크립션으로만 이용 가능
- 신경학 — 스트리밍 트랜스크립션에서만 사용 가능
- 종양학 — 스트리밍 트랜스크립션으로만 사용 가능
- 1차 진료 — 다음과 같은 유형의 의료 행위를 포함합니다.
  - 가정의학
  - 내과학
  - 산부인과학
  - 소아과학
- 비뇨기과 — 스트리밍 트랜스크립션으로만 이용 가능

의료 맞춤 어휘를 사용하여 필사 정확도를 높일 수 있습니다. 의료 사용자 지정 어휘의 작동 방식에 대한 자세한 내용은 [을 참조하십시오](#) [의료 맞춤형 어휘를 통한 필사 정확도 향상](#).

기본적으로 Amazon Transcribe Medical은 신뢰도가 가장 높은 트랜스크립션을 반환합니다. 대체 트랜스크립션을 반환하도록 구성하려면 [을 참조하십시오](#) [대체 트랜스크립션 생성](#).

수치와 의료 측정값이 전사 출력에 표시되는 방식에 대한 자세한 내용은 [숫자 표기하기](#) 및 [을 참조하십시오](#) [의료 용어 및 측정값 필사](#).

## 주제

- [의료 대화의 오디오 파일 녹음하기](#)
- [의료 대화를 실시간 스트림으로 받아쓰기](#)
- [스피커 분할 사용 설정 설정 설정 설정 설정 설정](#)
- [멀티 채널 오디오 트랜스크립팅](#)

## 의료 대화의 오디오 파일 녹음하기

일괄 트랜스크립션 작업을 사용하여 의료 대화의 오디오 파일을 변환할 수 있습니다. 이를 사용하여 임상가와 환자의 대화를 필사할 수 있습니다. [StartMedicalTranscriptionJob](#) API 또는 [에서 일괄 트랜스크립션 작업을 시작할 수](#) AWS Management Console 있습니다.

[StartMedicalTranscriptionJob](#) API를 사용하여 의료 기록 작업을 시작할 때는 Specialty 파라미터의 PRIMARYCARE 값으로 지정합니다.



## AWS Management Console

### 임상의-환자 대화를 필사하려면 (AWS Management Console)

를 사용하여 임상의와 환자의 대화를AWS Management Console 필사하려면 트랜스크립션 작업을 생성하고 [오디오 입력 유형용 대화] 를 선택합니다.

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창의Amazon Transcribe 의료에서 트랜스크립션 작업을 선택합니다.
3. 작업 생성(Create job)을 선택합니다.
4. Job 세부 정보 지정 페이지의 작업 설정에서 다음을 지정합니다.
  - a. 이름 — 트랜스크립션 작업의 이름입니다.
  - b. 오디오 입력 유형 — 대화
5. 나머지 필드에는 오디오 파일의Amazon S3 위치와 트랜스크립션 작업의 출력을 저장할 위치를 지정합니다.
6. [Next]를 선택합니다.
7. Create를 선택합니다.

## API

일괄 트랜스크립션 작업 (API) 을 사용하여 의료 대화를 기록하려면

- [StartMedicalTranscriptionJob](#)API의 경우 다음을 지정합니다.
  - a. 의 경우MedicalTranscriptionJobName, 에 고유한 이름을 지정하십시오AWS 계정.
  - b. 의 경우LanguageCode 오디오 파일에서 사용되는 언어 및 어휘 필터의 언어에 해당하는 언어 코드를 지정하십시오.
  - c. Media개체의MediaFileUri 매개 변수로 변환하려는 오디오 파일의 이름을 지정합니다.
  - d. 의Specialty 경우 오디오 파일에서 말하는 임상의의 전문 의료 분야를 다음과 같이 지정하십시오PRIMARYCARE.
  - e. Type에 CONVERSATION을 지정합니다.
  - f. 의OutputBucketName 경우 트랜스크립션 결과를 저장할Amazon S3 버킷을 지정합니다.

다음은 를 사용하여 해당AWS SDK for Python (Boto3)PRIMARYCARE 전문 분야의 임상의와 환자의 의료 대화를 필사하는 요청 예시입니다.

```
from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
job_name = "my-first-med-transcription-job"
job_uri = "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-audio-file.flac"
transcribe.start_medical_transcription_job(
    MedicalTranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'DOC-EXAMPLE-BUCKET',
    OutputKey = 'output-files/',
    LanguageCode = 'en-US',
    Specialty = 'PRIMARYCARE',
    Type = 'CONVERSATION'
)

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

다음 예제 코드는 임상의-환자 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

일괄 트랜스크립션 작업 (AWS CLI) 을 사용하여 의료 대화를 기록하려면

- 다음 코드를 실행합니다.

```

aws transcribe start-medical-transcription-job \
--region us-west-2 \
--cli-input-json file://example-start-command.json

```

다음 코드는 의 콘텐츠를 보여줍니다 *example-start-command.json*.

```
{
  "MedicalTranscriptionJobName": "my-first-med-transcription-job",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-audio-
file.flac"
  },
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",
  "OutputKey": "my-output-files/",
  "LanguageCode": "en-US",
  "Specialty": "PRIMARYCARE",
  "Type": "CONVERSATION"
}
```

## 의료 대화를 실시간 스트림으로 받아쓰기

HTTP/2 또는 [WebSocket](#) 프로토콜을 사용하여 의료 대화의 오디오 스트림을 녹음할 수 있습니다. WebSocket 프로토콜을 사용하여 스트림을 시작하는 방법에 대한 자세한 내용은 [을 참조하십시오](#) [WebSocket 스트림 설정](#). HTTP/2 스트림을 시작하려면 [StartMedicalStreamTranscriptionAPI](#) 를 사용하세요.

다음과 같은 의료 전문 분야에서 스트리밍 오디오를 녹음할 수 있습니다.

- 심장학
- 신경학
- 종양학
- 기본 진료
- 비뇨기과

각 의료 전문 분야에는 다양한 유형의 시술 및 예약이 포함됩니다. 따라서 임상 의는 다양한 유형의 메모를 생성합니다. WebSocket 요청의 specialty URI 파라미터 값 또는 [StartMedicalStreamTranscriptionAPI](#) 파라미터를 지정하는 데 도움이 되는 다음 예제를 지침으로 사용하십시오. Specialty

- 전기생리학 또는 심장초음파 상담을 원하시면 선택하십시오 CARDIOLOGY.
- 의학 종양학, 외과 종양학 또는 방사선 종양학 상담을 원하시면 선택하십시오 ONCOLOGY.

- 뇌졸중 환자 (일시적 허혈성 발작 또는 뇌혈관 발작 중 하나) 에게 상담을 제공하는 의사의 경우 선택하십시오NEUROLOGY.
- 요실금에 대한 상담을 원하시면 선택하십시오UROLOGY.
- 연간 검진 또는 긴급 진료를 받으려면 선택하십시오PRIMARYCARE.
- 입원 환자 내원의 경우 선택하십시오PRIMARYCARE.
- 불임, 난관 결찰술, IUD 삽입 또는 낙태에 관한 상담을 원하시면 선택하십시오PRIMARYCARE.

## AWS Management Console

스트리밍 중인 의료 대화를 녹음하려면 (AWS Management Console)

를 사용하여 실시간 스트림에서 임상가와 환자의 대화를 녹음하려면 의료 대화를 필사하고 스트림을 시작한 다음 마이크에 대고 말하기 시작하는 옵션을 선택합니다.AWS Management Console

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창의Amazon Transcribe Medical 아래에서 실시간 트랜스크립션을 선택합니다.
3. 대화를 선택합니다.
4. 의료 전문 분야의 경우 임상가의 전문 분야를 선택하십시오.
5. 스트리밍 시작을 선택합니다.
6. 마이크에 대고 말하세요.

HTTP/2 스트림에서 의료 대화를 텍스트로 옮기기

다음은 HTTP/2 요청의 파라미터 구문입니다.

의료 대화의 HTTP/2 스트림을 트랜스크립션하려면 [StartMedicalStreamTranscriptionAPI](#)를 사용하고 다음을 지정하세요.

- LanguageCode— 언어 코드. 유효한 값은en-US
- MediaEncoding— 입력 오디오에 사용되는 인코딩입니다. 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` 매개 변수 값을 사용하려는 사용자 지정 어휘의 이름으로 설정합니다. 자세한 정보는 [의료 맞춤형 어휘를 통한 필사 정확도 향상](#)을 참조하세요.

다른 화자의 음성에서 레이블을 지정하려면 `ShowSpeakerLabel` 파라미터를 `true`로 설정합니다. 자세한 정보는 [스피커 분할 사용 설정](#)을 참조하세요.

의료 대화를 기록하기 위한 HTTP/2 스트림 설정에 대한 자세한 내용은 [HTTP/2 스트림 설정](#)을 참조하십시오.

WebSocket 스트림에서 의료 대화를 텍스트로 옮기기

WebSocket 요청을 사용하여 의료 대화를 녹음할 수 있습니다. WebSocket 요청을 하면 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](#)을 참조하십시오.

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 `vocabulary-name` 활성화하려면 `value` 값을 사용하려는 사용자 지정 어휘의 이름으로 설정합니다. 자세한 정보는 [의료 맞춤형 어휘를 통한 필사 정확도 향상](#)을 참조하세요.

다른 화자의 음성에 레이블을 지정하려면 `show-speaker-label` 매개 변수를 `true` 로 설정합니다. 자세한 정보는 [스피커 분할 사용 설정 설정 설정 설정 설정 설정](#)을 참조하세요.

미리 서명된 URI 만들기에 대한 자세한 내용은 [WebSocket 스트림 설정](#)을 참조하십시오.

## 스피커 분할 사용 설정 설정 설정 설정 설정 설정

Amazon Transcribe 메디컬에서 스피커 파티셔닝을 활성화하려면 스피커 다이얼라이제이션을 사용하십시오. 이를 통해 환자가 말한 내용과 임상가가 전사본 출력에서 말한 내용을 확인할 수 있습니다.

스피커 이음 기능을 활성화하면 Amazon Transcribe 메디컬은 각 스피커의 고유 식별자로 각 스피커 음성에 라벨을 붙입니다. 발화는 일반적으로 침묵에 의해 다른 발화와 분리되는 음성의 단위입니다. 일괄 트랜스크립션에서는 임상가가 한 사람의 음성에 라벨을 붙일 수 `spk_0` 있고, 환자는 라벨을 받을 수 `spk_1` 있습니다.

한 화자의 음성이 다른 화자의 발화와 겹치는 경우 Amazon Transcribe 의료 기관은 시작 시간을 기준으로 녹취록을 작성하도록 지시합니다. 입력 오디오에서 중첩되는 `utterance`는 트랜스크립션 출력에서 중첩되지 않습니다.

일괄 트랜스크립션 작업을 사용하거나 실시간 스트림을 사용하여 오디오 파일을 트랜스크립션할 때 스피커 다이얼라이제이션을 활성화할 수 있습니다.

### 주제

- [일괄 트랜스크립션에서 스피커 파티셔닝 활성화](#)
- [실시간 스트림에서 스피커 파티셔닝 활성화](#)

## 일괄 트랜스크립션에서 스피커 파티셔닝 활성화

[StartMedicalTranscriptionJob](#) API 또는 `awscli` 를 사용하여 일괄 트랜스크립션 작업에서 스피커 파티셔닝을 활성화할 수 AWS Management Console 있습니다. 이를 통해 임상가-환자 대화에서 발언자당 텍스트를 분할하고 트랜스크립션 출력에서 누가 무엇을 말했는지 확인할 수 있습니다.

### AWS Management Console

`awscli` 를 사용하여 트랜스크립션 작업에서 스피커 다이얼라이제이션을 활성화하려면 오디오 식별을 활성화한 다음 스피커 파티셔닝을 활성화해야 합니다. AWS Management Console

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창의 Amazon Transcribe 의료에서 트랜스크립션 작업을 선택합니다.
3. 작업 생성(Create job)을 선택합니다.



4. 작업 세부 정보 지정 페이지에서 트랜스크립션 작업에 대한 정보를 제공합니다.
5. Next(다음)를 선택합니다.
6. 오디오 식별을 활성화합니다.
7. 오디오 식별 유형에서 스피커 파티셔닝을 선택합니다.
8. 최대 스피커 수에 오디오 파일에서 말하고 있다고 생각되는 최대 스피커 수를 입력합니다.
9. Create(생성)를 선택합니다.

## API

일괄 트랜스크립션 작업 (API) 을 사용하여 스피커 파티셔닝을 활성화하려면

- [StartMedicalTranscriptionJob](#) API의 경우 다음을 지정합니다.
  - a. 의 경우 `MedicalTranscriptionJobName`, 에 고유한 이름을 지정하십시오 AWS 계정.
  - b. 의 `LanguageCode` 경우 오디오 파일에서 사용되는 언어에 해당하는 언어 코드를 지정합니다.
  - c. `Media` 개체의 `MediaFileUri` 매개 변수로 변환하려는 오디오 파일의 이름을 지정합니다.
  - d. 의 `Specialty` 경우 오디오 파일에 말하는 임상 의의 의료 전문 분야를 지정하십시오.
  - e. `Type` 에 `CONVERSATION` 을 지정합니다.
  - f. 의 `OutputBucketName` 경우 트랜스크립션 결과를 저장할 Amazon S3 버킷을 지정합니다.
  - g. `Settings` 객체에 대해 다음을 지정합니다.
    - i. `ShowSpeakerLabels` - `true`.
    - ii. `MaxSpeakerLabels`— 오디오에서 말하고 있다고 생각되는 화자의 수를 나타내는 2에서 10 사이의 정수입니다.

다음 요청에서는 AWS SDK for Python (Boto3) 를 사용하여 스피커 파티셔닝이 활성화된 상태에서 1차 진료 임상 의 환자 대화의 일괄 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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
transcribe.start_medical_transcription_job(
```

```

MedicalTranscriptionJobName = job_name,
Media={
    'MediaFileUri': job_uri
},
OutputBucketName = 'DOC-EXAMPLE-BUCKET',
OutputKey = 'my-output-files/',
LanguageCode = 'en-US',
Specialty = 'PRIMARYCARE',
Type = 'CONVERSATION',
OutputBucketName = 'DOC-EXAMPLE-BUCKET',
Settings = {'ShowSpeakerLabels': True,
            'MaxSpeakerLabels': 2
            }
)
while True:
    status = transcribe.get_medical_transcription_job(MedicalTranscriptionJobName =
job_name)
    if status['MedicalTranscriptionJob']['TranscriptionJobStatus'] in ['COMPLETED',
'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
print(status)

```

다음 예제 코드는 스피커 파티셔닝이 활성화된 상태의 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 진료를 받는 임상 의와 환자 간의 대화를 오디오 파일로 녹음하려면 (AWS CLI)

- 다음 코드를 실행합니다.

```

aws transcribe start-transcription-job \
--region us-west-2 \
--cli-input-json file://example-start-command.json

```

다음 코드는 예 콘텐츠의 내용을 보여줍니다 *example-start-command.json*.

```

{
  "MedicalTranscriptionJobName": "my-first-med-transcription-job",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-audio-file.flac"
  },
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",
  "OutputKey": "my-output-files/",
  "LanguageCode": "en-US",
  "Specialty": "PRIMARYCARE",
  "Type": "CONVERSATION",
  "Settings": {
    "ShowSpeakerLabels": true,
    "MaxSpeakerLabels": 2
  }
}

```

}

## 실시간 스트림에서 스피커 파티셔닝 활성화

실시간 스트림에서 발언자를 분할하고 음성에 레이블을 지정하려면 AWS Management Console 또는 스트리밍 요청을 사용하십시오. 스피커 파티셔닝은 스트림에 있는 두 명에서 다섯 명 사이의 스피커에 가장 적합합니다. Amazon Transcribe Medical은 스트림에서 5개 이상의 스피커를 파티션할 수 있지만 이 수를 초과하면 파티션의 정확도가 떨어집니다.

HTTP/2 요청을 시작하려면 [StartMedicalStreamTranscription](#) API를 사용하세요. WebSocket 요청을 시작하려면 미리 서명된 URI를 사용하십시오. URI에는 애플리케이션과 Amazon Transcribe Medical 간에 양방향 통신을 설정하는 데 필요한 정보가 포함되어 있습니다.

마이크에 음성으로 전달되는 오디오의 스피커 파티셔닝 활성화 (AWS Management Console)

를 사용하여 의사-환자 대화의 실시간 스트리밍을 시작하거나 마이크에 실시간으로 음성으로 전달되는 받아쓰기를 시작할 수 있습니다. AWS Management Console

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창에서 Amazon Transcribe 의료에 대해 실시간 기록을 선택합니다.
3. 오디오 입력 유형에서 필사하려는 의료 음성의 유형을 선택합니다.
4. 추가 설정에서 스피커 파티셔닝을 선택합니다.
5. 스트리밍 시작을 선택하여 실시간 오디오 트랜스크립션을 시작합니다.
6. 마이크에 대고 말하세요.

HTTP/2 스트림의 스피커 분할 사용 설정 설정 설정 설정 설정 설정 설정 설정 설정 설정 설정

의료 대화의 HTTP/2 스트림에서 스피커 파티셔닝을 활성화하려면

[StartMedicalStreamTranscription](#) API를 사용하고 다음을 지정하십시오.

- 의 `LanguageCode` 경우 스트림의 언어에 해당하는 언어 코드를 지정합니다. 유효한 값은 en-US입니다.
- 의 `MediaSampleHertz` 경우 오디오의 샘플 속도를 지정합니다.
- 의 경우 `Specialty`, 제공자의 의료 전문 분야를 지정하십시오.
- `ShowSpeakerLabel - true`

의료 대화를 기록하기 위한 HTTP/2 스트림 설정에 대한 자세한 내용은 [을 참조하십시오](#) [HTTP/2 스트림 설정](#).

### WebSocket 요청에서 스피커 파티셔닝 활성화

API를 사용하여 WebSocket 스트림의 스피커를 분할하려면 다음 형식을 사용하여 미리 서명된 URI를 만들어 WebSocket 요청을 시작하고 `show-speaker-label` 로 설정합니다 `true`.

```
GET wss://transcribestreaming.us-west-2.amazonaws.com:8443/medical-stream-
transcription-websocket
?language-code=languageCode
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIAIOSFODNN7EXAMPLE%2F20220208%2Fus-
west-2%2Ftranscribe%2Faws4_request
&X-Amz-Date=20220208T235959Z
&X-Amz-Expires=300
&X-Amz-Security-Token=security-token
&X-Amz-Signature=Signature Version 4 signature
&X-Amz-SignedHeaders=host
&media-encoding=flac
&sample-rate=16000
&session-id=sessionId
&specialty=medicalSpecialty
&type=CONVERSATION
&vocabulary-name=vocabularyName
&show-speaker-label=boolean
```

다음 코드는 스트리밍 요청의 잘린 예제 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 메디컬은 자연스러운 음성 구간 (예: 스피커 변경 또는 오디오 일시 중지) 을 기반으로 수신 오디오 스트림을 중단합니다. 트랜스크립션은 전체 세그먼트가 텍스트로 변환되어 기록될 때까지 애플리케이션에 점진적으로 반환되고, 각 응답에는 텍스트로 변환된 음성이 포함되어 있습니다. 위 코드는 완전히 변환된 음성 세그먼트를 잘라낸 예제입니다. 스피커 레이블은 완전히 텍스트로 변환된 부분에만 표시됩니다.

다음 목록은 스트리밍 트랜스크립션 출력의 객체 및 매개 변수 구성을 보여줍니다.



## Transcript

각 음성 세그먼트에는 자체Transcript 객체가 있습니다.

## Results

각Transcript 객체에는 고유한Results 객체가 있습니다. 이 객체에는isPartial 필드가 포함되어 있습니다. 값이false 인 경우 전체 음성 세그먼트에 대한 결과가 반환됩니다.

## Alternatives

각Results 객체에는Alternatives 객체가 있습니다.

## Items

각Alternatives 개체에는 트랜스크립션 출력의 각 단어 및 구두점에 대한 정보가 들어 있는 자체Items 개체가 있습니다. 스피커 파티셔닝을 활성화하면 각 단어에는 완전히 녹음된 음성 세그먼트에 대한Speaker 레이블이 있습니다. Amazon Transcribe 메디컬은 이 레이블을 사용하여 스트림의 각 스피커에 고유한 정수를 할당합니다. 값이 인Type 매개변수는 한 사람이 말을 멈추었고 다른 사람이 곧 말을 시작하려 한다는 것을speaker-change 나타냅니다.

## Transcript

각 Items 객체에는 필드의 값으로 기록된 음성 세그먼트가 포함됩니다.Transcript

WebSocket 요청에 대한 자세한 내용은 을 참조하십시오 [WebSocket 스트림 설정](#).

## 멀티 채널 오디오 트랜스크립팅

여러 채널이 있는 오디오 파일이나 스트림이 있는 경우 채널 식별을 사용하여 각 채널의 음성을 변환할 수 있습니다. Amazon Transcribe 메디컬은 각 채널의 음성을 개별적으로 녹음합니다. 각 채널의 개별 트랜스크립션을 단일 트랜스크립션 출력으로 결합합니다.

채널 식별을 사용하여 오디오에서 개별 채널을 식별하고 각 채널의 음성을 변환할 수 있습니다. 발신자 및 상담원 시나리오와 같은 상황에서 이 기능을 활성화하세요. 이를 사용하여 약물 안전 모니터링을 수행하는 컨택 센터의 녹화 또는 스트림에서 발신자와 상담원을 구분할 수 있습니다.

일괄 처리와 실시간 스트리밍 모두에 대해 채널 식별을 활성화할 수 있습니다. 다음 목록은 각 메시드에 대해 활성화하는 방법을 설명합니다.

- Batch 트랜스크립션 —AWS Management Console 및 [StartMedicalTranscriptionJobAPI](#)

- 스트리밍 트랜스크립션 — WebSocket 스트리밍 및 [StartMedicalStreamTranscriptionAPI](#)

## 멀티 채널 오디오 파일 트랜스크립팅

오디오 파일을 텍스트로 변환하면 Amazon Transcribe Medical은 각 채널의 항목 목록을 반환합니다. 항목은 텍스트로 변환된 단어 또는 문장 부호입니다. 각 단어에는 시작 시간과 종료 시간이 있습니다. 한 채널에 있는 사람이 별도의 채널에 있는 사람을 대신하여 말하는 경우 각 채널의 항목의 시작 시간과 종료 시간이 서로 겹칩니다.

기본적으로 두 채널로 오디오 파일을 변환할 수 있습니다. 채널이 3개 이상인 파일을 트랜스크립션해야 하는 경우 할당량 증가를 요청할 수 있습니다. 할당량 증가의 요청에 관한 자세한 내용은 [AWS 서비스쿼터를](#) 참조하세요.

일괄 트랜스크립션 작업에서 다중 채널 오디오를 트랜스크립션하려면 AWS Management Console 또는 [StartMedicalTranscriptionJobAPI](#)를 사용하십시오.

### AWS Management Console

를 사용하여 일괄 트랜스크립션 작업에서 채널 식별을 활성화하려면 오디오 식별을 활성화한 다음 채널 식별을 활성화합니다. AWS Management Console 채널 식별은 의 오디오 식별의 하위 AWS Management Console 집합입니다.

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창의 Amazon Transcribe 의료에서 트랜스크립션 작업을 선택합니다.
3. 작업 생성(Create job)을 선택합니다.
4. 작업 세부 정보 지정 페이지에서 트랜스크립션 작업에 대한 정보를 제공합니다.
5. Next(다음)를 선택합니다.
6. 오디오 식별을 활성화합니다.
7. 오디오 식별 유형에서 채널 식별을 선택합니다.
8. Create(생성)를 선택합니다.

### API

다중 채널 오디오 파일 (API) 을 텍스트로 변환하려면

- [StartMedicalTranscriptionJobAPI](#)의 경우 다음을 지정합니다.

- a. `TranscriptionJobName` 경우 고유한 이름을 지정하십시오AWS 계정.
- b. `LanguageCode` 경우 오디오 파일에서 사용되는 언어에 해당하는 언어 코드를 지정합니다. 유효한 값은 `en-US`입니다.
- c. `Media` 오브젝트의 `MediaFileUri` 파라미터로 트랜스크립션하려는 미디어 파일의 이름을 지정합니다.
- d. `Settings` 객체의 경우 `ChannelIdentification` 로 설정합니다 `true`.

다음은  `boto3`  를 사용한 예제 요청입니다AWS SDK for Python (Boto3).

```

from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe', 'us-west-2')
job_name = "my-first-transcription-job"
job_name = "my-first-med-transcription-job"
job_uri = "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
transcribe.start_medical_transcription_job(
    MedicalTranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'DOC-EXAMPLE-BUCKET',
    OutputKey = 'output-files/',
    LanguageCode = 'en-US',
    Specialty = 'PRIMARYCARE',
    Type = 'CONVERSATION',
    Settings = {
        'ChannelIdentification': True
    }
)
while True:
    status = transcribe.get_transcription_job(MedicalTranscriptionJobName = job_name)
    if status['MedicalTranscriptionJob']['TranscriptionJobStatus'] in ['COMPLETED',
'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
print(status)

```



```
    }
  ],
  "channel_labels": {
    "channels": [
      {
        "channel_label": "ch_0",
        "items": [
          {
            "start_time": "12.282",
            "end_time": "12.592",
            "alternatives": [
              {
                "confidence": "1.0000",
                "content": "When"
              }
            ],
            "type": "pronunciation"
          },
          {
            "start_time": "12.592",
            "end_time": "12.692",
            "alternatives": [
              {
                "confidence": "0.8787",
                "content": "you"
              }
            ],
            "type": "pronunciation"
          },
          {
            "start_time": "12.702",
            "end_time": "13.252",
            "alternatives": [
              {
                "confidence": "0.8318",
                "content": "try"
              }
            ],
            "type": "pronunciation"
          },
          ...
        ]
      }
    ],
  }
```

```
"channel_label": "ch_1",
"items": [
  {
    "start_time": "12.379",
    "end_time": "12.589",
    "alternatives": [
      {
        "confidence": "0.5645",
        "content": "It"
      }
    ],
    "type": "pronunciation"
  },
  {
    "start_time": "12.599",
    "end_time": "12.659",
    "alternatives": [
      {
        "confidence": "0.2907",
        "content": "seems"
      }
    ],
    "type": "pronunciation"
  },
  {
    "start_time": "12.669",
    "end_time": "13.029",
    "alternatives": [
      {
        "confidence": "0.2497",
        "content": "to"
      }
    ],
    "type": "pronunciation"
  },
  ...
]
}
```

## 멀티 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 트랜스크립션해야 하는 경우 할당량 증가를 요청할 수 있습니다. 할당량 증가의 요청에 관한 자세한 내용은 [AWS 서비스 할당량을](#) 참조하세요.

### HTTP/2 스트림의 다중 채널 오디오 트랜스크립팅

HTTP/2 스트림에서 다중 채널 오디오를 변환하려면 [StartMedicalStreamTranscription](#) API를 사용하고 다음을 지정하십시오.

- `LanguageCode`— 오디오의 언어 코드입니다. 유효한 값은 en-US입니다.
- `MediaEncoding`— 오디오 인코딩. 유효한 값은 ogg-opus, flac, pcm입니다.
- `EnableChannelIdentification` - true
- `NumberOfChannels`— 스트리밍 오디오의 채널 수

의료 대화를 기록하기 위한 HTTP/2 스트림 설정에 대한 자세한 내용은 [HTTP/2 스트림 설정](#)을 참조하십시오.

### WebSocket 스트림의 다중 채널 오디오 트랜스크립팅

WebSocket 스트림에서 스피커를 분할하려면 다음 형식을 사용하여 미리 서명된 URI를 만들고 WebSocket 요청을 시작하십시오. `enable-channel-identificationtrueas`와 스트림의 채널 수를 `number-of-channels`에 지정하십시오. 사전 서명된 URI에는 애플리케이션과 Amazon Transcribe Medical 간에 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
```

매개변수 [정의는 API 참조에서](#) 찾을 수 있습니다. 모든 AWS API 작업에 공통적인 [매개변수는 일반 매개변수](#) 섹션에 나열되어 있습니다.

WebSocket 요청에 대한 자세한 내용은 [을 참조하십시오](#) [WebSocket 스트림 설정](#).

멀티 채널 스트리밍 출력

HTTP/2 및 WebSocket 요청의 스트리밍 트랜스크립션 출력은 동일합니다. 다음은 예제 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 해당 음성이 속한 채널을 나타내는 `channelId` 플래그가 있습니다.

## 의료 받아쓰기 받아쓰기

Amazon Transcribe Medical을 사용하면 일괄 기록 작업 또는 실시간 스트림을 사용하여 의사가 지시한 의료 기록을 기록할 수 있습니다. Batch 트랜스크립션 작업을 통해 오디오 파일을 트랜스크립션할 수 있습니다. 필사 작업 또는 스트림에서 임상외의 의료 전문 분야를 지정하여 Amazon Transcribe Medical 팀이 가능한 한 가장 높은 정확도로 전사 결과를 생성하도록 하십시오.

다음 전문 분야의 의료 받아쓰기를 필사할 수 있습니다.

- 심장학 — 스트리밍 트랜스크립션으로만 이용 가능
- 신경학 — 스트리밍 트랜스크립션에서만 사용 가능
- 종양학 — 스트리밍 트랜스크립션으로만 사용 가능
- 1차 진료 — 다음과 같은 유형의 의료 행위를 포함합니다.
  - 가정의학
  - 내과학
  - 산부인과학
  - 소아과학
- 방사선학 — 스트리밍 트랜스크립션으로만 제공
- 비뇨기과 — 스트리밍 트랜스크립션으로만 이용 가능

사용자 지정 어휘를 사용하여 필사 정확도를 높일 수 있습니다. 의료 사용자 지정 어휘의 작동 방식에 대한 자세한 내용은 [을 참조하십시오](#) [의료 맞춤형 어휘를 통한 필사 정확도 향상](#).

기본적으로 Amazon Transcribe Medical은 신뢰도가 가장 높은 트랜스크립션을 반환합니다. 대체 트랜스크립션을 반환하도록 구성하려면 [을 참조하십시오](#) [대체 트랜스크립션 생성](#).

수치와 의료 측정값이 전사 출력에 표시되는 방식에 대한 자세한 내용은 [숫자 표기하기](#) 및 [을 참조하십시오](#) [의료 용어 및 측정값 필사](#).

### 주제

- [의료 받아쓰기의 오디오 파일 녹음하기](#)
- [의료 받아쓰기를 실시간 스트림으로 받아쓰기](#)

## 의료 받아쓰기의 오디오 파일 녹음하기

일괄 트랜스크립션 작업을 사용하여 의료 대화의 오디오 파일을 변환할 수 있습니다. 이를 사용하여 임상 의사와 환자의 대화를 필사할 수 있습니다. [StartMedicalTranscriptionJob](#) API 또는 에서 일괄 트랜스크립션 작업을 시작할 수 AWS Management Console 있습니다.

[StartMedicalTranscriptionJob](#) API를 사용하여 의료 기록 작업을 시작할 때는 Specialty 파라미터의 PRIMARYCARE 값으로 지정합니다.

### AWS Management Console

임상의-환자 대화를 필사하려면 (AWS Management Console)

를 사용하여 임상 의사와 환자의 대화를 AWS Management Console 필사하려면 트랜스크립션 작업을 생성하고 [오디오 입력 유형용 대화] 를 선택합니다.

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창의 Amazon Transcribe '의료' 에서 '트랜스크립션 작업' 에서 '트랜스크립션 작업을 선택합니다.
3. 작업 생성(Create job)을 선택합니다.
4. Job 세부 정보 지정 페이지의 작업 설정에서 다음을 지정합니다.
  - a. 이름 — 트랜스크립션 작업의 이름입니다.
  - b. 오디오 입력 유형 — 받아쓰기
5. 나머지 필드에는 오디오 파일의 Amazon S3 위치와 트랜스크립션 작업의 출력을 저장할 위치를 지정합니다.
6. [Next]를 선택합니다.
7. Create를 선택합니다.

### API

일괄 트랜스크립션 작업 (API) 을 사용하여 의료 대화를 기록하려면

- [StartMedicalTranscriptionJob](#) API의 경우 다음을 지정합니다.
  - a. 의 경우 MedicalTranscriptionJobName, 에 고유한 이름을 지정하십시오 AWS 계정.
  - b. 의 경우 LanguageCode 오디오 파일에서 사용되는 언어 및 어휘 필터의 언어에 해당하는 언어 코드를 지정하십시오.

- c. Media개체의MediaFileUri 매개 변수에서 변환하려는 오디오 파일의 이름을 지정합니다.
- d. 의Specialty 경우 오디오 파일에 말하는 임상어의 의료 전문 분야를 지정하십시오.
- e. Type에 DICTATION을 지정합니다.
- f. 의OutputBucketName 경우 트랜스크립션 결과를 저장할Amazon S3 버킷을 지정합니다.

다음은 를 사용하여 해당PRIMARYCARE 전문 분야의 임상어가 작성한 의료 기록을 필사하는 요청 예시입니다.AWS SDK for Python (Boto3)

```

from __future__ import print_function
import time
import boto3
transcribe = boto3.client('transcribe')
job_name = "my-first-med-transcription-job"
job_uri = "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-audio-file.flac"
transcribe.start_medical_transcription_job(
    MedicalTranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'DOC-EXAMPLE-BUCKET',
    OutputKey = 'my-output-files/',
    LanguageCode = 'en-US',
    Specialty = 'PRIMARYCARE',
    Type = 'DICTATION'
)
while True:
    status = transcribe.get_medical_transcription_job(MedicalTranscriptionJobName =
job_name)
    if status['MedicalTranscriptionJob']['TranscriptionJobStatus'] in ['COMPLETED',
'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
print(status)

```

다음 예제 코드는 의료 받아쓰기의 기록 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

일괄 트랜스크립션 작업 (AWS CLI) 에서 스피커 파티셔닝을 활성화하려면

- 다음 코드를 실행합니다.

```
aws transcribe start-medical-transcription-job \
--region us-west-2 \
--cli-input-json file://example-start-command.json
```

다음 코드는 의 콘텐츠를 보여줍니다 *example-start-command.json*.

```
{
  "MedicalTranscriptionJobName": "my-first-med-transcription-job",
  "Media": {
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-audio-file.flac"
  },
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",
  "OutputKey": "my-output-files/",
  "LanguageCode": "en-US",
  "Specialty": "PRIMARYCARE",
  "Type": "DICTATION"
}
```

## 의료 받아쓰기를 실시간 스트림으로 받아쓰기

WebSocket 스트림을 사용하여 의료 받아쓰기를 오디오 스트림으로 변환하세요. 를 사용하여 본인이나 다른 사람이 말하는 음성을 마이크에 직접 녹음할 수도 있습니다. AWS Management Console

HTTP/2 또는 WebSocket 스트림의 경우 다음과 같은 의료 전문 분야의 오디오를 녹음할 수 있습니다.

- 심장학
- 종양학
- 신경학
- 기본 진료
- 방사선학
- 비뇨기과

각 의료 전문 분야에는 다양한 유형의 시술 및 예약이 포함됩니다. 따라서 임상의는 다양한 유형의 메모를 생성합니다. WebSocket 요청의 `specialty` URI 파라미터 값 또는 [StartMedicalStreamTranscription](#) API 파라미터를 지정하는 데 도움이 되는 다음 예제를 지침으로 사용하십시오. `Specialty`

- 전기생리학 또는 심장초음파 시술 후 받아쓰기를 원하시면 선택하십시오 `CARDIOLOGY`.
- 외과 종양학 또는 방사선 종양학 시술 후 받아쓰기를 원하시면 선택하십시오 `ONCOLOGY`.
- 의사가 뇌염 진단을 지시하는 메모를 작성하려면 선택하십시오 `NEUROLOGY`.
- 방광결석을 분해하기 위한 시술 기록을 보려면 선택하십시오 `UROLOGY`.
- 내과 상담 후 임상 기록을 받아쓰려면 선택하십시오 `PRIMARYCARE`.
- 의사가 CT 스캔, PET 스캔, MRI 또는 방사선 촬영 결과를 전달하도록 지시하려면 선택하십시오 `RADIOLOGY`.
- 산부인과 진료 후 의사 소견서를 받아쓰려면 선택하십시오 `PRIMARYCARE`.

실시간 스트림에서 특정 용어의 트랜스크립션 정확도를 높이려면 사용자 지정 어휘를 사용하세요. 사용자 지정 어휘를 `vocabulary-name` 활성화하려면 `의 값`을 사용하려는 사용자 지정 어휘의 이름으로 설정합니다.

를 사용하여 마이크에 음성 받아쓰기를 녹음하기 [AWS Management Console](#)

를 사용하여 의료 받아쓰기의 스트리밍 오디오를 변환하려면 의료 받아쓰기를 변환하고 스트림을 시작한 다음 마이크에 대고 말하기 시작하는 옵션을 선택합니다. [AWS Management Console](#)

의료 받아쓰기의 스트리밍 오디오를 녹음하려면 ([AWS Management Console](#))

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창의 `[Amazon TranscribeMedical]` 에서 '실시간 트랜스크립션' 에서 '실시간 트랜스크립션' 에서 '실시간 트랜스크립션'
3. 받아쓰기를 선택합니다.
4. 의료 전문 분야의 경우 스트림에서 말하는 임상의의 의료 전문 분야를 선택하십시오.
5. 스트리밍 시작을 선택합니다.
6. 마이크에 대고 말하세요.

## HTTP의 받아쓰기

의료 받아쓰기의 HTTP/2 스트림을 트랜스크립션하려면

[StartMedicalStreamTranscription](#) API를 사용하고 다음을 지정하세요.

- LanguageCode— 언어 코드. 유효한 값은 en-US
- MediaEncoding— 입력 오디오에 사용되는 인코딩입니다. 유효한 값은 pcm, ogg-opus, flac입니다.
- Specialty— 의료 전문가의 전문 분야.
- Type – DICTATION

의료 받아쓰기를 기록하기 위한 HTTP/2 스트림 설정에 대한 자세한 내용은 [이 링크](#)를 참조하십시오 [HTTP/2 스트림 설정](#).

WebSocket 스트리밍 요청을 사용하여 의료 받아쓰기 받아쓰기

WebSocket 요청을 사용하여 실시간 스트림에서 의료 받아쓰기를 기록하려면 미리 서명된 URI를 생성해야 합니다. 이 URI에는 애플리케이션과 Amazon Transcribe Medical 간에 오디오 스트림을 설정하는데 필요한 정보가 포함되어 있습니다. WebSocket 요청 생성에 대한 자세한 내용은 [이 링크](#)를 참조하십시오 [WebSocket 스트림 설정](#).

다음 템플릿을 사용하여 미리 서명된 URI의 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 만들기에 대한 자세한 내용은 [을 참조하십시오 WebSocket 스트림 설정](#).

## 의료 맞춤형 어휘를 통한 필사 정확도 향상

Amazon Transcribe의 의료 분야의 필사 정확도를 높이려면 하나 이상의 의료 사용자 지정 어휘를 만들어 사용하십시오. 사용자 지정 어휘는 영역별 단어 또는 구문의 모음입니다. 이 컬렉션은 이러한 단어나 문구를 필사하는 데 있어 Amazon Transcribe Medical의 성능을 향상시키는 데 도움이 됩니다.

Amazon Transcribe 메디컬을 사용할 때 자신의 데이터 무결성에 대한 책임은 귀하에게 있습니다. 사용자 지정 어휘에는 기밀 정보, 개인 정보 (PII) 또는 보호 대상 건강 정보 (PHI) 를 입력하지 마십시오.

최상의 결과를 얻으려면 각각 특정 오디오 녹음을 녹음하는 데 도움이 되는 별도의 작은 사용자 지정 어휘를 만드세요. 모든 녹음에 사용할 대용량의 사용자 지정 어휘를 하나 만들었을 때보다 트랜스크립션 정확도가 크게 향상됩니다.

기본적으로 최대 100개의 사용자 지정 어휘를 사용할 수 있습니다 AWS 계정. 사용자 지정 어휘는 크기가 50KB를 초과할 수 없습니다. 사용할 수 있는 사용자 지정 어휘의 수를 늘려 달라고 요청하는 방법에 대한 자세한 내용은 [AWS 서비스 할당량을 참조하십시오](#). AWS 계정

사용자 지정 어휘는 미국 영어 (en-US) 로 제공됩니다.

### 주제

- [의료 맞춤 어휘를 위한 텍스트 파일 만들기](#)
- [텍스트 파일을 사용하여 맞춤형 의료 어휘 만들기](#)
- [의료 맞춤 어휘를 사용하여 오디오 파일 녹음하기](#)
- [의료 맞춤 어휘를 사용하여 실시간 스트림 자막 작성하기](#)
- [Amazon Transcribe 의료용 문자 세트](#)

## 의료 맞춤 어휘를 위한 텍스트 파일 만들기

사용자 지정 어휘를 만들려면 UTF-8 형식의 텍스트 파일을 만듭니다. 이 파일에서는 각 열에 필드를 지정하는 4열 테이블을 만듭니다. 각 필드는 분야별 용어가 어떻게 발음되는지 또는 이러한 용어를 트랜스크립션에 표시하는 방법을 Amazon Transcribe 메디컬에 알려줍니다. 이러한 필드가 포함된 텍스트 파일을 Amazon S3 버킷에 저장합니다.

## 텍스트 파일 형식 지정 방법 이해하기

의학 사용자 지정 어휘를 만들려면 열 이름을 헤더 행으로 입력합니다. 헤더 행 아래에 각 열의 값을 입력합니다.

다음은 테이블에 있는 4개 열의 이름입니다.

- `Phrase`— 열 필수, 값 필수
- `IPA`— 열 필수, 값은 선택 사항일 수 있습니다.
- `SoundsLike`— 열 필수, 값은 선택 사항일 수 있습니다.
- `DisplayAs`— 열 필수, 값은 선택 사항일 수 있습니다.

사용자 지정 어휘를 생성합니다. 다음 사항을 확인하십시오.

- 각 열을 단일 탭 문자로 구분합니다. Amazon Transcribe 공백이나 여러 Tab 문자 여러 Tab 문자 여러 문자를 사용하여 열을 구분하려고 시도할 경우 오류 메시지가 발생합니다.
- 열 내의 각 값 뒤에 후행 공백이나 공백이 없어야 합니다.

각 열에 값을 입력해야 합니다.

- 하이픈을 포함하여 256자 미만이어야 합니다.
- 허용된 문자 집합의 문자만 사용하십시오. 을 참조하십시오 [Amazon Transcribe의 의료용 문자 세트](#).

## 테이블의 열에 대한 값 입력

다음 정보는 테이블의 4개 열에 값을 지정하는 방법을 보여줍니다.

- **Phrase**— 인식되어야 하는 문자 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 하이

폰을 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 의료용 문자 세트](#) 단원을 참조하십시오. DisplayAs 열을 지정하지 않는 경우 Amazon Transcribe Medical은 출력 파일의 Phrase 열 콘텐츠를 사용합니다.

- **IPA**(열 필수, 값은 선택 사항일 수 있음) — 단어나 구의 발음을 지정하려면 이 열에 [IPA \(국제 발음 알파벳\)](#) 문자를 포함시킬 수 있습니다. 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 의료용 문자 세트](#) 단원을 참조하십시오. 미국 영어는 Amazon Transcribe 의학에서 사용할 수 있는 유일한 언어입니다.

- **SoundsLike**(열 필수, 값은 선택 사항일 수 있음) — 단어나 구를 더 작은 부분으로 나누고 언어의 표준 철자법을 사용하여 단어의 발음을 모방하여 각 부분에 대한 발음을 제공할 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-ty-with-sub-cor-ti-cul-in-farcts-and-lewk-o-en-ce-phul-ah-pu-ty**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 Transcribe의 의료용 문자 세트](#) 단원을 참조하십시오.

- **DisplayAs**(열 필수, 값은 선택 사항일 수 있음) — 단어나 문구가 출력될 때 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.

DisplayAs열을 지정하지 않는 경우 Amazon Transcribe Medical은 입력 파일의 Phrase 열을 출력에 사용합니다.

DisplayAs 열에 UTF-8 문자를 사용할 수 있습니다.

IPA 및 DisplayAs 열의 값에만 공백을 포함할 수 있습니다.

사용자 지정 어휘의 텍스트 파일을 만들려면 텍스트 파일의 각 단어나 문구를 별도의 줄에 배치하세요. 탭 문자로 열을 구분합니다. IPA 및 DisplayAs 열의 값에만 공백을 포함합니다. 확장자가 있는 파일을 .txt Amazon Transcribe Medical을 사용하여 사용자 지정 어휘를 만들 때와 동일한 Amazon S3 AWS 리전 버킷에 저장하세요.

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 예제에서는 사용자 지정 어휘를 작성하는 데 사용할 수 있는 텍스트를 보여줍니다. 이러한 예제에서 사용자 정의 어휘를 작성하려면 예제를 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 예제에서는 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 예제를 html 형식으로 더 명확하게 보여줍니다. 이는 예제를 설명하기 위한 것일 뿐입니다.

구절	IPA	SoundsLike	DisplayAs
acute-respiratory-distress-syndrome			acute respiratory distress syndrome
A.L.L.	eɪ ɛ   ɛ l		모두
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   ɛ l	모두

DisplayAs	SoundsLike	IPA	구절
acute respiratory distress syndrome			acute-respiratory-distress-syndrome

DisplayAs	SoundsLike	IPA	구절
모두		eɪ ε   ε	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 맞춤형 의료 어휘 만들기

사용자 지정 어휘를 만들려면 단어나 구문 모음이 포함된 텍스트 파일을 준비해야 합니다. Amazon Transcribe Medical은 이 문자 문자 어휘를 사용하여 문자 어휘를 생성합니다.

[CreateMedicalVocabulary](#) API 또는 Amazon Transcribe Medical 콘솔을 사용하여 사용자 지정 어휘를 만들 수 있습니다.

### AWS Management Console

를 사용하여 사용자 지정 어휘를 AWS Management Console 만들려면 단어나 문구가 포함된 텍스트 파일의 Amazon S3 URI를 제공해야 합니다.

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창의 Amazon Transcribe Medical에서 사용자 지정 어휘를 선택합니다.
3. [이름]의 [어휘 설정]에서 사용자 지정 어휘의 이름을 선택합니다.
4. Amazon S3에서 오디오 파일 또는 비디오 파일의 위치를 지정합니다.
  - S3의 어휘 입력 파일 위치의 어휘 설정에서 사용자 지정 어휘를 생성하는 데 사용할 텍스트 파일을 식별하는 Amazon S3 URI를 지정합니다.
  - S3에서의 어휘 입력 파일 위치의 경우 Browse S3를 선택하여 텍스트 파일을 찾아 선택합니다.
5. 어휘 만들기를 선택합니다.

에서 사용자 지정 어휘의 처리 상태를 확인할 수 AWS Management Console 있습니다.

### API

의료 사용자 지정 어휘 (API) 를 만들려면

- [StartTranscriptionJob](#) API의 경우 다음을 지정합니다.

- a. LanguageCode에 en-US을 지정합니다.
- b. 의VocabularyFileUri 경우 사용자 지정 어휘를 정의하는 데 사용할 텍스트 파일의 Amazon S3 위치를 지정합니다.
- c. 의VocabularyName 경우 사용자 지정 어휘의 이름을 지정하십시오. 지정하는 이름은 내에서 고유해야 합니다AWS 계정.

사용자 지정 어휘의 처리 상태를 보려면 [GetMedicalVocabulary](#) API를 사용하세요.

다음은 를 사용하여 사용자 지정AWS SDK for Python (Boto3) 어휘를 생성합니다.

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

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

## AWS CLI

일괄 트랜스크립션 작업 (AWS CLI) 에서 스피커 파티셔닝을 활성화하려면

- 다음 코드를 실행합니다.

```
aws transcribe create-medical-vocabulary \
--vocabulary-name my-first-vocabulary \
```

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

## 의료 맞춤 어휘를 사용하여 오디오 파일 녹음하기

[StartMedicalTranscriptionJob](#) 또는 [CLI](#) 를 사용하여 사용자 지정 AWS Management Console 어휘를 사용하여 필사 정확도를 높이는 트랜스크립션 작업을 시작할 수 있습니다.

### AWS Management Console

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창의 Amazon Transcribe 의료에서 트랜스크립션을 선택합니다.
3. 작업 생성(Create job)을 선택합니다.
4. 작업 세부 정보 지정 페이지에서 트랜스크립션 작업에 대한 정보를 제공합니다.
5. Next(다음)를 선택합니다.
6. 사용자 지정에서 사용자 지정 어휘를 활성화합니다.
7. 어휘 선택에서 사용자 지정 어휘를 선택합니다.
8. Create(생성)를 선택합니다.

### API

일괄 트랜스크립션 작업 (API) 을 사용하여 오디오 파일에서 스피커 파티셔닝을 활성화하려면

- [StartMedicalTranscriptionJob](#) API의 경우 다음을 지정합니다.
  - a. 의 경우 `MedicalTranscriptionJobName`, 에 고유한 이름을 지정하십시오 AWS 계정.
  - b. 의 경우 `LanguageCode` 오디오 파일에서 사용되는 언어 및 어휘 필터의 언어에 해당하는 언어 코드를 지정하십시오.
  - c. `Media` 개체의 `MediaFileUri` 매개 변수로 변환하려는 오디오 파일의 이름을 지정합니다.
  - d. 의 `Specialty` 경우 오디오 파일에 말하는 임상 의의 의료 전문 분야를 지정하십시오.
  - e. 의 `Type` 경우 오디오 파일이 대화인지 받아쓰기인지를 지정합니다.
  - f. 의 `OutputBucketName` 경우 트랜스크립션 결과를 저장할 Amazon S3 버킷을 지정합니다.
  - g. `Settings` 객체에 대해 다음을 지정합니다.



- VocabularyName— 사용자 지정 어휘의 이름

다음 요청에서는 AWS SDK for Python (Boto3) 를 사용하여 사용자 지정 어휘로 일괄 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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
transcribe.start_medical_transcription_job(
    MedicalTranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'DOC-EXAMPLE-BUCKET',
    OutputKey = 'my-output-files/',
    LanguageCode = 'en-US',
    Specialty = 'PRIMARYCARE',
    Type = 'CONVERSATION',
    Settings = {
        'VocabularyName': 'example-med-custom-vocab'
    }
)

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

## 의료 맞춤 어휘를 사용하여 실시간 스트림 자막 작성하기

실시간 스트림에서 트랜스크립션 정확도를 높이려면 HTTP/2 또는 WebSocket 스트림을 사용하여 사용자 지정 어휘를 사용할 수 있습니다. HTTP/2 요청을 시작하려면 [StartMedicalStreamTranscription](#) API를 사용하세요.

[StartMedicalStreamTranscription](#) API를 사용하거나 WebSocket 프로토콜을 사용하여 사용자 지정 어휘를 실시간으로 사용할 수 있습니다. AWS Management Console

음성으로 들리는 받아쓰기를 마이크에 옮기기 (AWS Management Console)

를 사용하여 의료 받아쓰기의 스트리밍 오디오를 변환하려면 의료 받아쓰기를 변환하고 스트림을 시작한 다음 마이크에 대고 말하기 시작하는 옵션을 선택합니다. AWS Management Console

의료 받아쓰기의 스트리밍 오디오를 녹음하려면 (AWS Management Console)

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창의 Amazon Transcribe Medical에서 실시간 트랜스크립션을 선택합니다.
3. 의료 전문 분야의 경우 스트림에서 말하는 임상 의의 의료 전문 분야를 선택하십시오.
4. 오디오 입력 유형에서 대화 또는 받아쓰기를 선택합니다.
5. 추가 설정에서 사용자 지정 어휘를 선택합니다.
  - 어휘 선택에서 사용자 지정 어휘를 선택합니다.
6. 스트리밍 시작을 선택합니다.
7. 마이크에 대고 말하세요.

HTTP/2 스트림의 스피커 파티셔닝 활성화 스트림에서 스피커 파티셔닝을 활성화합니다.

다음은 HTTP/2 요청의 파라미터 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
```

## 매개변수 설명:

- **host:** 호출 중인 이름으로 AWS 리전 (위 예제의 'us-west-2') AWS 리전 를 업데이트합니다. 유효한 AWS 리전 목록은 [AWS 리전 및 엔드포인트](#)를 참조하십시오.
- **승인:** 생성된 필드입니다. 서명을 만드는 방법에 대해 자세히 알아보려면 서명 [버전 4를 사용한 서명 AWS 요청](#)을 참조하십시오.
- **x-amz-target:** 이 필드를 변경하지 말고 위 예제에 표시된 콘텐츠를 사용하십시오.
- **x-amz-content-sha256:** 생성된 필드입니다. 서명 계산에 대해 자세히 알아보려면 서명 [버전 4를 사용한 서명 AWS 요청](#)을 참조하십시오.
- **x-amz-date:** 서명이 생성된 날짜 및 시간입니다. 형식은 YYYYMMDDTHHMMSSZ이며, 여기서 YYYY=년, MM=월, DD=일, HH=시간, MM=분, SS=초, 'T'와 'Z'는 고정 문자입니다. 자세한 정보는 [서명 버전 4의 날짜 처리](#) 를 참조하세요.
- **x-amzn-transcribe-session-id:** 스트리밍 세션의 이름입니다.
- **x-amzn-transcribe-language-code:** 입력 오디오에 사용되는 인코딩입니다. 유효한 값 목록은 [StartMedicalStreamTranscription](#) 또는 [지원되는 언어 및 언어별 기능](#) 를 참조하십시오.
- **x-amzn-transcribe-media-인코딩:** 입력 오디오에 사용되는 인코딩입니다. 유효한 값은 pcm, ogg-opus, flac입니다.
- **x-amzn-transcribe-sample-rate:** 입력 오디오의 샘플 레이트 (Hertz 단위). Amazon Transcribe 8,000Hz ~ 48,000Hz 범위를 지원합니다. 전화 오디오와 같은 저품질 오디오는 일반적으로 약 8,000Hz입니다. 고품질 오디오의 범위는 일반적으로 16,000Hz ~ 48,000Hz입니다. 지정한 샘플 속도는 오디오의 샘플 속도와 일치해야 한다는 점에 유의하세요.
- **x-amzn-transcribe-vocabulary-이름:** 필사본에 사용하려는 어휘의 이름입니다.
- **x-amzn-transcribe-specialty:** 의료 전문 분야를 필사 중입니다.
- **x-amzn-transcribe-type:** 받아쓰기인지 대화인지 선택합니다.
- **x-amzn-transcribe-show-스피커 레이블:** 다이얼라이제이션을 활성화하려면 이 값이 이어야 합니다 `true`.
- **content-type:** 이 필드를 변경하지 말고 위 예제에 표시된 콘텐츠를 사용하십시오.

## WebSocket요청에서 스피커 파티셔닝 활성화

API를 사용하여 WebSocket 스트림의 스피커를 분할하려면 다음 형식을 사용하여 미리 서명된 URI를 만들어 WebSocket 요청을 시작하고 사용자 지정 어휘의 이름으로 설정하십시오 `vocabulary-name`.

```
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 Transcribe의료용 문자 세트

Amazon Transcribe의학에서 사용자 지정 어휘를 사용하려면 다음 문자 세트를 사용하십시오.

영어 문자 문자 문자 문자 문자 문자

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
ɫ	006C 0329	ɪ	026A
m	006D	ɹ	0279
n	006E	ʃ	0283
ŋ	006E 0329	ʊ	028A
ou	006F 028A	ʌ	028C
p	0070	ɹ	028D
s	0073	ʒ	0292
t	0074	ɔʒ	02A4

문자	코드	문자	코드
u	0075	ㅈ	02A7
v	0076	ㅊ	03B8

## 필사본에서 개인 건강 정보 (PHI) 를 식별하기

개인 Health 정보 식별을 사용하여 트랜스크립션 결과에 개인 건강 정보 (PHI) 를 표시하십시오. 라벨을 검토하면 환자를 식별하는 데 사용할 수 있는 PHI를 찾을 수 있습니다.

실시간 스트림 또는 일괄 트랜스크립션 작업을 사용하여 PHI를 식별할 수 있습니다.

자체 포스트 프로세싱을 사용하여 트랜스크립션 출력에서 식별된 PHI를 수정할 수 있습니다.

개인 Health 정보 식별을 사용하여 다음 유형의 PHI를 식별하십시오.

- 개인 건강 정보 양도 PHI:
  - 이름 — 전체 이름 또는 성 및 이니셜
  - 성별
  - 나이
  - 전화 번호
  - 환자와 직접 관련된 날짜 (연도 제외)
  - 이메일 주소
- 지리적 PHI:
  - 물리적 주소
  - 우편번호
  - 의료 센터 또는 진료소 이름
- 계정 PHI:
  - 팩스 번호
  - 소셜 시큐리티 번호 (SSN)
  - 건강보험 수혜자 번호
  - Account Numb
  - 인증서 또는 라이선스 번호

- 차량 PHI:
  - 차량 식별 번호(VIN)
  - 자동차 번호판 번호
- 기타 PHI:
  - 웹 유니폼 리소스 위치 (URL)
  - 인터넷 프로토콜 (IP) 주소 번호

Amazon Transcribe HIPAA (미국 Health 보험 양도 및 책임에 관한 법) 는 HIPAA (미국 건강 보험 양도 및 책임에 관한 법) 에 관한 법 자세한 정보는 [Amazon Transcribe 의료](#) 을 참조하세요. 오디오 파일에서 PHI 식별에 대한 자세한 내용은 을 참조하십시오 [오디오 파일에서 PHI 식별하기](#). 스트림에서 PHI 식별에 대한 자세한 내용은 을 참조하십시오 [실시간 스트림에서 PHI 식별](#).

주제

- [오디오 파일에서 PHI 식별하기](#)
- [실시간 스트림에서 PHI 식별](#)

## 오디오 파일에서 PHI 식별하기

일괄 트랜스크립션 작업을 사용하여 오디오 파일을 트랜스크립션하고 그 안에 있는 개인 건강 정보 (PHI) 를 식별하세요. 개인 Health 정보 (PHI) 식별을 활성화하면 Amazon Transcribe 의료 기관은 필사 결과에서 식별한 PHI에 라벨을 붙입니다. Amazon Transcribe 의료 부서에서 식별할 수 있는 PHI에 대한 자세한 내용은 을 참조하십시오 [필사본에서 개인 건강 정보 \(PHI\) 를 식별하기](#).

[StartMedicalTranscriptionJob](#) API 또는 를 사용하여 일괄 트랜스크립션 작업을 시작할 수 AWS Management Console 있습니다.

### AWS Management Console

를 사용하여 임상 의와 환자의 대화를 AWS Management Console 필사하려면 트랜스크립션 작업을 생성하고 [오디오 입력 유형용 대화] 를 선택합니다.

오디오 파일을 변환하고 PHI 식별하기 (AWS Management Console)

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창의 [Amazon Transcribe Medical]에서 트랜스크립션 작업을 선택합니다.
3. 작업 생성(Create job)을 선택합니다.

4. Job 세부 정보 지정 페이지의 작업 설정에서 다음을 지정합니다.
  - a. 이름 — 고유한 트랜스크립션 작업의 이름입니다AWS 계정.
  - b. 오디오 입력 유형 — 대화 또는 받아쓰기.
5. 나머지 필드에는 오디오 파일의Amazon S3 위치와 트랜스크립션 작업의 출력을 저장할 위치를 지정합니다.
6. Next(다음)를 선택합니다.
7. 오디오 설정에서 PHI 식별을 선택합니다.
8. Create(생성)를 선택합니다.

## API

일괄 트랜스크립션 작업 (API) 을 사용하여 오디오 파일을 변환하고 PHI 식별하기

- [StartMedicalTranscriptionJob](#)API의 경우 다음을 지정합니다.
  - a. 의MedicalTranscriptionJobName 경우 고유한 이름을 지정하십시오AWS 계정.
  - b. 의LanguageCode 경우 오디오 파일에서 사용되는 언어에 해당하는 언어 코드를 지정하십시오.
  - c. Media개체의MediaFileUri 매개 변수로 변환하려는 오디오 파일의 이름을 지정합니다.
  - d. 의Specialty 경우 오디오 파일에서 말하는 임상어의 전문 의료 분야를 다음과 같이 지정하십시오PRIMARYCARE.
  - e. Type에 CONVERSATION 또는 DICTATION를 지정합니다.
  - f. 의OutputBucketName 경우 트랜스크립션 결과를 저장할Amazon S3 버킷을 지정합니다.

다음은 를 사용하여 오디오 파일을 변환하고 환자의 PHI를 식별하는 예제 요청입니다.AWS SDK for Python (Boto3)

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



```

    Media = {'MediaFileUri': job_uri},
    LanguageCode = 'en-US',
    ContentIdentificationType = 'PHI',
    Specialty = 'PRIMARYCARE',
    Type = 'type', # Specify 'CONVERSATION' for a medical conversation. Specify
    'DICTATION' for a medical dictation.
    OutputBucketName = 'DOC-EXAMPLE-BUCKET'
)
while True:
    status = transcribe.get_medical_transcription_job(MedicalTranscriptionJobName =
job_name)
    if status['MedicalTranscriptionJob']['TranscriptionJobStatus'] in ['COMPLETED',
'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
print(status)

```

다음 예제 코드는 식별된 환자 PHI를 사용한 전사 결과를 보여줍니다.

```

{
  "jobName": "my-medical-transcription-job-name",
  "accountId": "111122223333",
  "results": {
    "transcripts": [{
      "transcript": "The patient's name is Bertrand."
    }],
    "items": [{
      "start_time": "0.0",
      "end_time": "0.37",
      "alternatives": [{
        "confidence": "0.9993",
        "content": "The"
      }],
      "type": "pronunciation"
    }, {
      "start_time": "0.37",
      "end_time": "0.44",
      "alternatives": [{
        "confidence": "0.9981",

```

```

        "content": "patient's"
    }],
    "type": "pronunciation"
}, {
    "start_time": "0.44",
    "end_time": "0.52",
    "alternatives": [{
        "confidence": "1.0",
        "content": "name"
    }],
    "type": "pronunciation"
}, {
    "start_time": "0.52",
    "end_time": "0.92",
    "alternatives": [{
        "confidence": "1.0",
        "content": "is"
    }],
    "type": "pronunciation"
}, {
    "start_time": "0.92",
    "end_time": "0.9989",
    "alternatives": [{
        "confidence": "1.0",
        "content": "Bertrand"
    }],
    "type": "pronunciation"
}, {
    "alternatives": [{
        "confidence": "0.0",
        "content": "."
    }],
    "type": "punctuation"
}],
"entities": [{
    "content": "Bertrand",
    "category": "PHI*-Personal*",
    "startTime": 0.92,
    "endTime": 1.2,
    "confidence": 0.9989
}],
},
"status": "COMPLETED"
}

```

## AWS CLI

일괄 트랜스크립션 작업 (AWS CLI) 을 사용하여 오디오 파일을 변환하고 PHI 식별하기

- 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://DOC-EXAMPLE-BUCKET/my-input-files/my-audio-file.flac" \
--output-bucket-name DOC-EXAMPLE-BUCKET \
--specialty PRIMARYCARE \
--type type \ # Choose CONVERSATION to transcribe a medical conversation.
               Choose DICTATION to transcribe a medical dictation.
--content-identification-type PHI
```

## 실시간 스트림에서 PHI 식별

HTTP/2 또는 WebSocket 스트림에서 개인 Health 정보 (PHI) 를 식별할 수 있습니다. PHI 식별을 활성화하면 Amazon Transcribe 의료 기관은 전사 결과에서 식별하는 PHI에 라벨을 붙입니다. Amazon Transcribe 의료 부서에서 식별할 수 있는 PHI에 대한 자세한 내용은 [필사본에서 개인 건강 정보 \(PHI\) 를 식별하기](#).

마이크에 대고 말하는 받아쓰기에서 PHI 식별하기

를 사용하여 마이크에 포착된 음성을 녹음하고 PHI를 AWS Management Console 식별하려면 오디오 입력 유형으로 받아쓰기를 선택하고 스트림을 시작한 다음 컴퓨터의 마이크에 대고 말하기 시작합니다.

를 사용하여 받아쓰기에서 PHI를 식별하려면 AWS Management Console

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창에서 실시간 트랜스크립션을 선택합니다.
3. 오디오 입력 유형에서 받아쓰기를 선택합니다.

4. 추가 설정에서 PHI ID를 선택합니다.
5. 스트리밍 시작을 선택하고 마이크에 대고 말하세요.
6. 받아쓰기를 종료하려면 스트리밍 중지를 선택합니다.

## HTTP/2 스트림에서 PHI 식별하기

PHI 식별이 활성화된 상태에서 HTTP/2 스트림을 시작하려면

[StartMedicalStreamTranscription](#) API를 사용하고 다음을 지정합니다.

- 의LanguageCode 경우 스트림에서 사용되는 언어의 언어 코드를 지정합니다. 미국 영어의 경우 지정하십시오 en-US.
- 의MediaSampleHertz 경우 오디오의 샘플 속도를 지정합니다.
- content-identification-type에 PHI을 지정합니다.

## WebSocket 스트림에서 PHI 식별하기

PHI 식별이 활성화된 상태로 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
```

매개변수 [정의는 API 참조에서](#) 찾을 수 있습니다. 모든 AWS API 작업에 공통적인 [매개변수는 일반 매개변수](#) 섹션에 나열되어 있습니다.

## 대체 트랜스크립션 생성

Amazon TranscribeMedical을 사용하면 신뢰도가 가장 높은 트랜스크립션을 얻을 수 있습니다. 하지만 신뢰도가 낮은 추가 트랜스크립션을 반환하도록 Amazon Transcribe 메디컬을 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 트랜스크립션을 AWS Management Console 생성하려면 작업을 구성할 때 대체 결과를 활성화해야 합니다.

1. [AWS Management Console](#)에 로그인합니다.
2. 탐색 창의 [LOAD] 에서 [Amazon Transcribe TarcALANCING] 를 선택합니다.
3. 작업 생성(Create job)을 선택합니다.
4. 작업 세부 정보 지정 페이지에서 트랜스크립션 작업에 대한 정보를 제공합니다.
5. Next(다음)를 선택합니다.
6. 대체 결과를 활성화합니다.
7. 최대 대안 항목에 2에서 10 사이의 정수 값을 입력하여 출력물에 포함시키려는 대체 트랜스크립션의 최대 개수를 입력합니다.
8. Create(생성)를 선택합니다.

### API

일괄 트랜스크립션 작업 (API) 을 사용하여 오디오 파일에서 스피커당 텍스트를 분리하려면

- [StartMedicalTranscriptionJob](#) API의 경우 다음을 지정합니다.
  - a. 의 경우 `MedicalTranscriptionJobName`, 에 고유한 이름을 지정하십시오 AWS 계정.
  - b. 의 경우 `LanguageCode` 오디오 파일에서 사용되는 언어 및 어휘 필터의 언어에 해당하는 언어 코드를 지정하십시오.
  - c. `Media` 개체의 `MediaFileUri` 매개 변수에서 변환하려는 오디오 파일의 위치를 지정합니다.

- d. 의Specialty 경우 오디오 파일에 말하는 임상어의 의료 전문 분야를 지정하십시오.
- e. 의Type 경우 의료 대화를 필사할지 아니면 받아쓰기를 원하는지 지정하세요.
- f. 의OutputBucketName 경우 트랜스크립션 결과를 저장할 Amazon S3 버킷을 지정합니다.
- g. Settings 객체에 대해 다음을 지정합니다.
  - i. ShowAlternatives - true.
  - ii. MaxAlternatives- 트랜스크립션 출력에서 원하는 대체 트랜스크립션의 수를 나타내는 2에서 10 사이의 정수입니다.

다음 요청에서는 AWS SDK for Python (Boto3) 를 사용하여 최대 두 개의 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://DOC-EXAMPLE-BUCKET/my-input-files/my-audio-file.flac
transcribe.start_medical_transcription_job(
    MedicalTranscriptionJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'DOC-EXAMPLE-BUCKET',
    OutputKey = 'my-output-files/',
    LanguageCode = 'en-US',
    Specialty = 'PRIMARYCARE',
    Type = 'CONVERSATION',
    Settings = {
        'ShowAlternatives': True,
        'MaxAlternatives': 2
    }
)

while True:
    status = transcribe.get_medical_transcription_job(MedicalTranscriptionJobName =
job_name)
    if status['MedicalTranscriptionJob']['TranscriptionJobStatus'] in ['COMPLETED',
'FAILED']:
```

```
break
print("Not ready yet...")
time.sleep(5)
print(status)
```

## AWS CLI

오디오 파일 (AWS CLI) 에 있는 1차 진료 임상 의와 환자 간의 대화의 오디오 파일을 녹음하려면

- 다음 코드를 실행합니다.

```
aws transcribe start-transcription-job \  
--cli-input-json file://filepath/example-start-command.json
```

다음 코드는 의 콘텐츠를 보여줍니다 `example-start-command.json`.

```
{  
  "MedicalTranscriptionJobName": "my-first-transcription-job",  
  "LanguageCode": "en-US",  
  "Specialty": "PRIMARYCARE",  
  "Type": "CONVERSATION",  
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",  
  "Media": {  
    "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-audio-  
file.flac"  
  },  
  "Settings": {  
    "ShowAlternatives": true,  
    "MaxAlternatives": 2  
  }  
}
```

# Amazon Transcribe의료 및 인터페이스 VPC 엔드포인트 (AWS PrivateLink)

VPC와 VPC 간에 프라이빗 연결을 설정할 수 있습니다. Amazon Transcribe를 생성하여 의료 인터페이스 VPC 엔드포인트. 인터페이스 엔드포인트는 다음에 의해 구동됩니다. [AWS PrivateLink](#), 비공개로 액세스할 수 있게 해주는 기술 Amazon Transcribe 인터넷 게이트웨이, NAT 장치, VPN 연결이 없는 의료 API 또는 AWS Direct Connect 연결. VPC의 인스턴스는 통신하기 위해 퍼블릭 IP 주소가 필요하지 않습니다. Amazon Transcribe 의료 API. VPC와 VPC 간의 트래픽 Amazon Transcribe 메디컬은 아마존 네트워크를 떠나지 않습니다.

각 인터페이스 엔드포인트는 서브넷에서 하나 이상의 [탄력적 네트워크 인터페이스](#)로 표현됩니다.

자세한 내용은 을 참조하십시오. [인터페이스: VPC 엔드포인트 \(AWS PrivateLink\)](#)에서 Amazon VPC 사용자 가이드.

## 에 대한 고려 사항 Amazon Transcribe 의료용 VPC 엔드포인트

인터페이스 VPC 엔드포인트를 설정하기 전 Amazon Transcribe 의료, 반드시 검토하십시오 [인터페이스 엔드포인트 속성 및 제한](#)에서 Amazon VPC 사용자 가이드.

Amazon Transcribe 메디컬은 VPC에서 모든 API 작업을 호출할 수 있도록 지원합니다.

## 에 대한 인터페이스 VPC 엔드포인트 생성 Amazon Transcribe 의료

에 대한 VPC 엔드포인트를 생성할 수 있습니다. Amazon Transcribe 다음 중 하나를 이용한 의료 서비스 AWS Management Console 또는 AWS CLI. 자세한 내용은 을 참조하십시오. [인터페이스 엔드포인트 생성](#)에서 Amazon VPC 사용자 가이드.

에서 일괄 트랜스크립션용 Amazon Transcribe 의료, 다음 서비스 이름을 사용하여 VPC 엔드포인트를 생성하십시오.

- com.amazonaws.## ## 2. 고쳐 쓰다

내 스트리밍 트랜스크립션용 Amazon Transcribe 의료, 다음 서비스 이름을 사용하여 VPC 엔드포인트를 생성하십시오.

- com.amazonaws.## ## 2. 트랜스크립트 스트리밍



엔드포인트에 프라이빗 DNS를 활성화하면 다음 주소로 API 요청을 보낼 수 있습니다. Amazon Transcribe 기본 DNS 이름을 사용하는 의료 AWS 리전, 예를 들면, `transcribestreaming.us-east-2.amazonaws.com`.

자세한 내용은 을 참조하십시오. [인터페이스 엔드포인트를 통한 서비스 액세스](#)에서 Amazon VPC 사용자 가이드.

## 다음에 대한 VPC 엔드포인트 정책 생성 Amazon Transcribe 의료 스트리밍

VPC 엔드포인트에 대한 액세스를 제어하는 엔드포인트 정책을 연결할 수 있습니다. Amazon Transcribe 의료. 이 정책은 다음 정보를 지정합니다.

- 태스크를 수행할 수 있는 보안 주체.
- 수행할 수 있는 작업입니다.
- 태스크를 수행할 있는 리소스.

자세한 정보는 Amazon VPC 사용 설명서의 [VPC 엔드포인트를 통해 서비스에 대한 액세스 제어](#)를 참조하십시오.

예: 다음에 대한 VPC 엔드포인트 정책 Amazon Transcribe 의료 스트리밍 트랜스크립션 작업

다음은 스트리밍 트랜스크립션을 위한 엔드포인트 정책의 예입니다. Amazon Transcribe 의료. 엔드포인트에 연결되면 이 정책은 목록에 있는 항목에 대한 액세스 권한을 부여합니다. Amazon Transcribe 모든 자원에 대한 모든 교장에 대한 의료 조치.

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Allow",
      "Action": [
        "transcribe:StartMedicalStreamTranscription",
      ],
      "Resource": "*"
    }
  ]
}
```

예: 다음에 대한 VPC 엔드포인트 정책 Amazon Transcribe 의료용 배치 트랜스크립션 작업

다음은 에서 일괄 트랜스크립션을 위한 엔드포인트 정책의 예입니다. Amazon Transcribe의 엔드포인트에 연결되면 이 정책은 목록에 있는 항목에 대한 액세스 권한을 부여합니다. Amazon Transcribe 모든 자원에 대한 모든 교장에 대한 의료 조치.

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Allow",
      "Action": [
        "transcribe:StartMedicalTranscriptionJob"
      ],
      "Resource": "*"
    }
  ]
}
```

## 공유 서브넷

공유되는 서브넷에서는 VPC 엔드포인트를 생성, 설명, 수정 또는 삭제할 수 없습니다. 그러나 공유하는 서브넷의 VPC 엔드포인트를 사용할 수는 있습니다. VPC 공유에 대한 자세한 내용은 [VPC를 다른 계정과 공유](#)에서 Amazon Virtual Private Cloud 안내서.

# AWS HealthScribe

AWS HealthScribe는 음성 인식과 생성형 AI를 결합하여 환자와 의료진의 대화를 트랜스크립션하고 검토하기 쉬운 임상 노트를 생성하는 새로운 HIPAA 적격 기계 학습(ML) 기능입니다. AWS HealthScribe를 사용하면 의료 소프트웨어 공급업체가 문서화 부담을 줄이고 상담 경험을 개선하는 임상 애플리케이션을 구축할 수 있습니다. 이 서비스는 풍부한 대화 트랜스크립트를 자동으로 제공하고, 화자 역할을 식별하고, 대화를 분류하고, 의학 용어를 추출하고, 예비 임상 노트를 생성합니다. AWS 이러한 기능이 결합된 HealthScribe는 별도의 AI 서비스를 통합하고 최적화할 필요가 없으므로 이행을 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 오디오 품질, 배경 소음, 화자 명료성, 의학 용어의 복잡성, 컨텍스트별 언어 뉘앙스, [기계 학습 및 생성형 AI의 특성](#) 등 다양한 요인으로 인해 항상 정확하지 않을 수 있습니다. AWS HealthScribe는 의사와 의료 서기의 보조 역할로 사용하도록 설계되었습니다. AWS HealthScribe 출력은 숙련된 의료 전문가가 정확성과 올바른 의학적 판단 여부를 검토한 후 전자 의료 기록을 포함하되 이에 국한되지 않는 환자 진료 시나리오에서만 사용해야 합니다. AWS HealthScribe 출력은 전문적인 의학적 조언, 진단 또는 치료를 대체하지 않으며 질병이나 건강 상태를 치료, 완화, 예방 또는 진단하기 위한 것도 아닙니다.

AWS HealthScribe는 공동 책임 모델에 따라 운영됩니다. 이에 따라 AWS는 AWS HealthScribe를 실행하는 인프라를 보호할 책임이 있으며 사용자는 데이터를 관리할 책임이 있습니다. 자세한 내용은 [공동 책임 모델](#)을 참조하세요.

AWS HealthScribe는 미국 동부(버지니아 북부) 리전에서 사용 가능합니다.

이 서비스는 미국 영어(en-US)로 제공됩니다. 최상의 결과를 얻으려면 PCM 16비트로 인코딩된 무손실 오디오 형식(예: FLAC 또는 WAV)을 사용해야 합니다. AWS HealthScribe는 16,000Hz 이상의 샘플링 속도를 지원합니다.

AWS HealthScribe는 현재 일반 의학 및 정형외과 전문 분야를 지원합니다.

AWS HealthScribe 작업은 의료 상담을 분석하여 [트랜스크립트](#) 파일과 [임상 문서](#)라는 두 개의 JSON 출력 파일을 생성합니다.

AWS HealthScribe는 트랜스크립트 파일에서 워드 레벨 타임스탬프가 포함된 표준 턴바이턴 트랜스크립션 출력 외에도 다음과 같은 기능을 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.

AWS HealthScribe는 임상 문서 파일에서 다음을 제공합니다.

- 요약: 주요 불편 사항, 현재 질병 이력, 신체 증상, 과거 병력, 평가 및 계획 등 임상 문서의 주요 섹션에 대한 요약된 메모가 포함됩니다.
- 증거 링크: AI 생성형 요약에 사용된 모든 문장을 원본 상담 트랜스크립트에 연결하므로 사용자가 해당 애플리케이션 내 요약의 정확성을 쉽게 검증할 수 있습니다.

AWS HealthScribe 관련 API 작업:

- StartMedicalScribeJob
- ListMedicalScribeJobs
- GetMedicalScribeJob
- DeleteMedicalScribeJob

예시 AWS HealthScribe 요청을 보려면 [AWS HealthScribe 작업 시작](#)을 참조하세요.

## 트랜스크립트 파일

트랜스크립트 파일은 대화 내용을 턴바이턴 형식으로 제공합니다.

또한 각 대화 턴에 대해 다음과 같은 인사이트가 제공됩니다.

- **참가자 역할** — 각 참가자는 임상 의 또는 환자로 분류됩니다. 대화에 범주별로 한 명 이상의 참가자가 있는 경우 각 참가자에게 번호가 할당됩니다. 예를 들면 CLINICIAN\_1, CLINICIAN\_2 및 PATIENT\_1, PATIENT\_2입니다.
- **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** - 대화에서 임상적 관련성이 있는 엔터티(ClinicalEntity)를 추출합니다. AWS HealthScribe는 [Amazon Comprehend Medical](#)에서 지원하는 모든 임상 엔터티를 감지합니다.

자세한 출력 정보는 [트랜스크립트 출력 예시](#)를 참조하세요.

## 임상 문서 파일

문서 인사이트 파일에는 임상 문서의 다음 주요 섹션에 대한 요약이 포함되어 있습니다.

섹션	Description
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.
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 임상 의의 평가 정보를 제공하는 노트.

섹션	Description
PLAN	모든 의학적 치료, 생활 습관 조정 및 추가 진료 예약을 언급하는 메모입니다.

Summary에 있는 모든 문장에는 원본 상담 트랜스크립트에 대한 참조가 포함되어 있어 사용자가 해당 애플리케이션 내 요약의 정확성을 쉽게 검증할 수 있습니다. AI 생성형 인사이트에 추적성과 투명성을 제공하는 것은 Responsible AI원칙(예: 설명 가능성)과 일치합니다. 이러한 참고 자료를 요약 노트와 함께 임상이나 의료진에게 제공하면 신뢰를 높이고 임상 환경에서 AI를 안전하게 사용하도록 장려하는데 도움이 됩니다.

Summary의 모든 문장에는 요약된 트랜스크립트의 관련 대화 내용에 대한 SegmentId를 제공하는 EvidenceLinks가 함께 제공됩니다.

자세한 출력 정보는 [임상 문서 출력 예시](#)를 참조하세요.

## AWS HealthScribe 작업 시작

AWS CLI 또는 AWS SDK를 사용하여 AWS HealthScribe 작업을 시작할 수 있습니다. 예시는 다음을 참조하십시오.

### 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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac \
  --output-bucket-name DOC-EXAMPLE-BUCKET \
  --DataAccessRoleArn=arn:aws:iam::111122223333:role/ExampleRole \
  --settings ShowSpeakerLabels=false,ChannelIdentification=true \
  --channel-definitions ChannelId=0,ParticipantRole=CLINICIAN
  ChannelId=1,ParticipantRole=PATIENT
```

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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
  },
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",
  "DataAccessRoleArn": "arn:aws:iam::111122223333:role/ExampleRole",
  "Settings": {
    "ShowSpeakerLabels": false,
    "ChannelIdentification": true
  },
  "ChannelDefinitions": [
    {
      "ChannelId": 0,
      "ParticipantRole": "CLINICIAN"
    }, {
      "ChannelId": 1,
      "ParticipantRole": "PATIENT"
    }
  ]
}
```

## AWS SDK for Python (Boto3)

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 boto3
transcribe = boto3.client('transcribe', 'us-west-2')
job_name = "my-first-medical-scribe-job"
job_uri = "s3://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
```

```

transcribe.start_medical_scribe_job(
    MedicalScribeJobName = job_name,
    Media = {
        'MediaFileUri': job_uri
    },
    OutputBucketName = 'DOC-EXAMPLE-BUCKET',
    DataAccessRoleArn = 'arn:aws:iam::111122223333:role/ExampleRole',
    Settings = {
        'ShowSpeakerLabels': false,
        'ChannelIdentification': true
    },
    ChannelDefinitions = [
        {
            'ChannelId': 0,
            'ParticipantRole': 'CLINICIAN'
        }, {
            'ChannelId': 1,
            'ParticipantRole': 'PATIENT'
        }
    ]
)
while True:
    status = transcribe.get_medical_scribe_job(MedicalScribeJobName = job_name)
    if status['MedicalScribeJob']['MedicalScribeJobStatus'] in ['COMPLETED', 'FAILED']:
        break
    print("Not ready yet...")
    time.sleep(5)
print(status)

```

### Note

AWS 관리 콘솔은 현재 AWS HealthScribe 작업을 지원하지 않습니다.

## 출력 예시

StartMedicalScribeJob 요청은 트랜스크립트 외에도 별도의 음성 문서 파일을 생성합니다. 두 파일 모두 JSON 형식이며 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,
"EndAudioTime": 0,
"Type": "PUNCTUATION"
},
{
    "Alternatives": [
        {
            "Confidence": 1,
            "Content": "Good"
        }
    ],
    "BeginAudioTime": 0.61,
    "EndAudioTime": 0.92,
    "Type": "PRONUNCIATION"
},
{
    "Alternatives": [
        {
            "Confidence": 1,
            "Content": "afternoon"
        }
    ],
    "BeginAudioTime": 0.92,
    "EndAudioTime": 1.54,
    "Type": "PRONUNCIATION"
},
{
    "Alternatives": [
        {
            "Confidence": 0,
            "Content": "."
        }
    ],
    "BeginAudioTime": 0,
    "EndAudioTime": 0,
    "Type": "PUNCTUATION"
},
{
    "Alternatives": [
        {

```

```
        "Confidence": 0.9924,
        "Content": "You"
    }
],
"BeginAudioTime": 1.55,
"EndAudioTime": 1.88,
"Type": "PRONUNCIATION"
},
{
  "Alternatives": [
    {
      "Confidence": 1,
      "Content": "lost"
    }
  ],
  "BeginAudioTime": 1.88,
  "EndAudioTime": 2.19,
  "Type": "PRONUNCIATION"
},
{
  "Alternatives": [
    {
      "Confidence": 1,
      "Content": "one"
    }
  ],
  "BeginAudioTime": 2.19,
  "EndAudioTime": 2.4,
  "Type": "PRONUNCIATION"
},
{
  "Alternatives": [
    {
      "Confidence": 1,
      "Content": "lb"
    }
  ],
  "BeginAudioTime": 2.4,
  "EndAudioTime": 2.97,
  "Type": "PRONUNCIATION"
}
],
"TranscriptSegments": [
  {
```

```
"BeginAudioTime": 0.16,
"Content": "Okay.",
"EndAudioTime": 0.6,
"ParticipantDetails": {
  "ParticipantRole": "CLINICIAN_0"
},
"SectionDetails": {
  "SectionName": "SUBJECTIVE"
},
"SegmentId": "uuid1"
},
{
  "BeginAudioTime": 0.61,
"Content": "Good afternoon.",
"EndAudioTime": 1.54,
"ParticipantDetails": {
  "ParticipantRole": "CLINICIAN_0"
},
"SectionDetails": {
  "SectionName": "OTHER"
},
"SegmentId": "uuid2"
},
{
  "BeginAudioTime": 1.55,
"Content": "You lost one lb.",
"EndAudioTime": 2.97,
"ParticipantDetails": {
  "ParticipantRole": "CLINICIAN_0"
},
"SectionDetails": {
  "SectionName": "SUBJECTIVE"
},
"SegmentId": "uuid3"
},
{
  "BeginAudioTime": 2.98,
"Content": "Yeah, I think it, uh, do you feel more energy?",
"EndAudioTime": 6.95,
"ParticipantDetails": {
  "ParticipantRole": "CLINICIAN_0"
},
"SectionDetails": {
  "SectionName": "SUBJECTIVE"
```

```
    },
    "SegmentId": "uuid5"
  },
  {
    "BeginAudioTime": 6.96,
    "Content": "Yes.",
    "EndAudioTime": 7.88,
    "ParticipantDetails": {
      "ParticipantRole": "CLINICIAN_0"
    },
    "SectionDetails": {
      "SectionName": "SUBJECTIVE"
    },
    "SegmentId": "uuid6"
  },
  {
    "BeginAudioTime": 7.89,
    "Content": "Uh, how about craving for the carbohydrate or sugar or fat or
anything?",
    "EndAudioTime": 17.93,
    "ParticipantDetails": {
      "ParticipantRole": "CLINICIAN_0"
    },
    "SectionDetails": {
      "SectionName": "SUBJECTIVE"
    },
    "SegmentId": "uuid7"
  }
]
}
```

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://DOC-EXAMPLE-BUCKET/my-input-files/my-media-file.flac"
  },
  "OutputBucketName": "DOC-EXAMPLE-BUCKET",
  "DataAccessRoleArn": "arn:aws:iam::111122223333:role/ExampleRole",
  "Settings": {
    "ShowSpeakerLabels": false,
    "ChannelIdentification": true
  },
  "ChannelDefinitions": [
    {
      "ChannelId": 0,
      "ParticipantRole": "CLINICIAN"
    }, {
      "ChannelId": 1,
      "ParticipantRole": "PATIENT"
    }
  ]
}
```

## 임상 문서 출력 예시

StartMedicalScribeJob 요청에서 비롯된 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"
        }
      ],

```



- 키 암호화 자료 교체
- 태그 추가
- 키 별칭 생성
- 키 삭제 일정 수립

자세한 내용은 AWS Key Management Service 개발자 안내서의 [고객 관리형 키](#)를 참조하세요.

#### Note

AWS HealthScribe는 AWS 소유 키를 사용해 저장 데이터를 자동으로 암호화하여 개인 식별 데이터를 무료로 보호할 수 있습니다. 그러나 고객 관리형 키 사용에는 AWS KMS 비용이 부과됩니다. 요금에 대한 자세한 내용은 [AWS Key Management Service 요금](#)을 참조하세요. AWS KMS에 대한 자세한 내용은 [AWS Key Management Service 소개](#) 단원을 참조하십시오.

## 고객 관리형 키를 생성하려면

AWS Management Console 또는 AWS KMS API를 사용하여 대칭형 고객 관리형 키를 생성할 수 있습니다. 대칭 고객 관리형 키를 생성하려면 AWS Key Management Service 개발자 안내서의 [대칭 고객 관리형 키 생성](#) 단계를 따르세요.

키 정책은 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 생성할 때 키 정책을 지정할 수 있습니다. 자세한 내용은 AWS Key Management Service 개발자 안내서의 [고객 관리형 키에 대한 액세스 관리](#)를 참조하세요.

[StartMedicalScribeJob](#) 요청에서 [DataAccessRoleArn](#)으로 지정한 IAM 역할과 동일한 계정의 키를 사용하는 경우 키 정책을 업데이트할 필요가 없습니다. 다른 계정의 고객 관리형 키를 [DataAccessRole](#)로 사용하려면 다음 작업에 대해 키 정책에서 [DataAccessRoleArn](#)을 신뢰해야 합니다.

- [kms:Encrypt](#) - 고객 관리형 키를 사용한 암호화를 허용합니다.
- [kms:Decrypt](#) - 고객 관리형 키를 사용한 복호화를 허용합니다.
- [kms:DescribeKey](#) - AWS HealthScribe에서 키를 검증할 수 있도록 고객 관리형 키 세부 정보를 제공합니다.

다음은 IAM 역할에 고객 관리형 키를 사용할 수 있는 크로스 계정 권한을 부여하기 위해 추가할 수 있는 정책 문 예시입니다.

```

"Statement" : [
  {
    "Sid": "Allow access to the DataAccessRole for StartMedicalScribeJob",
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::111122223333:role/DataAccessRole"
    },
    "Action": [
      "kms:DescribeKey",
      "kms:Encrypt",
      "kms:Decrypt",
      "kms:GenerateDataKey"
    ],
    "Resource" : "*"
  }
]

```

고객 관리형 키와 DataAccessRole이 동일한 계정에 있는 다른 계정에 있는 관계없이 DataAccessRole에는 고객 관리형 키를 사용하여 위에서 언급한 작업을 수행할 수 있는 권한이 필요합니다. 다음은 DataAccessRole에 추가할 수 있는 정책 문 예시입니다.

```

"Statement" : [
  {
    "Sid": "Allow role to perform AWS KMS actions for customer managed key",
    "Effect": "Allow",
    "Action": [
      "kms:DescribeKey",
      "kms:Encrypt",
      "kms:Decrypt"
    ],
    "Resource": "*"
  }
]

```

[정책에서 권한 지정](#)에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서를 참조하세요. [키 액세스 문제 해결](#)에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서를 참조하세요.

## AWS HealthScribe에 대한 고객 관리형 키 지정

고객 관리형 키를 StartMedicalScribeJob 요청에 대한 두 번째 계층 암호화로 지정할 수 있습니다.

[StartMedicalScribeJob](#) 요청을 생성할 때 요청에 [OutputEncryptionKMSKeyId](#) 필드를 포함하여 고객 관리형 키를 지정할 수 있습니다.

### 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. AWS HealthScribe에서 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 로그에 일반 텍스트로 표시됩니다(따라서 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 암호화 컨텍스트를 제공하려면 [OutputEncryptionKMSKeyId](#) 파라미터가 대칭 AWS KMS 키 ID를 참조해야 합니다.

IAM 정책과 함께 [AWS KMS 조건 키](#)를 사용하여 [암호화 작업](#) 요청의 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.



# 에 대한 문서 기록 Amazon Transcribe

- 최신 문서 업데이트: 2023년 11월 13일

다음 표에는 각 릴리스의 Amazon Transcribe 중요한 변경 사항이 설명되어 있습니다. 이 설명서에 대한 업데이트 알림을 받으려면 RSS 피드를 구독하면 됩니다.

변경 사항	설명	날짜
<a href="#">기능 업데이트</a>	다이어라이제이션의 최대 스피커 수를 10개가 아닌 30개로 업데이트하십시오.	2024년 5월 10일
<a href="#">섹션 업데이트</a>	생성 호출 요약 업데이트 및 오류 출력 세부 정보 추가	2024년 4월 30일
<a href="#">섹션 업데이트</a>	사용자 지정 어휘 컬럼에 대한 업데이트 - IPA 및 SoundsLike	2024년 4월 30일
<a href="#">기능 업데이트</a>	Amazon Transcribe 통화 분석은 이제 생성적 통화 요약을 지원합니다.	2023년 11월 29일
<a href="#">섹션 업데이트</a>	새 PII 수정 및 언어 식별 출력 형식을 업데이트했습니다.	2023년 11월 13일
<a href="#">기능 업데이트</a>	이제 분할을 채널 식별과 결합할 수 있습니다.	2023년 3월 6일
<a href="#">기능 업데이트</a>	이제 채널 식별을 분할과 결합할 수 있습니다.	2023년 3월 6일
<a href="#">섹션 업데이트</a>	IAM 모범 사례가 업데이트되었습니다.	2023년 2월 13일
<a href="#">새로운 언어</a>	Amazon Transcribe 이제 베트남어와 스웨덴어를 지원합니다.	2022년 12월 6일

<a href="#">새 기능</a>	Amazon Transcribe 이제 실시간 통화 분석을 지원합니다.	2022년 11월 28일
<a href="#">기능 업데이트</a>	이제 힌디어와 태국어로 스트리밍 편집 및 식별을 사용할 수 있습니다.	2022년 11월 11일
<a href="#">섹션 업데이트</a>	스트리밍 편집 및 식별을 위해 새 PII 범주를 사용할 수 있습니다.	2022년 9월 14일
<a href="#">섹션 업데이트</a>	사용자 지정 언어 모델 섹션이 수정되었습니다.	2022년 6월 18일
<a href="#">섹션 업데이트</a>	이제 배치 언어 식별을 통해 오디오 파일당 여러 언어를 식별할 수 있습니다.	2022년 5월 31일
<a href="#">설명서 업데이트</a>	Amazon Transcribe API 레퍼런스는 이제 독립형 가이드가 되었습니다.	2022년 4월 1일
<a href="#">새로운 장</a>	Amazon Transcribe 의료 및 Amazon Transcribe 통화 분석에 대한 Amazon Transcribe 새로운 비교표가 포함되었습니다.	2022년 3월 21일
<a href="#">새로운 장</a>	새 SDK 코드 예시 장이 포함되었습니다.	2022년 3월 21일
<a href="#">기능 업데이트</a>	이제 통화 분석에서 통화 요약 을 제공합니다.	2022년 3월 21일
<a href="#">장 업데이트</a>	이제 소개 장에서는 Amazon Transcribe 사용 사례를 보여줍니다.	2022년 3월 21일

<a href="#">장 업데이트</a>	시작하기 장이 메서드별로 업데이트되었습니다.	2022년 3월 21일
<a href="#">장 업데이트</a>	스트리밍 장이 업데이트 및 재구성되었습니다.	2022년 3월 21일
<a href="#">기능 업데이트</a>	이제 언어 식별은 스트리밍 트랜스크립션을 통한 사용자 지정 어휘와 사용자 지정 어휘 필터를 지원합니다.	2022년 3월 11일
<a href="#">새로운 이벤트</a>	새 이벤트 유형인 어휘 이벤트가 있습니다.	2022년 2월 7일
<a href="#">섹션 업데이트</a>	사용자 지정 어휘 섹션이 업데이트되었습니다.	2022년 1월 20일
<a href="#">새 기능</a>	이제 스트리밍 트랜스크립션에 언어 식별을 사용할 수 있습니다.	2021년 11월 23일
<a href="#">새 기능</a>	이제 사용자 지정 언어 모델, 사용자 지정 어휘, 어휘 필터링 및 콘텐츠 편집과 함께 언어 식별을 사용할 수 있습니다.	2021년 10월 29일
<a href="#">새 기능</a>	Amazon Transcribe 이제 스트리밍 트랜스크립션이 포함된 사용자 지정 언어 모델을 지원합니다.	2021년 10월 20일
<a href="#">새 기능</a>	Amazon Transcribe 이제 비디오 파일의 자막을 생성할 수 있습니다.	2021년 9월 16일
<a href="#">새 기능</a>	Amazon Transcribe 이제 스트리밍을 위한 PII 수정 및 식별이 지원됩니다.	2021년 9월 14일

<a href="#">새 기능</a>	Amazon Transcribe 이제 리소스 보안 수준을 높이기 위해 AWS KMS 암호화 컨텍스트를 지원합니다. AWS 계정	2021년 9월 10일
<a href="#">새로운 언어</a>	Amazon Transcribe 이제 아프리칸스어, 덴마크어, 중국어 (번체), 태국어, 뉴질랜드 영어, 남아프리카 영어를 지원합니다.	2021년 8월 26일
<a href="#">새 기능</a>	Amazon Transcribe 이제 리소스 태깅을 지원합니다.	2021년 8월 24일
<a href="#">새 기능</a>	Amazon Transcribe 이제 일괄 트랜스크립션 작업에 대한 콜 애널리틱스를 지원합니다.	2021년 8월 4일
<a href="#">새 기능</a>	Amazon Transcribe 이제 일괄 사용자 지정 언어 모델에서 사용자 지정 어휘를 사용할 수 있습니다.	2021년 5월 12일
<a href="#">새 기능</a>	Amazon Transcribe 이제 스트리밍 트랜스크립션의 부분 결과 안정화를 지원합니다.	2021년 5월 11일
<a href="#">새 기능</a>	Amazon Transcribe 이제 사용자 지정 언어 모델에 호주 영어, 영국 영어, 힌디어, 미국 스페인어를 지원합니다.	2021년 3월 19일
<a href="#">새 기능</a>	Amazon Transcribe 이제 스트리밍 오디오 트랜스크립션을 위한 OGG/OPUS 및 FLAC 코덱을 지원합니다.	2020년 11월 24일

<a href="#">새로운 언어</a>	Amazon Transcribe 스트리밍 오디오 트랜스크립션을 위한 이탈리아어 및 독일어 지원을 추가합니다.	2020년 11월 4일
<a href="#">AWS 리전 확장</a>	Amazon Transcribe 현재 프랑크푸르트 (eu-central-1) 와 런던 (eu-west-2) 에서 사용할 수 있습니다.	2020년 11월 4일
<a href="#">새 기능</a>	Amazon Transcribe 일괄 트랜스크립션에서 인터페이스 VPC 엔드포인트에 대한 지원을 추가합니다.	2020년 10월 9일
<a href="#">새 기능</a>	Amazon Transcribe 스트리밍의 채널 식별에 대한 지원을 추가합니다.	2020년 9월 17일
<a href="#">새 기능</a>	Amazon Transcribe 일괄 트랜스크립션에서 자동 언어 식별에 대한 지원을 추가합니다.	2020년 9월 15일
<a href="#">새 기능</a>	Amazon Transcribe 스트리밍에서 스피커 파티셔닝에 대한 지원을 추가합니다.	2020년 8월 19일
<a href="#">새 기능</a>	Amazon Transcribe 사용자 지정 언어 모델에 대한 지원을 추가합니다.	2020년 8월 5일
<a href="#">새로운 특성</a>	Amazon Transcribe 스트리밍에서 인터페이스 VPC 엔드포인트에 대한 지원을 추가합니다.	2020년 6월 26일

<a href="#">새 기능</a>	Amazon Transcribe 스트리밍의 어휘 필터링 지원을 추가합니다.	2020년 5월 20일
<a href="#">새 기능</a>	Amazon Transcribe 개인 식별이 가능한 정보를 자동으로 삭제하는 지원을 추가합니다.	2020년 2월 26일
<a href="#">새 기능</a>	Amazon Transcribe 트랜스크립션에서 필터링할 사용자 지정 단어 어휘를 만들 수 있는 지원을 추가합니다.	2019년 12월 20일
<a href="#">새 기능</a>	Amazon Transcribe 트랜스크립션 작업 대기열에 대한 지원을 추가합니다.	2019년 12월 19일
<a href="#">새로운 언어</a>	Amazon Transcribe 걸프 아랍어, 히브리어, 일본어, 말레이어, 스위스 독일어, 텔루구어, 터키어에 대한 지원을 추가합니다.	2019년 11월 21일
<a href="#">AWS 리전 확장</a>	Amazon Transcribe 이제 아시아 태평양 (도쿄) 에서 사용할 수 있습니다 (ap-northeast-1).	2019년 11월 21일
<a href="#">새 기능</a>	Amazon Transcribe 대체 트랜스크립션에 대한 지원을 추가합니다.	2019년 11월 20일
<a href="#">새로운 언어</a>	Amazon Transcribe 네덜란드어, 페르시아어, 인도네시아어, 아일랜드어 영어, 포르투갈어, 스코틀랜드 영어, 타밀어, 웨일스어 영어에 대한 지원을 추가합니다.	2019년 11월 12일

<a href="#">새로운 언어</a>	Amazon Transcribe 이제 호주 영어 (en-AU) 의 스트리밍 트랜스크립션을 지원합니다.	2019년 10월 25일
<a href="#">AWS 리전 확장</a>	Amazon Transcribe 현재 중국 (베이징) (cn-north-1) 및 중국 (닝샤) (cn-북서쪽-1) 에서 사용할 수 있습니다.	2019년 10월 9일
<a href="#">새 기능</a>	Amazon Transcribe 트랜스크립션 출력 파일을 암호화하기 위한 자체 파일을 제공할 수 있습니다. KMS key 자세한 내용은 API의 <a href="#">OutputEncryptionKMS KeyId</a> 매개변수를 참조하십시오. <a href="#">StartStreamTranscription</a>	2019년 9월 24일
<a href="#">새로운 언어</a>	Amazon Transcribe 중국어 (북경어), 간체, 중국 본토 및 러시아어에 대한 지원을 추가합니다.	2019년 8월 23일
<a href="#">새 기능</a>	Amazon Transcribe 프로토콜을 사용한 스트리밍 오디오 트랜스크립션에 대한 지원을 추가합니다. WebSocket	2019년 7월 19일
<a href="#">새 기능</a>	AWS CloudTrail 이제 <a href="#">StartStreamTranscription</a> API의 이벤트를 기록합니다.	2019년 7월 19일
<a href="#">AWS 리전 확장</a>	Amazon Transcribe 이제 미국 서부 (캘리포니아 북부) (us-west-1) 에서 사용할 수 있습니다.	2019년 6월 27일

<a href="#">새로운 언어</a>	Amazon Transcribe 현대 표준 아랍어에 대한 지원을 추가합니다.	2019년 5월 28일
<a href="#">새 기능</a>	Amazon Transcribe 이제 미국 영어의 숫자 단어를 숫자로 변환합니다. 예를 들어 "마흔 둘"은 "42"로 트랜스크립션됩니다.	2019년 5월 23일
<a href="#">새로운 언어</a>	Amazon Transcribe 힌디어 및 인도 영어에 대한 지원을 추가합니다.	2019년 5월 15일
<a href="#">새 SDK</a>	이제 AWS Amazon Transcribe C++용 SDK가 지원됩니다.	2019년 5월 8일
<a href="#">새로운 언어</a>	Amazon Transcribe 스페인어 지원을 추가합니다.	2019년 4월 19일
<a href="#">AWS 리전 확장</a>	Amazon Transcribe 현재 EU (프랑크푸르트) (eu-central-1) 및 아시아 태평양 (서울) (ap-northeast-2) 에서 사용할 수 있습니다.	2019년 4월 18일
<a href="#">새로운 언어</a>	Amazon Transcribe 영국 영어, 프랑스어, 캐나다 프랑스어로 스트리밍 트랜스크립션 지원을 추가합니다.	2019년 4월 5일
<a href="#">새 기능</a>	Ruby AWS V3용 SDK는 이제 다음을 지원합니다. Amazon Transcribe	2019년 3월 25일



<a href="#">새 기능</a>	Amazon Transcribe 오디오 입력에서 Amazon Transcribe 인식하려는 특정 단어 목록인 사용자 지정 어휘를 사용할 수 있습니다.	2019년 3월 25일
<a href="#">새로운 언어</a>	Amazon Transcribe 독일어 및 한국어 지원을 추가합니다.	2019년 3월 22일
<a href="#">새로운 언어</a>	Amazon Transcribe 이제 미국 스페인어 (ES-US) 의 스트리밍 트랜스크립션을 지원합니다.	2019년 2월 7일
<a href="#">AWS 리전 확장</a>	Amazon Transcribe 는 이제 남아메리카 (상파울루) 에서 사용할 수 있습니다 (sa-east-1).	2019년 2월 7일
<a href="#">AWS 리전 확장</a>	Amazon Transcribe 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
<a href="#">새로운 언어</a>	Amazon Transcribe 프랑스어, 이탈리아어, 브라질식 포르투갈어에 대한 지원이 추가되었습니다.	2018년 12월 20일
<a href="#">새 기능</a>	Amazon Transcribe 이제 오디오 스트림의 트랜스크립션을 지원합니다.	2018년 11월 19일
<a href="#">새로운 언어</a>	Amazon Transcribe 호주 영어, 영국 영어 및 캐나다 프랑스어에 대한 지원을 추가합니다.	2018년 11월 15일

<a href="#">AWS 리전 확장</a>	Amazon Transcribe 이제 캐나다 (중부) (ca-central-1) 및 아시아 태평양 (시드니) (ap-south-east-2) 에서 사용할 수 있습니다.	2018년 7월 17일
<a href="#">새 기능</a>	이제 트랜스크립션 작업의 출력을 저장할 위치를 지정할 수 있습니다.	2018년 7월 11일
<a href="#">새 기능</a>	추가 및 통합. AWS CloudTrail Amazon CloudWatch Events	2018년 6월 28일
<a href="#">새 기능</a>	Amazon Transcribe 사용자 지정 어휘에 대한 지원을 추가합니다.	2018년 4월 4일
<a href="#">새 안내서</a>	이 문서는 첫 번째 Amazon Transcribe 개발자 안내서 릴리스입니다.	2017년 11월 29일

# AWS 용어집

최신 AWS 용어는 AWS 용어집 참조의 [AWS 용어집](#)을 참조하세요.

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.