aws

개발자 가이드

Amazon Polly

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

Amazon Polly: 개발자 가이드

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 과 제휴 관계이거나 관련이 있거나 후원 관계와 관계없이 해당 소유자의 자산입니다.

Table of Contents

Amazon Polly란? 1
이점 1
를 처음 사용하십니까? 2
작동 방식 3
를 처음 사용하십니까? 2
시작하기 4
Amazon Polly 설정 4
가입해 보세요. AWS 계정 4
관리자 액세스 권한이 있는 사용자 생성 5
콘솔에서 Amazon Polly 사용하기 6
1.1단계: 콘솔에서 음성 합성 퀵 스타트 7
1.2단계: 콘솔의 일반 텍스트 입력으로 음성을 합성합니다. 7
에서 Amazon Polly 사용하기 AWS CLI 8
2.1단계: 설정 AWS CLI 8
2.2단계: 를 사용하여 연습을 시작하기 AWS CLI 11
Python 예제 14
Python 설정 및 예제 테스트 (SDK) 14
Amazon Polly의 음성 16
음성 듣기 16
사용 가능한 목소리 17
브랜드 보이스 24
음성 속도 24
음성 속도 변경 25
이중 언어 음성 26
특유의 억양이 드러나는 이중 언어 음성 26
완벽한 이중 언어 음성 27
뉴스캐스터 목소리 27
Amazon Polly의 언어 30
지원되는 언어의 음소 및 Viseme 표 32
아랍어(arb) 33
아랍어(걸프)(ar-AE) 37
카탈루냐어(ca-ES) 41
중국어(광동어)(yue-CN) 44
표준 중국어(cmn-CN) 48

덴마크어(da-DK) 51
네덜란드어(벨기에)(nl-BE) 54
네덜란드어(nl-NL) 57
영어(미국)(en-US) 60
영어(호주)(en-AU) 63
영어(영국)(en-GB) 66
영어(인도)(en-IN) 69
영어(아일랜드)(en-IE) 72
영어(뉴질랜드)(en-NZ) 75
영어(남아프리카)(en-ZA) 80
영어(웨일스)(en-GB-WLS) 84
핀란드어(fi-FI) 87
프랑스어(fr-FR) 91
프랑스어(벨기에)(fr-BE) 93
프랑스어(캐나다)(fr-CA) 96
독일어(de-DE) 98
독일어(오스트리아)(de-AT) 102
힌디어(hi-IN) 105
아이슬란드어(is-IS) 108
이탈리아어(it-IT) 111
일본어(ja-JP) 113
한국어(ko-KR) 116
노르웨이어(nb-NO) 118
폴란드어(pl-PL) 121
포르투갈어(pt-PT) 124
포르투갈어(브라질)(pt-BR) 126
루마니아어(ro-RO) 129
러시아어(ru-RU) 131
스페인어(es-ES) 134
스페인어(멕시코)(es-MX) 137
스페인어(미국)(es-US) 139
스웨덴어(sv-SE) 142
터키어(tr-TR) 145
웨일스어(cy-GB) 148
음성 엔진 152
제너레이티브 엔진 152

사용 가능한 제너레이티브 보이스 153
기능 및 지역 호환성 153
콘솔의 제너레이티브 엔진 사용 154
롱폼 엔진 155
사용 가능한 장편 음색 155
기능 및 지역 호환성 156
콘솔에서 롱폼 엔진 사용 156
뉴럴 엔진 157
사용 가능한 뉴럴 보이스 158
기능 및 지역 호환성 161
콘솔에서 뉴럴 엔진 사용하기 162
표준 엔진 162
사용 가능한 표준 음성 162
기능 및 지역 호환성 166
콘솔에서 스탠다드 보이스 사용하기 167
스피치 마크 168
스피치 마크 유형 168
Viseme과 Amazon Polly 169
스피치 부호 사용 170
스피치 마크 요청 170
스피치 마크 출력 171
스피치 마크 예제 172
콘솔에서 스피치 마크 요청하기 174
SSML 사용 176
예약된 문자 177
콘솔에서 SSML 사용 180
다음 사이트에서 SSML을 사용합니다. AWS CLI 181
SSML을 음성 합성 명령과 함께 사용 182
SSML로 개선된 문서 합성하기 183
일반적인 Amazon Polly 작업에 SSML 사용 184
지원되는 SSML 태그 187
SSML로 확장된 텍스트 식별 189
일시 중지 추가 190
단어 강조 191
특정 단어에 대하여 다른 언어 지정 191
텍스트에 사용자 지정 태그 배치 193

단락 사이에 일시 중지 추가 ... 193
철자대로의 발음 사용 ... 194
볼륨, 말하기 속도 및 피치 제어 ... 195
합성된 스피치의 최대 기간 설정 ... 198
문장 사이에 일시 중지 추가 ... 201
특별한 유형의 단어를 말하는 방식 제어 ... 202
두문자어 및 약어 발음 ... 205
스피치의 일부를 구체적으로 지정함으로써 발음 개선 ... 206
숨소리 추가 ... 207
뉴스 진행자 말투 ... 211
동적 범위 압축 추가 ... 212
부드러운 어투 ... 214
음색 제어 ... 215
속삭임 기능 ... 216
어휘집 관리 ... 218
여러 어휘집 적용 ... 219
콘솔에서의 어휘집 관리 ... 220
콘솔에 어휘집 업로드 ... 220
콘솔에서 어휘 적용 (음성 합성) ... 221
콘솔에서 어휘 목록 필터링 ... 222
콘솔에서 어휘집 다운로드 ... 222
콘솔에서 어휘집 삭제하기 ... 223
의 어휘집 관리 AWS CLI ... 223
PutLexicon ... 224
GetLexicon ... 230
ListLexicons ... 231
DeleteLexicon ... 232
긴 오디오 파일 생성 ... 233
비동기 합성을 위한 IAM 정책 설정 ... 234
콘솔에서 긴 오디오 파일 생성 ... 235
에서 긴 오디오 파일 만들기 AWS CLI ... 236
코드 및 애플리케이션 예제 ... 239
샘플 코드 ... 239
자바 샘플 ... 239
Python 샘플 ... 249
애플리케이션의 예 ... 255

Python 예제 255
Java 예제 269
iOS 예제 274
안드로이드 예제 276
할당량 279
지원되는 리전 279
할당량 및 제한 속도 280
동시 요청 281
제한 완화를 위한 모범 사례 281
발음 어휘 281
SynthesizeSpeech API 오퍼레이션 282
SpeechSynthesisTask API 오퍼레이션 283
SSML(Speech Synthesis Markup Language) 283
보안 284
데이터 보호 284
유휴 데이터 암호화 285
전송 중 데이터 암호화 285
인터네트워크 트래픽 개인 정보 285
ID 및 액세스 관리 286
고객 286
자격 증명을 통한 인증 287
정책을 사용한 액세스 관리 290
Amazon Polly에서 IAM을 사용하는 방법 292
자격 증명 기반 정책 예시 299
Amazon Polly API 권한 참조 306
문제 해결 307
로깅 및 모니터링 309
규정 준수 검증 310
복원력 310
인프라 보안 311
보안 모범 사례 311
인터페이스 VPC 종단점 사용 311
가용성 312
Amazon Polly에 대한 VPC 엔드포인트 정책 생성 312
VPC와 Amazon Polly 간의 연결 테스트 312
Amazon Polly 엔드포인트에 대한 액세스 제어 313

VPC 컨텍스트 키에 대한 지원 314
를 사용하여 Amazon Polly API 호출을 로깅합니다. AWS CloudTrail 315
Amazon Polly 정보: CloudTrail 315
예: Amazon Polly 로그 파일 항목 316
CloudWatch 통합 318
CloudWatch 지표 가져오기 (콘솔) 318
에 대한 CloudWatch 지표 가져오기 AWS CLI 318
Amazon Polly 지표 319
Amazon Polly 지표의 차원 320
API 참조 321
작업 321
DeleteLexicon 322
DescribeVoices 324
GetLexicon 328
GetSpeechSynthesisTask 331
ListLexicons 334
ListSpeechSynthesisTasks 337
PutLexicon 340
StartSpeechSynthesisTask 343
SynthesizeSpeech 351
데이터 유형 357
Lexicon 358
LexiconAttributes 359
LexiconDescription 361
SynthesisTask 362
Voice 367
문서 기록 370
AWS 용어집 382
..... ccclxxxiii

Amazon Polly란?

Amazon Polly는 텍스트를 생생한 스피치로 변환하는 클라우드 서비스입니다. Amazon Polly를 사용하여 참여도와 접근성을 높이는 애플리케이션을 개발할 수 있습니다. Amazon Polly는 여러 언어를 지원하며 다양하고 생생한 음성을 포함합니다. Amazon Polly를 사용하면 여러 위치에서 작동하고 고객에게 적합한 음성을 사용하는 음성 지원 애플리케이션을 구축할 수 있습니다. 또한 합성한 텍스트에 대해서만 비용을 지불하면 됩니다. 또한 Amazon Polly에서 생성된 스피치를 캐시하여 추가 요금 없이 재생할 수 있습니다.

Amazon Polly는 생성형, 장편, 신경망 및 표준 text-to-speech (TTS) 옵션을 비롯한 다양한 음성 옵션을 제공합니다. 이러한 음성은 새로운 기계 학습 기술을 사용하여 음성 품질을 획기적으로 개선하여 가능한 가장 자연스럽고 인간과 유사한 음성을 제공합니다. text-to-speech Neural TTS 기술은 뉴스 내레이션 사용 사례에 맞게 조정된 뉴스캐스터 스피킹 스타일도 지원합니다.

Amazon Polly의 일반적인 사용 사례에는 뉴스리더, 게임, e러닝 플랫폼과 같은 모바일 애플리케이션, 시각 장애인을 위한 접근성 애플리케이션, 빠르게 성장하고 있는 사물 인터넷 (IoT) 부문 등이 포함되며 이에 국한되지는 않습니다.

Amazon Polly는 HIPAA(Health Insurance Portability and Accountability Act of 1996) 및 PCI DSS(Payment Card Industry Data Security Standard) 등의 규제 워크로드에 사용하도록 인증되었습니다.

이점

Amazon Polly를 사용하면 다음과 같은 이점이 있습니다.

- 고품질 — Amazon Polly는 고성능 제너레이티브, 롱 폼, 뉴럴 및 고품질 (TTS) 음성을 제공합니다. text-to-speech 이러한 기술은 높은 발음 정확도로 자연스러운 음성을 합성합니다 (약어, 두문자어 확장, 날짜/시간 해석, 동음이의어 명확화 포함).
- 짧은 지연 시간 — Amazon Polly는 빠른 응답을 제공하므로 대화 시스템과 같이 지연 시간이 짧은 사용 사례에 적합한 옵션입니다.
- 광범위한 언어 및 음성 포트폴리오 지원 — Amazon Polly는 수십 개의 음성과 언어를 지원하며 대부분의 언어에 대해 남성 및 여성 음성 옵션을 제공합니다. 온라인에 제공되는 신경 음성이 늘어남에 따라 이 숫자도 계속 늘어날 전망입니다. 미국 영어 음성 Matthew 및 Joanna는 전문 뉴스 앵커가 말하는 것처럼 신경 뉴스 진행자 말투를 사용할 수도 있습니다.
- 비용 효율적 — Amazon Polly의 pay-per-use 모델은 설치 비용이 없음을 의미합니다. 소규모로 시작해서 애플리케이션이 성장함에 따라 규모를 확장하십시오.

- 클라우드 기반 솔루션 – 디바이스 기반 TTS 솔루션을 사용하려면 충분한 컴퓨팅 리소스, 매우 높은 CPU 전력, RAM 및 디스크 공간이 필요합니다. 이로 인해 개발 비용이 증가하고 태블릿, 스마트폰 등과 같은 디바이스의 전력 소비가 증가할 수 있습니다. 반면 TTS 변환은 로컬 리소스 요구 사항을 AWS 클라우드 크게 줄입니다. 이를 통해 사용 가능한 모든 언어와 음성을 뛰어난 품질로 지원할 수 있습니다. 또한 음성 개선 기능은 모든 최종 사용자가 즉시 사용할 수 있으며 장치를 추가로 업데이트할 필요가 없습니다.

> **Note**
>
> 브라우저에서 Amazon Polly 음성의 예를 들으려면 Amazon Polly 제품 개요를 참조하십시오.

를 처음 사용하십니까?

Amazon Polly를 처음 사용하는 경우 다음 섹션을 나열된 순서대로 읽는 것이 좋습니다.

1. Amazon Polly의 작동 방식— 이 섹션에서는 간단한 환경을 만들기 위해 사용할 수 있는 다양한 Amazon Polly 입력 및 옵션을 소개합니다.
2. 아마존 Polly 시작하기 – 이 섹션에서는 계정을 설정하고 Amazon Polly 스피치 합성을 테스트합니다.
3. 애플리케이션의 예 – 이 섹션에서는 Amazon Polly 탐색에 사용할 수 있는 추가 예제를 제공합니다.

Amazon Polly의 작동 방식

Amazon Polly는 입력 텍스트를 실제 같은 스피치로 변환합니다. Amazon Polly 음성을 사용하려면 음성 엔진을 선택하고 음성 합성 메서드를 호출하여 합성할 텍스트를 제공한 다음 오디오 출력 형식을 지정합니다. 그러면 Amazon Polly는 제공된 텍스트를 고품질 스피치 오디오 스트림으로 합성합니다.

- 입력 텍스트 – 합성하려는 텍스트를 제공하면 Amazon Polly에서 오디오 스트림을 반환합니다. 입력은 일반 텍스트 또는 SSML (음성 합성 마크업 언어) 형식으로 제공할 수 있습니다. SSML을 사용하면 발음, 볼륨, 음조 및 스피치 속도 등 음성의 다양한 측면을 제어할 수 있습니다. 자세한 내용은 SSML 문서에서 음성 생성을(를) 참조하세요.
- 사용 가능한 음성 – Amazon Polly에서는 언어 포트폴리오와 이중 언어 음성(영어 및 힌디어)을 비롯한 다양한 음성을 제공합니다. 대부분의 언어는 남성과 여성이 둘 다 지원되는 여러 음성 중에서 선택할 수 있습니다. 스피치 합성 작업을 시작할 때 음성 ID를 지정하면 Amazon Polly에서 이 음성을 사용하여 텍스트를 스피치로 변환합니다. Amazon Polly는 번역 서비스가 아닙니다. 즉, 합성된 스피치는 텍스트와 동일한 언어를 사용합니다. 숫자로 표시되는 숫자 (예: 53이 아닌 53자리) 는 텍스트가 아닌 음성 언어로 합성됩니다. 자세한 내용은 Amazon Polly의 음성을 참조하세요.
- 출력 형식 – Amazon Polly는 여러 가지 형식으로 합성된 스피치를 전달할 수 있습니다. 사용자의 요구 사항에 맞는 음성 형식을 선택할 수 있습니다. 예를 들어 웹 및 모바일 애플리케이션에서 사용할 수 있게 MP3 또는 Ogg Vorbis 형식의 스피치를 요청하거나, 또는 장치 및 전화 통신 솔루션에서 사용할 PCM 출력 형식을 요청할 수도 있습니다. AWS IoT

> **Note**
>
> 브라우저에서 Amazon Polly 음성의 예를 들으려면 Amazon Polly 제품 개요를 참조하십시오.

를 처음 사용하십니까?

Amazon Polly를 처음 사용하는 경우 다음 주제를 순서대로 읽는 것이 좋습니다.

- 아마존 Polly 시작하기
- 애플리케이션의 예
- Amazon Polly의 할당량

아마존 Polly 시작하기

Amazon Polly는 기존 애플리케이션과 쉽게 통합할 수 있는 여러 API 작업을 제공합니다. 지원되는 작업 목록은 작업을(를) 참조하세요. 다음 두 가지 옵션 중 하나를 사용할 수 있습니다.

- AWS SDK — SDK를 사용할 때 Amazon Polly에 대한 요청은 사용자가 제공한 자격 증명을 사용하여 자동으로 서명 및 인증됩니다. 이것은 애플리케이션을 구축할 때 권장되는 선택 사항입니다.
- AWS CLI — 코드를 작성하지 않고도 AWS CLI 를 사용하여 Amazon Polly를 사용할 수 있습니다.

다음 섹션에서는 Amazon Polly를 사용하기 시작하는 방법을 설명합니다.

주제

- Amazon Polly 설정
- 콘솔에서 Amazon Polly 사용하기
- 에서 Amazon Polly 사용하기 AWS CLI
- Python 예제

Amazon Polly 설정

Amazon Polly를 처음 사용하기 전에 먼저 가입해야 합니다. AWS Amazon Web Services (AWS) 에 가입하면 Amazon Polly를 비롯한 내의 AWS모든 서비스에 AWS 계정이 자동으로 가입됩니다. 사용한 서비스에 대해서만 청구됩니다. 신규 AWS 고객인 경우 Amazon Polly를 무료로 시작할 수 있습니다. 자세한 내용은 AWS 프리 티어를 참조하세요.

이미 AWS 계정이 있는 경우 다음 활동 중 하나로 이동할 수 있습니다.

- 콘솔에서 Amazon Polly 사용하기
- 에서 Amazon Polly 사용하기 AWS CLI

가입해 보세요. AWS 계정

계정이 없는 경우 다음 단계를 완료하여 계정을 만드세요. AWS 계정

가입하려면 AWS 계정

1. https://portal.aws.amazon.com/billing/signup을 여세요.

2. 온라인 지시 사항을 따르세요.

   등록 절차 중에는 전화를 받고 키패드로 인증 코드를 입력하는 과정이 있습니다.

   에 AWS 계정가입하면 AWS 계정 루트 사용자a가 생성됩니다. 루트 사용자에게는 계정의 모든 AWS 서비스 및 리소스 액세스 권한이 있습니다. 보안 모범 사례는 사용자에게 관리 액세스 권한을 할당하고, 루트 사용자만 사용하여 루트 사용자 액세스 권한이 필요한 작업을 수행하는 것입니다.

AWS 가입 절차가 완료된 후 확인 이메일을 보냅니다. 언제든지 https://aws.amazon.com/으로 가서 내 계정(My Account)을 선택하여 현재 계정 활동을 보고 계정을 관리할 수 있습니다.

관리자 액세스 권한이 있는 사용자 생성

등록한 AWS 계정후에는 일상적인 작업에 루트 사용자를 사용하지 않도록 관리 사용자를 보호하고 AWS IAM Identity Center활성화하고 생성하십시오 AWS 계정 루트 사용자.

보안을 유지하세요. AWS 계정 루트 사용자

1. 루트 사용자를 선택하고 AWS 계정 이메일 주소를 입력하여 계정 AWS Management Console소유자로 로그인합니다. 다음 페이지에서 비밀번호를 입력합니다.

   루트 사용자를 사용하여 로그인하는 데 도움이 필요하면AWS 로그인 사용 설명서의 루트 사용자로 로그인을 참조하세요.
2. 루트 사용자의 다중 인증(MFA)을 활성화합니다.

   지침은 IAM 사용 설명서의 AWS 계정 루트 사용자 (콘솔) 에 대한 가상 MFA 디바이스 활성화를 참조하십시오.

관리자 액세스 권한이 있는 사용자 생성

1. IAM Identity Center를 활성화합니다.

   지침은 AWS IAM Identity Center 사용 설명서의 AWS IAM Identity Center설정을 참조하세요.
2. IAM Identity Center에서 사용자에게 관리 액세스 권한을 부여합니다.

   를 ID 소스로 사용하는 방법에 대한 자습서는 사용 설명서의 기본값으로 IAM Identity Center 디렉터리사용자 액세스 구성을 참조하십시오. IAM Identity Center 디렉터리 AWS IAM Identity Center

관리 액세스 권한이 있는 사용자로 로그인

- IAM IDentity Center 사용자로 로그인하려면 IAM IDentity Center 사용자를 생성할 때 이메일 주소로 전송된 로그인 URL을 사용합니다.

  IAM Identity Center 사용자를 사용하여 로그인하는 데 도움이 필요하면 사용 설명서의 AWS 액세스 포털에 로그인을 참조하십시오.AWS 로그인

추가 사용자에게 액세스 권한 할당

1. IAM Identity Center에서 최소 권한 적용 모범 사례를 따르는 권한 세트를 생성합니다.

   지침은AWS IAM Identity Center 사용 설명서의 Create a permission set를 참조하세요.
2. 사용자를 그룹에 할당하고, 그룹에 Single Sign-On 액세스 권한을 할당합니다.

   지침은AWS IAM Identity Center 사용 설명서의 Add groups를 참조하세요.

IAM에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Identity and Access Management (IAM)
- 시작하기
- IAM 사용 설명서

> **Note**
>
> AWS 계정 ID를 기록해 둡니다. 다음 단계에서 필요합니다.

콘솔에서 Amazon Polly 사용하기

Amazon Polly 콘솔에서 Amazon Polly의 음성 합성 기능을 빠르게 테스트하고 사용할 수 있습니다. Amazon Polly 콘솔은 일반 텍스트 또는 SSML 입력으로부터의 음성 합성을 지원합니다.

주제

- 1.1단계: 콘솔에서 음성 합성 퀵 스타트
- 1.2단계: 콘솔의 일반 텍스트 입력으로 음성을 합성합니다.

1.1단계: 콘솔에서 음성 합성 퀵 스타트

콘솔에서 Amazon Polly 음성 합성의 음성 품질을 빠르게 테스트할 수 있습니다.

콘솔에서 Amazon Polly 음성을 들으려면

1. AWS Management Console 로그인하고 https://console.aws.amazon.com/polly/ 에서 Amazon Polly 콘솔을 엽니다.
2. 텍스트 투 스피치 탭을 선택합니다. 텍스트 필드에 예제 텍스트가 로드되므로 Amazon Polly를 빠르게 사용해 볼 수 있습니다.
3. SSML을 비활성화합니다.
4. 엔진에서 제너레이티브, 롱 폼, 뉴럴 또는 스탠다드를 선택합니다.
5. 언어와 AWS 지역을 선택한 다음 음성을 선택합니다. (엔진용 Neural (Neural) 을 선택하면 NTTS를 지원하는 언어와 음성만 사용할 수 있습니다. 스탠다드 및 롱 폼 보이스는 모두 비활성화됩니다.
6. 듣기를 선택합니다.

심층 테스트에 대한 자세한 내용은 다음 항목을 참조하세요.

- 1.2단계: 콘솔의 일반 텍스트 입력으로 음성을 합성합니다.
- 콘솔에서 SSML 사용
- 콘솔에서 어휘 적용 (음성 합성)

1.2단계: 콘솔의 일반 텍스트 입력으로 음성을 합성합니다.

다음 절차는 일반 텍스트 입력을 사용하여 음성을 합성합니다. (“W3C”와 날짜 “10/3" (10월 3일) 이 어떻게 합성되는지 주의하십시오.)

콘솔에서 일반 텍스트 입력을 사용하여 음성을 합성하려면

1. Amazon Polly 콘솔에 로그온한 후 [Amazon Polly 사용해 보기] 를 선택합니다. 그런 다음 '텍스트 음성 변환' 탭을 선택합니다.
2. SSML을 비활성화합니다.
3. 이 텍스트를 입력란에 입력하거나 붙여 넣습니다.

```
He was caught up in the game.
```

```
In the middle of the 10/3/2014 W3C meeting
he shouted, "Score!" quite loudly.
```

4. 엔진의 경우 제너레이티브, 롱 폼, 뉴럴 또는 스탠다드를 선택합니다.
5. 언어와 AWS 지역을 선택한 다음 음성을 선택합니다. (Engine용 Neural 옵션을 선택하면 NTTS를 지원하는 언어와 음성만 사용할 수 있습니다. 스탠다드 및 롱 폼 보이스는 모두 비활성화됩니다.
6. 스피치를 즉시 들으려면 듣기를 선택합니다.
7. 스피치를 파일로 저장하려면 다음 중 하나를 수행합니다.

   a. 다운로드를 선택합니다.

   b. 다른 파일 형식으로 변경하려면 추가 설정을 확장하고, 스피치 파일 형식 설정을 활성화하고, 원하는 파일 형식을 선택한 다음 다운로드를 선택합니다.

심층 예제에 대한 자세한 내용은 다음 주제를 참조하세요.

- 콘솔에서 어휘 적용 (음성 합성)
- 콘솔에서 SSML 사용

에서 Amazon Polly 사용하기 AWS CLI

Amazon Polly 콘솔과 에서 거의 모든 동일한 작업을 수행할 수 있습니다. AWS CLI하지만 에서는 합성된 음성을 들을 수 없습니다. AWS CLI에서 오디오를 사용하여 작업하려면 텍스트를 파일에 저장하십시오. AWS CLI그런 다음 선택한 오디오 애플리케이션에서 파일을 엽니다.

주제

- 2.1단계: 설정 AWS CLI
- 2.2단계: 를 사용하여 연습을 시작하기 AWS CLI

2.1단계: 설정 AWS CLI

다음 단계에 따라 Amazon Polly와 함께 AWS CLI 작동하도록 다운로드하고 구성하십시오.

> **Important**
>
> 이 연습의 단계는 수행할 AWS CLI 필요가 없습니다. 이 안내서의 일부 연습에서는 AWS CLI 를 사용합니다. 이 단계를 건너뛰고 으로 2.2단계: 를 사용하여 연습을 시작하기 AWS CLI 이동한 다음 AWS CLI 나중에 필요할 때 설정해도 됩니다.

설정하기 AWS CLI

설정하려면 AWS Command Line Interface

1. AWS CLI를 다운로드하고 구성합니다. 관련 지침은 AWS Command Line Interface 사용 설명서에서 다음 주제를 참조하세요.
   - 를 사용하여 설정하기 AWS Command Line Interface
   - 구성하기 AWS Command Line Interface
2. AWS CLI AWS Config 파일에 관리자 사용자의 이름이 지정된 프로필을 추가합니다. AWS CLI 명령을 실행할 때 이 프로필을 사용할 수 있습니다. 프로파일 명명에 대한 자세한 설명은AWS Command Line Interface 사용자 가이드의 프로파일 명명을 참조하십시오.

   ```
   [profile adminuser]
       aws_access_key_id = adminuser access key ID
       aws_secret_access_key = adminuser secret access key
       region = aws-region
   ```

   사용 가능한 AWS 지역 및 Amazon Polly에서 지원하는 지역 목록은 의 지역 및 엔드포인트를 참조하십시오. Amazon Web Services 일반 참조

   > **Note**
   >
   > 구성 시 지정한 Amazon Polly 지원 지역을 사용하는 경우 코드 예제에서 다음 줄을 생략하십시오. AWS CLI AWS CLI
   >
   > ```
   > --region aws-region
   > ```

3. 명령 프롬프트에서 다음 help 명령을 입력하여 설정을 확인합니다.

```
aws help
```

유효한 AWS 명령 목록이 창에 나타나야 합니다. AWS CLI

에서 Amazon Polly를 활성화하십시오. AWS CLI

이전에 을 (를) 다운로드하여 구성한 경우 AWS CLI, 다시 구성하지 않는 한 Amazon Polly를 사용하지 못할 수 있습니다. AWS CLI다음 절차는 이것이 필요한지 여부를 확인합니다.

Amazon Polly를 활성화하려면 AWS CLI

1. 명령 프롬프트에 다음 도움말 명령을 입력하여 Amazon Polly의 가용성을 확인합니다. AWS CLI

   ```
   aws polly help
   ```

   Amazon Polly에 대한 설명이 표시되고 유효한 명령 목록이 AWS CLI 창에 나타나면 Amazon Polly를 바로 사용할 수 있습니다. AWS CLI 이 경우 이 절차의 나머지 부분은 건너뛸 수 있습니다. 이 메시지가 표시되지 않으면 2단계로 진행합니다.
2. 다음 두 옵션 중 하나를 사용하여 Amazon Polly를 활성화합니다.

   a. 를 제거하고 다시 설치합니다. AWS CLI

      지침은 AWS Command Line Interface 사용 설명서의 AWS Command Line Interface설치를 참조하세요.

      또는

   b. service-2.json 파일을 다운로드하세요.

      명령 프롬프트에서 다음 명령을 실행합니다.

      ```
      aws configure add-model --service-model file://service-2.json --service-name
       polly
      ```
3. Amazon Polly가 사용 가능한지 다시 확인합니다.

   ```
   aws polly help
   ```

Amazon Polly에 대한 설명이 표시되어야 합니다.

에서 음성 엔진을 설정합니다. AWS CLI

에서 `engine` 매개 변수는 선택 사항이며,, 4개 값 (`generative`, `long-form`neural,) 을 사용할 수 `standard` 있습니다. AWS CLI예를 들어, 다음 코드를 사용하여 미국 서부-2 (오레곤) 지역에서 `start-speech-synthesis-task` AWS CLI 명령을 실행하는 경우

```
aws polly start-speech-synthesis-task \
  --engine neural
  --region us-west-2 \
  --endpoint-url "https://polly.us-west-1.amazonaws.com/" \
  --output-format mp3 \
  --output-s3-bucket-name your-bucket-name \
  --output-s3-key-prefix optional/prefix/path/file \
  --voice-id Joanna \
  --text file://text_file.txt
```

출력은 다음과 비슷합니다.

```
"SynthesisTask":
{
     "CreationTime": [..],
     "Engine": "neural",
     "OutputFormat": "mp3",
     "OutputUri": "https://s3.us-west-1.amazonaws.com/your-bucket-name/optional/prefix/
path/file.<task_id>.mp3",
     "TextType": "text",
     "RequestCharacters": [..],
     "TaskStatus": "scheduled",
     "TaskId": [task_id],
     "VoiceId": "Joanna"
 }
```

2.2단계: 를 사용하여 연습을 시작하기 AWS CLI

이미 설정한 경우 Amazon AWS CLI Polly에서 제공하는 음성 합성을 테스트할 수 있습니다. 이 연습에서는 입력 텍스트를 전달하여 `SynthesizeSpeech` 작업을 호출합니다. 결과 오디오를 파일로 저장하고 콘텐츠를 확인할 수 있습니다.

1. synthesize-speech AWS CLI 명령을 실행하여 샘플 텍스트를 오디오 파일에 합성합니다 (hello.mp3).

   다음 AWS CLI 예제는 유닉스, 리눅스, macOS용으로 포맷되었습니다. Windows의 경우 각 줄 끝에 있는 백슬래시(\) Unix 연속 문자를 캐럿(^)으로 바꿉니다. 입력 텍스트는 큰 따옴표(")로 감싸고 내부 태그에는 작은 따옴표(')를 사용합니다.

   ```
   aws polly synthesize-speech \
       --output-format mp3 \
       --voice-id Joanna \
       --text 'Hello, my name is Joanna. I learned about the W3C on 10/3 of last
    year.' \
       hello.mp3
   ```

   를 호출할 synthesize-speech 때는 선택한 음성으로 합성할 샘플 텍스트를 제공합니다. 음성 ID (다음 단계에서 설명) 와 출력 형식을 제공해야 합니다. 이 명령은 결과 오디오를 hello.mp3 파일에 저장합니다. 이 작업은 MP3 파일 외에도 다음과 같은 출력을 콘솔에 전송합니다.

   ```
   {
           "ContentType": "audio/mpeg",
           "RequestCharacters": "71"
   }
   ```

2. hello.mp3 결과 파일을 재생하여 합성된 스피치를 확인합니다.
3. DescribeVoices 작업을 사용해 사용 가능한 음성 목록을 가져옵니다. 다음 describe-voices AWS CLI 명령을 실행합니다.

   ```
   aws polly describe-voices
   ```

   이에 대한 응답으로 Amazon Polly에서는 사용 가능한 모든 음성 목록을 반환합니다. 각 음성에 대해 응답으로 음성 ID, 언어 코드, 언어 이름 및 성별과 같은 메타데이터를 제공합니다. 다음은 응답 예입니다.

   ```
   {
       "Voices": [
           {
               "Gender": "Female",
               "Name": "Salli",
               "LanguageName": "US English",
               "Id": "Salli",
   ```

```
            "LanguageCode": "en-US",
            "SupportedEngines": [
                "neural",
                "standard",
                "generative"
            ]
        },
        {
            "Gender": "Female",
            "Name": "Danielle",
            "LanguageName": "US English",
            "Id": "Danielle",
            "LanguageCode": "en-US",
            "SupportedEngines": [
                "long-form"
            ]
        }
    ]
}
```

필요에 따라 언어 코드를 지정하여 특정 언어에 사용 가능한 음성을 찾을 수 있습니다. Amazon Polly는 수십 개의 음성을 지원합니다. 다음 예에서는 브라질 포르투갈어에 대한 모든 음성이 나열됩니다.

```
aws polly describe-voices \
    --language-code pt-BR
```

언어 코드 목록은 Amazon Polly의 언어를 참조하세요. 이러한 언어 코드는 W3C 언어 식별 태그입니다 (*## ### ISO 639 ## - ISO 3166* 국가 코드). 예로, en-US(미국 영어), en-GB(영국 영어) 및 es-ES(스페인어) 등입니다. AWS CLI 에서 `help` 옵션을 사용하여 언어 코드 목록을 얻을 수도 있습니다.

```
aws polly describe-voices help
```

Python 예제

이 가이드에서는 Amazon Polly에 API를 호출하는 AWS SDK for Python (Boto) 데 사용하는 몇 가지 Python 코드 예제를 제공합니다. Python을 설정하고 다음 단원에 나오는 예제 코드를 테스트하는 것이 좋습니다. 추가 예제는 다음(애플리케이션의 예)을 참조하십시오.

Python 설정 및 예제 테스트 (SDK)

Python 예제 코드를 테스트하려면 AWS SDK for Python (Boto)가 필요합니다. 지침은 AWS SDK for Python (Boto3)을(를) 참조하세요.

Python 코드 예제를 테스트하려면

다음 Python 코드 예제는 다음과 같은 작업을 수행합니다.

- 를 AWS SDK for Python (Boto) 호출하여 Amazon Polly에 SynthesizeSpeech 요청을 전송합니다 (일부 텍스트를 입력으로 제공).
- 응답에서 결과 오디오 스트림에 액세스하고 오디오를 로컬 디스크의 파일(speech.mp3)에 저장합니다.
- 로컬 시스템의 기본 오디오 플레이어로 오디오 파일을 재생합니다.

코드를 파일(example.py)에 저장하고 실행합니다.

```
"""Getting Started Example for Python 2.7+/3.3+"""
from boto3 import Session
from botocore.exceptions import BotoCoreError, ClientError
from contextlib import closing
import os
import sys
import subprocess
from tempfile import gettempdir

# Create a client using the credentials and region defined in the [adminuser]
# section of the AWS credentials file (~/.aws/credentials).
session = Session(profile_name="adminuser")
polly = session.client("polly")

try:
    # Request speech synthesis
    response = polly.synthesize_speech(Text="Hello world!", OutputFormat="mp3",
```

```
                                        VoiceId="Joanna")
except (BotoCoreError, ClientError) as error:
    # The service returned an error, exit gracefully
    print(error)
    sys.exit(-1)

# Access the audio stream from the response
if "AudioStream" in response:
    # Note: Closing the stream is important because the service throttles on the
    # number of parallel connections. Here we are using contextlib.closing to
    # ensure the close method of the stream object will be called automatically
    # at the end of the with statement's scope.
        with closing(response["AudioStream"]) as stream:
           output = os.path.join(gettempdir(), "speech.mp3")

           try:
            # Open a file for writing the output as a binary stream
                with open(output, "wb") as file:
                   file.write(stream.read())
           except IOError as error:
              # Could not write to file, exit gracefully
              print(error)
              sys.exit(-1)

else:
    # The response didn't contain audio data, exit gracefully
    print("Could not stream audio")
    sys.exit(-1)

# Play the audio using the platform's default player
if sys.platform == "win32":
    os.startfile(output)
else:
    # The following works on macOS and Linux. (Darwin = mac, xdg-open = linux).
    opener = "open" if sys.platform == "darwin" else "xdg-open"
    subprocess.call([opener, output])
```

예제 애플리케이션을 포함한 추가 예제는 애플리케이션의 예을 참조하세요.

Amazon Polly의 음성

Amazon Polly는 다양한 언어로 수십 개의 생생한 음성을 제공합니다. 각 음성은 모국어 사용자를 통해 만들어지므로, 동일한 언어 내에서도 음성마다 차이가 있습니다. 를 사용하여 원하는 AWS Management Console 텍스트로 각 음성을 테스트할 수도 있습니다. 대부분의 언어에서 남성과 여성 음성이 하나 이상 있으며, 각각 하나 이상 있는 경우가 많습니다. 음성이 하나만 있는 경우는 적습니다.

> **Note**
>
> 브라우저에서 Amazon Polly 음성의 예를 들으려면 Amazon Polly 제품 개요를 참조하십시오.

주제

- 음성 듣기
- 사용 가능한 목소리
- 음성 속도
- 이중 언어 음성
- 뉴스캐스터 목소리

음성 듣기

Amazon Polly를 설정하고 나면 콘솔에서 사용자 지정 텍스트를 사용하여 음성을 테스트할 수 있습니다.

콘솔에서 Amazon Polly 음성을 들으려면

1. AWS Management Console 로그인하고 https://console.aws.amazon.com/polly/ 에서 Amazon Polly 콘솔을 엽니다.
2. 텍스트 투 스피치 탭을 선택합니다.
3. 엔진의 경우 제너레이티브, 롱 폼, 뉴럴 또는 스탠다드를 선택합니다.
4. 언어와 지역을 선택합니다. 그런 다음 음성을 선택하세요.
5. 말하려는 음성의 텍스트를 입력하거나 기본 문구를 사용한 후 듣기를 선택합니다.

> **Note**
>
> 포함된 언어 수와 음성 인벤토리는 추가 선택을 포함하도록 지속적으로 업데이트되고 있습니다. 새 언어 또는 음성을 제안하려면 언제든 이 페이지에 대한 피드백을 제출하세요. 안타깝게도 출시되는 특정 새 언어에 대한 계획에 대해서는 언급할 수 없습니다.

사용 가능한 목소리

Amazon Polly는 텍스트에서 음성을 합성하기 위해 여러 언어로 다양하고 생생한 음성을 제공합니다. 다음 표에는 Amazon Polly가 제공하는 모든 음성이 나와 있습니다.

	언어 및 사투리	언어 코드	이름/ID	Gender	제너레이티브 보이스	롱폼 보이스	신경 음성	스탠다드 보이스
1	아랍어	arb	Zeina	여성	아니요	아니요	아니요	예
2	아랍어 (걸프)	ar-AE	Hala*	여성	아니요	아니요	예	아니요
			Zayd*	남성	아니요	아니요	예	아니요
3	네덜란드어(벨기에)	nl-BE	Lisa	여성	아니요	아니요	예	아니요
4	카탈루냐어	ca-ES	Arlet	여성	아니요	아니요	예	아니요
5	중국어 (광동어)	yue-CN	Hiujin	여성	아니요	아니요	예	아니요
6	표준 중국어	cmn-CN	Zhiyu	여성	아니요	아니요	예	예
7	덴마크어	da-DK	Naja	여성	아니요	아니요	아니요	예
			Mads	남성	아니요	아니요	아니요	예

	언어 및 사투리	언어 코드	이름/ID	Gender	제너레이티브 보이스	롱폼 보이스	신경 음성	스탠다드 보이스
			Sofie	여성	아니요	아니요	예	아니요
8	네덜란드어	nl-NL	Laura	여성	아니요	아니요	예	아니요
			Lotte	여성	아니요	아니요	아니요	예
			Ruben	남성	아니요	아니요	아니요	예
9	영어(호주)	en-AU	Nicole	여성	아니요	아니요	아니요	예
			Olivia	여성	아니요	아니요	예	아니요
			Russell	남성	아니요	아니요	아니요	예
10	영어(영국)	en-GB	Amy**	여성	예	아니요	예	예
			Emma	여성	아니요	아니요	예	예
			Brian	남성	아니요	아니요	예	예
			Arthur	남성	아니요	아니요	예	아니요
11	영어(인도)	en-IN	Aditi*	여성	아니요	아니요	아니요	예
			Raveena	여성	아니요	아니요	아니요	예
			Kajal*	여성	아니요	아니요	예	아니요
12	영어(아일랜드)	en-IE	Niamh	여성	아니요	아니요	예	아니요
13	영어(뉴질랜드)	en-NZ	Aria	여성	아니요	아니요	예	아니요
14	영어(남아프리카)	en-ZA	Ayanda	여성	아니요	아니요	예	아니요

	언어 및 사투리	언어 코드	이름/ID	Gender	제너레이티브 보이스	롱폼 보이스	신경 음성	스탠다드 보이스
15	영어(미국)	en-US	Danielle	여성	아니요	예	예	아니요
			Gregory	남성	아니요	예	예	아니요
			Ivy	여성(어린이)	아니요	아니요	예	예
			Joanna**		아니요	아니요	예	예
			Kendra	여성	아니요	아니요	예	예
			Kimberly	여성	아니요	아니요	예	예
			Salli	여성	아니요	아니요	예	예
			Joey	여성	아니요	아니요	예	예
			Justin	남성	아니요	아니요	예	아니요
			Kevin	남성(어린이)	아니요	아니요	예	예
			Matthew*,	남성(어린이)	예	아니요	예	아니요
			Ruth		예	예	예	아니요
			Stephen	남성	아니요	아니요	예	아니요
				여성				
				남성				
16	영어(웨일스)	en-GB-WLS	Geraint	남성	아니요	아니요	아니요	예
17	핀란드어	fi-FI	Suvi	여성	아니요	아니요	예	아니요

	언어 및 사투리	언어 코드	이름/ID	Gender	제너레이티브 보이스	롱폼 보이스	신경 음성	스탠다드 보이스
18	프랑스어	fr-FR	Céline/ Celine	여성	아니요	아니요	아니요	예
			Léa	여성	아니요	아니요	예	예
			Mathieu	남성	아니요	아니요	아니요	예
			Rémi	남성	아니요	아니요	예	아니요
19	프랑스어(벨기에)	fr-BE	Isabelle	여성	아니요	아니요	예	아니요
20	프랑스어(캐나다)	fr-CA	Chantal	여성	아니요	아니요	아니요	예
			Gabrielle	여성	아니요	아니요	예	아니요
			Liam	남성	아니요	아니요	예	아니요
21	독일어	de-DE	Marlene	여성	아니요	아니요	아니요	예
			Vicki	여성	아니요	아니요	예	예
			Hans	남성	아니요	아니요	아니요	예
			Daniel	남성	아니요	아니요	예	아니요
22	독일어(오스트리아)	de-AT	Hannah	여성	아니요	아니요	예	아니요
23	힌디어	hi-IN	Aditi*	여성	아니요	아니요	아니요	예
			Kajal*	여성	아니요	아니요	예	아니요

	언어 및 사투리	언어 코드	이름/ID	Gender	제너레이티브 보이스	롱폼 보이스	신경 음성	스탠다드 보이스
24	아이슬란드어	is-IS	Dóra/Dora	여성	아니요	아니요	아니요	예
			Karl	남성	아니요	아니요	아니요	예
25	이탈리아어	it-IT	Carla	여성	아니요	아니요	아니요	예
			Bianca	여성	아니요	아니요	예	예
			Giorgio	남성	아니요	아니요	아니요	예
			Adriano	남성	아니요	아니요	예	아니요
26	일본어	ja-JP	Mizuki	여성	아니요	아니요	아니요	예
			Takumi	남성	아니요	아니요	예	예
			Kazuha	여성	아니요	아니요	예	아니요
			Tomoko	여성	아니요	아니요	예	아니요
27	한국어	ko-KR	Seoyeon	여성	아니요	아니요	예	예
28	노르웨이어	nb-NO	Liv	여성	아니요	아니요	아니요	예
			Ida	여성	아니요	아니요	예	아니요
29	폴란드어	pl-PL	Ewa	여성	아니요	아니요	아니요	예
			Maja	여성	아니요	아니요	아니요	예
			Jacek	남성	아니요	아니요	아니요	예
			Jan	남성	아니요	아니요	아니요	예
			Ola	여성	아니요	아니요	예	아니요

	언어 및 사투리	언어 코드	이름/ID	Gender	제너레이티브 보이스	롱폼 보이스	신경 음성	스탠다드 보이스
30	포르투갈어(브라질)	pt-BR	Camila	여성	아니요	아니요	예	예
			Vitória/ Vitoria	여성	아니요	아니요	예	예
			Ricardo	남성	아니요	아니요	아니요	예
			Thiago	남성	아니요	아니요	예	아니요
31	포르투갈어(유럽)	pt-PT	Inês/ Ines	여성	아니요	아니요	예	예
			Cristiano	남성	아니요	아니요	아니요	예
32	루마니아어	ro-RO	Carmen	여성	아니요	아니요	아니요	예
33	러시아어	ru-RU	Tatyana	여성	아니요	아니요	아니요	예
			Maxim	남성	아니요	아니요	아니요	예
34	스페인어(유럽)	es-ES	Conchita	여성	아니요	아니요	아니요	예
			Lucia	여성	아니요	아니요	예	예
			Enrique	남성	아니요	아니요	아니요	예
			Sergio	남성	아니요	아니요	예	아니요
35	스페인어(멕시코)	es-MX	Mia	여성	아니요	아니요	예	예
			Andrés	남성	아니요	아니요	예	아니요

	언어 및 사투리	언어 코드	이름/ID	Gender	제너레이티브 보이스	롱폼 보이스	신경 음성	스탠다드 보이스
36	스페인어(미국)	es-US	Lupe**	여성	아니요	아니요	예	예
			Penélope/Penelope	여성	아니요	아니요	아니요	예
			Miguel	남성	아니요	아니요	아니요	예
			Pedro	남성	아니요	아니요	예	아니요
37	스웨덴어	sv-SE	Astrid	여성	아니요	아니요	아니요	예
			Elin	여성	아니요	아니요	예	아니요
38	터키어	tr-TR	Filiz	여성	아니요	아니요	아니요	예
			버킷	여성	아니요	아니요	예	아니요
39	웨일스어	cy-GB	Gwyneth	여성	아니요	아니요	아니요	예

* 이 음성은 이중 언어입니다. 자세한 내용은 이중 언어 음성을(를) 참조하세요.

** 이 음성은 신경 형식으로 사용할 때 뉴스 진행자 말투로 사용할 수 있습니다. 자세한 정보는 뉴스캐스터 목소리을 참조하세요.

각 Amazon Polly 음성 엔진에는 고유한 기능이 있습니다. Amazon Polly에서 제공하는 음성 엔진의 기능 및 지역 가용성에 대해 자세히 알아보십시오.

- 제너레이티브 보이스
- 롱폼 보이스
- 뉴럴 보이스
- 표준 음성

브랜드 보이스

이전 표에 나열된 사용 가능한 음성 외에도 Amazon Polly를 사용하여 브랜드 페르소나에 맞는 사용자 지정 음성을 구축할 수 있습니다. 브랜드 보이스를 사용하면 고객에게 독특하고 독점적인 목소리를 제공할 수 있습니다. Amazon Polly 브랜드 보이스에 대해 자세히 알아보려면 브랜드 보이스를 참조하십시오.

음성 속도

음성의 자연스러운 차이로 인해 사용 가능한 각 음성은 약간 다른 속도로 말합니다. 예를 들어 미국 영어 음성의 경우 Ivy와 Joanna는 Matthew보다 약간 빠르며 Joey보다 상당히 빠릅니다. 음성의 차이가 매우 크기 때문에 Amazon Polly 음성에 사용할 수 있는 표준 속도 (분당 단어 수) 는 없습니다. 하지만 음성 표시를 사용하여 선택한 텍스트를 음성으로 말하는 데 걸리는 시간을 확인할 수 있습니다.

음성 텍스트 구절의 길이를 측정하려면

1. 를 여세요 AWS CLI.
2. 다음 코드를 실행하고 필요에 따라 입력합니다.

```
aws polly synthesize-speech \
    --language-code optional language code if needed
    --output-format json \
    --voice-id [name of desired voice] \
    --text '[desired text]' \
    --speech-mark-types='["viseme"]' \
    LengthOfText.txt
```

3. `LengthOfText.txt`를 엽니다.

텍스트가 "Mary had a little lamb"이었던 경우, Amazon Polly가 반환하는 마지막 몇 줄은 다음과 같습니다.

```
{"time":882,"type":"viseme","value":"t"}
{"time":964,"type":"viseme","value":"a"}
{"time":1082,"type":"viseme","value":"p"}
```

마지막 viseme인, 본질적으로 "lamb"의 마지막 문자의 소리는 스피치가 시작된 후 1,082밀리초에 시작됩니다. 이는 결코 오디오 길이는 아니지만 그에 근접하며, 음성 간 비교의 기준이 될 수 있습니다.

음성 속도 변경

특정 애플리케이션에서 좋아하는 음성의 속도를 낮추거나 높이고자 할 수 있습니다. 음성 속도를 고려해야 할 경우를 위해 Amazon Polly는 SSML 태그를 사용하여 이를 수정하는 기능을 제공합니다. 예를 들어, 조직에서 이민자 대상 독자에게 책을 읽어주는 애플리케이션을 만들고 있다면 음성 속도를 다르게 설정하는 것이 좋습니다. 청중은 영어를 구사할 수 있지만 유창함은 제한적입니다. <prosody>Amazon Polly는 SSML 태그를 사용하여 음성 속도를 늦출 수 있도록 도와줍니다.

백분율을 사용할 수 있습니다.

```
<speak>
     In some cases, it might help your audience to <prosody rate="85%">slow
     the speaking rate slightly to aid in comprehension.</prosody>
</speak>
```

또는 사전 설정된 속도:

```
<speak>
     In some cases, it might help your audience to <prosody rate="slow">slow
     the speaking rate slightly to aid in comprehension.</prosody>
</speak>
```

Amazon Polly에서 SSML을 사용할 때 두 가지 속도 옵션을 사용할 수 있습니다.

- 사전 설정 속도: `x-slow`slow,`medium`,`fast`, 및`x-fast`. 이러한 경우 각 옵션의 속도는 근사값으로, 선호하는 음성에 따라 다릅니다. `medium` 옵션은 일반 음성 속도입니다.
- 음성 속도 n%: 20% ~ 200% 사이의 음성 속도 중 원하는 비율을 사용할 수 있습니다. 이러한 경우 원하는 정확한 속도를 선택할 수 있습니다. 그러나 실제 음성 속도는 근사값으로, 선택한 음성에 따라 다릅니다. 100%는 음성의 일반 속도입니다.

> **Note**
>
> 선택한 음성을 다양한 속도로 테스트하십시오. 각 옵션의 속도는 대략적인 것이며 선택한 음성에 따라 달라집니다.

`prosody`태그 사용에 대한 자세한 내용은 을 참조하십시오볼륨, 말하기 속도 및 피치 제어 .

이중 언어 음성

Amazon Polly에는 이중 언어 음성을 생성하는 두 가지 방법이 있습니다.

- 특유의 억양이 드러나는 이중 언어 음성
- 완벽한 이중 언어 음성

특유의 억양이 드러나는 이중 언어 음성

원하는 Amazon Polly를 사용하여 특유의 억양이 드러나는 이중 언어 음성을 생성할 수 있지만 반드시 SSML 태그를 사용해야 합니다.

일반적으로 입력 텍스트 안의 모든 단어를 사용 중인 지정한 기본 음성 언어로 읽습니다.

예를 들어, (미국 영어 원어민인) Joanna의 음성을 사용 중인 경우 Amazon Polly가 다음을 프랑스어 억양 없이 Joanna의 음성으로 읽습니다.

```
<speak>
     Why didn't she just say, 'Je ne parle pas français?'
</speak>
```

이 경우, Je ne parle pas français라는 단어를 영어처럼 읽습니다.

그러나 <lang> 태그와 함께 Joanna 음성을 사용하면 Amazon Polly는 그 문장을 Joanna 음성을 통해 미국식 억양의 프랑스어로 읽습니다.

```
<speak>
     Why didn't she just say, <lang xml:lang="fr-FR">'Je ne parle pas français?'</
lang>.
</speak>
```

Joanna의 음성은 원어민 프랑스어가 아니기 때문에 발음은 그녀의 모국인 미국 영어를 따릅니다. 예를 들면, 완벽한 프랑스어 발음에서는 단어 français에 목젖 울림 /R/이 있지만, Joanna의 미국식 영어 음성은 이 음소를 그에 해당하는 /r/ 소리로 발음합니다.

다음 텍스트에 이탈리아어 원어민인 Giorgio의 음성을 사용하면 Amazon Polly는 해당 문장을 Giorgio의 음성을 통해 이탈리아어식으로 발음합니다.

```
<speak>
     Mi piace Bruce Springsteen.
```

```
</speak>
```

완벽한 이중 언어 음성

Aditi 또는 Kajal(인도 영어 및 힌디어)과 같은 완벽한 이중 언어 음성은 2개 언어를 유창하게 말할 수 있습니다. 이렇게 하면 동일한 음성을 사용하여 단일 텍스트에서 두 언어의 단어 및 구를 사용할 수 있습니다.

현재 완전한 이중 언어를 구사하는 목소리는 Aditi, Kajal, Hala, Zayd뿐입니다.

이중 언어 음성(예: Aditi) 사용

Aditi는 인도 영어(en-IN)와 힌디어(hi-IN)를 유창하게 구사합니다. 영어와 힌디어로 스피치를 합성할 수 있으며 같은 문장 내에서도 두 언어 간에 음성을 전환할 수 있습니다.

힌디어는 두 가지 형태로 사용할 수 있습니다.

- Devanagari: "उसेन कहाँ, खेल तोह अब शुर होगा"
- Romanagari(라틴 알파벳 사용): "Usne kahan, khel toh ab shuru hoga"

또한 영어와 힌디어 중 하나 또는 둘 모두를 한 문장으로 조합할 수 있습니다.

- Devanagari + 영어: "This is the song कभी कभी अदिति"
- Romanagari + 영어: "This is the song from the movie Jaane Tu Ya Jaane Na."
- Devanagari + Romanagari + 영어: "This is the song कभी कभी अदिति from the movie Jaane Tu Ya Jaane Na."

Aditi는 이중 언어 음성입니다. Amazon Polly는 언어와 스크립트를 구분할 수 있으므로 이 모든 경우의 텍스트가 올바르게 읽혀집니다.

Amazon Polly는 영어(아라비아 숫자)와 힌디어(Devanagari 숫자)에서 숫자, 날짜, 시간 및 통화 확장을 지원합니다. 기본적으로 아라비아 숫자는 인도 영어로 읽힙니다. Amazon Polly에서 힌디어로 읽게 하려면 `hi-IN` 언어 코드 파라미터를 사용해야 합니다.

뉴스캐스터 목소리

사람의 말투는 상황에 따라 달라집니다. 가령 편안한 대화는 TV나 라디오 뉴스와 전혀 다르게 들립니다. 표준 음성이 만들어지는 방식으로 인해 다른 스타일을 만들어 낼 수 없습니다. 하지만 신경 음성에

서는 이것이 가능합니다. 특정 스타일을 훈련하고 변형하며 이 스타일에서 비롯되는 스피치의 일부분을 강조합니다.

Amazon Polly는 기본 신경 음성 외에도 신경 시스템을 사용하여 TV 또는 라디오 뉴스 진행자 말투의 음성을 생성하는 뉴스 진행자 스피치 스타일을 제공합니다. 뉴스 진행자 말투의 경우 미국 영어(en-US)는 Matthew 및 Joanna 음성으로 제공되고 미국 스페인어(es-US)는 Lupe 음성으로, 영국 영어(en-GB)는 Amy 음성으로 제공됩니다.

뉴스 진행자 스타일을 사용하려면 먼저 신경 엔진을 선택한 다음 입력 텍스트에서 다음 단계에 설명된 구문을 사용합니다.

> **Note**
>
> - 신경 말투를 사용하려면 신경 음성을 지원하는 AWS 리전 중 하나를 사용해야 합니다. 이 옵션은 모든 리전에서 사용할 수 없습니다. 자세한 내용은 기능 및 지역 호환성을(를) 참조하세요.

뉴스 진행자 말투를 적용하려면(콘솔)

1. https://console.aws.amazon.com/polly/에서 Amazon Polly 콘솔을 엽니다.
2. 뉴럴 보이스가 지원되는 AWS 지역을 사용하고 있는지 확인하세요.
3. 텍스트 투 스피치 페이지에서 엔진에 대해 신경을 선택합니다.
4. 사용하려는 언어 및 음성을 선택합니다. 뉴스 진행자 음성의 경우 미국 영어(en-US)는 Matthew 및 Joanna 음성으로만 제공되고 미국 스페인어(es-US)는 Lupe 음성으로, 영국 영어(en-GB)는 Amy 음성으로만 제공됩니다.
5. SSML을 활성화합니다.
6. Newscaster 스타일의 SSML 구문을 사용하여 text-to-speech 요청에 입력 텍스트를 추가하세요.

```
<amazon:domain name="news">text</amazon:domain>
```

예를 들어 다음과 같이 뉴스 진행자 태그를 사용할 수 있습니다.

```
<speak>
<amazon:domain name="news">
From the Tuesday, April 16th, 1912 edition of The Guardian newspaper:
```

```
The maiden voyage of the White Star liner Titanic, the largest ship ever launched
ended in disaster.

The Titanic started her trip from Southampton for New York on Wednesday. Late on
Sunday night she struck an iceberg off the Grand Banks of Newfoundland. By
wireless telegraphy she sent out signals of distress, and several liners were
near enough to catch and respond to the call.
</amazon:domain>
</speak>
```

7. 듣기를 선택합니다.

뉴스 진행자 말투를 적용하려면(CLI)

1. API 요청에 엔진 파라미터와 `neural` 값을 포함합니다.

```
  --engine neural
```

2. 뉴스 진행자 말투 SSML 구문을 사용하여 API 요청에 입력 텍스트를 추가합니다.

```
<amazon:domain name="news">text</amazon:domain>
```

예를 들어 다음과 같이 뉴스 진행자 태그를 사용할 수 있습니다.

```
<speak>
<amazon:domain name="news">
From the Tuesday, April 16th, 1912 edition of The Guardian newspaper:

The maiden voyage of the White Star liner Titanic, the largest ship ever launched
ended in disaster.

The Titanic started her trip from Southampton for New York on Wednesday. Late on
Sunday night she struck an iceberg off the Grand Banks of Newfoundland. By
wireless telegraphy she sent out signals of distress, and several liners were
near enough to catch and respond to the call.
</amazon:domain>
</speak>
```

SSML에 대한 자세한 내용은 지원되는 SSML 태그을(를) 참조하세요.

Amazon Polly의 언어

다음 언어는 Amazon Polly에서 지원되며 스피치를 합성하는 데 사용할 수 있습니다. 각 언어에는 고유한 언어 코드가 있습니다. 이러한 언어 코드는 W3C 언어 식별 태그 (언어 이름은 *ISO 639-3*, 국가 코드는 *ISO 3166*) 입니다.

Amazon Polly에서 제공하는 음소 및 비젬에 대한 세부 정보를 보려면 다음 표에서 언어를 선택하십시오.

	언어	언어 코드
1	아랍어	arb
2	아랍어(걸프)	ar-AE
3	카탈루냐어	ca-ES
4	중국어(광동어)	yue-CN
5	표준 중국어	cmn-CN
6	덴마크어	da-DK
7	네덜란드어(벨기에)	nl-BE
8	네덜란드어	nl-NL
9	영어(호주)	en-AU
10	영어(영국)	en-GB
11	영어(인도)	en-IN
12	영어(뉴질랜드)	en-NZ
13	영어(남아프리카)	en-ZA
14	영어(미국)	en-US
15	영어(웨일스)	en-GB-WLS

	언어	언어 코드
16	핀란드어	fi-FI
17	프랑스어	fr-FR
18	프랑스어(벨기에)	fr-BE
19	프랑스어(캐나다)	fr-CA
20	힌디어	hi-IN
21	독일어	de-DE
22	독일어(오스트리아)	de-AT
23	아이슬란드어	is-IS
24	이탈리아어	it-IT
25	일본어	ja-JP
26	한국어	ko-KR
27	노르웨이어	nb-NO
28	폴란드어	pl-PL
29	포르투갈어(브라질)	pt-BR
30	포르투갈어(유럽)	pt-PT
31	루마니아어	ro-RO
32	러시아어	ru-RU
33	스페인어(유럽)	es-ES
34	스페인어(멕시코)	es-MX
35	스페인어(미국)	es-US

	언어	언어 코드
36	스웨덴어	sv-SE
37	터키어	tr-TR
38	웨일스어	cy-GB

자세한 내용은 지원되는 언어의 음소 및 Viseme 표을(를) 참조하세요.

지원되는 언어의 음소 및 Viseme 표

다음 표에는 Amazon Polly에서 지원하는 언어의 음소와 예제 및 해당하는 viseme이 나와 있습니다.

주제

- 아랍어(arb)
- 아랍어(걸프)(ar-AE)
- 카탈루냐어(ca-ES)
- 중국어(광동어)(yue-CN)
- 표준 중국어(cmn-CN)
- 덴마크어(da-DK)
- 네덜란드어(벨기에)(nl-BE)
- 네덜란드어(nl-NL)
- 영어(미국)(en-US)
- 영어(호주)(en-AU)
- 영어(영국)(en-GB)
- 영어(인도)(en-IN)
- 영어(아일랜드)(en-IE)
- 영어(뉴질랜드)(en-NZ)
- 영어(남아프리카)(en-ZA)
- 영어(웨일스)(en-GB-WLS)
- 핀란드어(fi-FI)

- 프랑스어(fr-FR)
- 프랑스어(벨기에)(fr-BE)
- 프랑스어(캐나다)(fr-CA)
- 독일어(de-DE)
- 독일어(오스트리아)(de-AT)
- 힌디어(hi-IN)
- 아이슬란드어(is-IS)
- 이탈리아어(it-IT)
- 일본어(ja-JP)
- 한국어(ko-KR)
- 노르웨이어(nb-NO)
- 폴란드어(pl-PL)
- 포르투갈어(pt-PT)
- 포르투갈어(브라질)(pt-BR)
- 루마니아어(ro-RO)
- 러시아어(ru-RU)
- 스페인어(es-ES)
- 스페인어(멕시코)(es-MX)
- 스페인어(미국)(es-US)
- 스웨덴어(sv-SE)
- 터키어(tr-TR)
- 웨일스어(cy-GB)

아랍어(arb)

다음 표에는 Amazon Polly에서 지원하는 Zeina의 아랍어 음성에 대해 IPA(International Phonetic Alphabet) 음소와 X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet) 기호 및 해당하는 viseme이 나와 있습니다.

음소/Viseme 표

IPA	X-SAMPA	설명	예	Viseme
자음				
ʔ	?	성문 폐쇄음	أنا	
ʕ	?\	유성 인두 마찰음	عُمَر	k
b	b	유성 양순 파열음	بَلَد	p
d	d	유성 치경 파열음	داري	t
dˤ	d_?\	강조 유성 치경 파열음	ضَوء	t
͡dʒ	dZ	유성 후치경 파찰음	جَميل	S
ð	D	유성 치 마찰음	ذلِكَ	T
ðˤ	D_?\	강조 유성 치 마찰음	ظَلام	T
f	f	무성 순치 마찰음	فَصل	f
g	g	유성 연구개 파열음	إنجلترا	k
ɣ	G	유성 연구개 마찰음	غَرب	k
h	h	무성 성문 마찰음	هذا	k
j	j	경구개 접근음	يَمشي	i
k	k	무성 연구개 파열음	كَلب	k
l	l	치경 설측 접근음	لاقى	t
lˠ	l_G	강조 치경 설측 접근음	عبدالله	t
m	m	양순 비음	ماذا	p

IPA	X-SAMPA	설명	예	Viseme
n	n	치경 비음	نور	t
p	p	무성 양순 파열음	سبَخ	p
q	q	무성 구개수 파열음	بيرَق	k
r	r	치경 전동음	لمَر	r
s	s	무성 치경 마찰음	السُّؤال	s
sˤ	s_?\	강조 무성 치경 마찰음	بحاص	s
ʃ	S	무성 후치경 마찰음	شُكر	S
t	t	무성 치경 파열음	تَمر	t
tˤ	t_?\	강조 무성 치경 파열음	بالط	t
θ	T	무성 치 마찰음	ثَلاث	T
v	v	유성 순치 마찰음	فيتامين	f
w	w	양순 연구개 접근음	وَلَد	u
x	x	무성 연구개 마찰음	فَوْخ	k
ħ	X\	무성 인두 마찰음	لَوْحَ	k
z	z	유성 치경 마찰음	زُور	s
모음				
a	a	전설 비원순 저모음	بَرد	a
a:	a:	전설 비원순 저장모음	دار	a

IPA	X-SAMPA	설명	예	Viseme
ɑˤ	A_?\	강조 후설 비원순 저모음	طَبل	a
ɑˤː	A_?\:	강조 후설 비원순 저장모음	ظالِم	a
u	u	후설 원순 고모음	بُرش	u
uː	uː	후설 원순 고장모음	سور	u
uˤ	u_?\	강조 후설 원순 고모음	بُدّ	u
uˤː	u_?\:	강조 후설 원순 고장모음	طول	u
i	i	전설 비원순 고모음	بِنت	i
iː	iː	전설 비원순 고장모음	حَزين	i
iˤ	i_?\	강조 전설 비원순 고모음	ضِدّ	i
iˤː	i_?\:	강조 전설 비원순 고장모음	يِضام	i
e	e	전설 비원순 중고모음	تِكرام	e
eː	eː	전설 비원순 중고장모음	لِيدوم	e
ɔ	O	후설 원순 중저모음	يِجولونكت	O
ɔː	Oː	후설 원순 중저장모음	نويزفيلت	O

아랍어(걸프)(ar-AE)

다음 표에는 Amazon Polly에서 지원하는 Hala의 아랍어 음성에 대해 IPA(International Phonetic Alphabet) 음소와 X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet) 기호 및 해당하는 viseme이 나와 있습니다.

음소/Viseme 표

IPA	X-SAMPA	설명	예	발음	Viseme
자음					
b	b	유성 양순 파열음	دلب	/ " b a . l a d /	b
d	d	유성 치경 파열음	درد	/ " r a d d /	d
dˤ	d_?\	인두 유성 치경 파열음	وضء	/ " d_?\ a w ? /	D
f	f	무성 순치 마찰음	فرن	/ " f I . r I n /	f
g	g	유성 연구개 파열음	قال	/ " g a: l /	k
j	j	유성 경구개 접근음	يشمي	/ " j I m . S i: /	i
k	k	무성 연구개 파열음	كامل	/ " k a: . m i l /	k
l	l	유성 치경 설측 접근음	ليل	/ " l e: l /	t
lˤ	l_G	인두 유성 치경 설측 접근음	هلالبع	/ ?\ a b . " d A_?\ l_G . l_G A_?\ /	t

IPA	X-SAMPA	설명	예	발음	Viseme
m	m	양순 비강 폐쇄음	ةئم	/ " m I j . j a /	p
n	n	치경 비강 폐쇄음	رون	/ " n u: r /	t
p	p	무성 양순 파열음	أرابو	/ " ? O . p e . r a: /	p
q	q	무성 구개수 파열음	رصق	/ " q A_?\ s_?\ r /	k
r	r	치경 전동음	لمر	/ " r a . m I l /	r
s	s	무성 치경 마찰음	مسمس	/ " s I m . s I m /	s
sˤ	s_?\	인두 무성 치경 마찰음	بحاص	/ " s_?\ A_?: . X \ I b /	s
t	t	무성 치경 파열음	رمت	/ "t a . m a r /	t
tˤ	t_?\	인두 무성 치경 마찰음	بلاط	/ " t_?\ A_?: . l I b /	t
v	v	유성 순치 마찰음	نيماتيف	/ v i: . t A . " m i: n /	f
w	w	유성 순구개 접근음	دياو	/ " w a: . j I d /	u
x	x	무성 연구개 마찰음	فوروخ	/ x a . " r u: f /	k
z	z	무성 연구개 마찰음	زهور	/ " z h u: r /	s

IPA	X-SAMPA	설명	예	발음	Viseme
ð	D	유성 치간 마찰음	كلذ	/ " D a: . l l k /	D
ðˤ	D_?\	인두 유성 치간 마찰음	ماظلا	/ D_?\ A_?\ . " l a: m /	D
ħ	X\	무성 인두 마찰음	نيجلا	/ ? a l . " X\ i: n /	k
ŋ	N	연구개 비강 폐쇄음	غنوكغنوه	/ h O N . " k O N g /	k
ɣ	G	유성 연구개 마찰음	ةبيرغ	/ G I . " r i: . b a /	k
ʃ	S	무성 후치경 마찰음	سمش	/ " S a m s /	S
ʒ	Z	유성 후치경 마찰음	تيكاج	/ Z a . " k e: t /	S
ʔ	?	성문 폐쇄음	ةسسؤم	/ m u . " ? a s . s a . s a /	
ʕ	?\	유성 인두 마찰음	امع	/ " ?\ a: m m /	k
ʤ	dZ	유성 후치경 파찰음	ةعماج	/ " dZ a: m . ?\ a /	S
θ	T	유성 치간 마찰음	ةثالث	/ T a . " l a: . T a /	T
ɦ	h	유성 성문 마찰음	الله	/ " h l a: l /	k
모음					

IPA	X-SAMPA	설명	예	발음	Viseme
æ	a	전중설 비원순 단모음	رفس	/ " s a . f a r /	a
ɑˤ	A_?\	인두 후설 비원순 단모음	بلص	/ " s_?\ A_?\ l b /	a
æː	a:	전중설 비원순 장모음	باب	/ " b a: b /	a
ɑˤː	A_?\:	인두 후설 비원순 장모음	جضان	/ " n A_?: . D_? \ i_?\ dZ /	a
a	A	중설 비원순 단모음	wifi	/ " w A j . f A j /	a
i	i	강전설 비원순 단모음(MSA)	قاحسإ	/ ? i s . " X\ A_? \: q /	i
ɪ	I	약전설 비원순 단모음	تنب	/ " b I n t /	i
iˤ	i_?\	인두 전설 비원순 단모음	لفط	/ " t_?\ i_?\ f I l /	i
iː	i:	전설 비원순 장모음	ليبس	/ s a . " b i: l /	i
iˤː	i_?:	인두 전설 비원순 장모음	بيطر	/ r A_?\ . " t_?\ i_?: b /	i
u	u	강후설 비원순 단모음(MSA)	عرتخم	/ " m u x . t a . r i ?\ /	u
ʊ	U	약후설 원순 단모음	موسر	/ r U . " s u: m /	u
uˤ	u_?\	인두 후설 원순 단모음	روفصع	/ ?\ u_?\ s_?\ . " f u: r /	u

IPA	X-SAMPA	설명	예	발음	Viseme
u:	u:	후설 원순 장모음	توت	/ " t u: t /	u
uˤː	u_?\:	인두 후설 원순 장모음	رصو	/ " s_?\ u_?\: r /	u
e	e	중전설 비원순 단모음	إِنْتَرْنِت	/ " s e n t /	e
e:	e:	중전설 비원순 장모음	إيش	/ " ? e: S /	e
ɔ	O	중후설 원순 단모음	دولار	/ d O . " l A r /	O
ɔː	O:	중후설 원순 장모음	لون	/ " l O: n /	O

카탈루냐어(ca-ES)

다음 표에는 Amazon Polly에서 지원하는 Arlet의 카탈루냐어 음성에 대해 IPA(International Phonetic Alphabet) 음소와 X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet) 기호 및 해당하는 viseme이 나와 있습니다.

음소/Viseme 표

IPA	X-SAMPA	설명	예	Viseme
자음				
p	p	무성 양순 파열음	ploure	p
t	t	무성 치경 파열음	Tarragona	t
k	k	무성 연구개 파열음	com	k
b	b	유성 양순 파열음	bata	p

IPA	X-SAMPA	설명	예	Viseme
d	d	유성 치경 파열음	endoll	t
g	g	유성 연구개 파열음	gros	k
m	m	유성 양순 비음	manera	p
n	n	유성 치경 비음	donar	t
ɲ	J	유성 경구개 비음	any	J
ŋ	N	유성 연구개 비음	pingüí	k
ɬ	5	유성 연구개 치경 설측 접근음(다크 l)	albercoc	l
ʎ	L	유성 경구개 설측 접근음	llop	J
r	r	유성 치경 전동음	parra	r
ɾ	4	유성 치경 타음	para	t
f	f	무성 순치 마찰음	èmfasi	f
s	s	무성 치경 마찰음	sac	s
z	z	유성 치경 마찰음	calzes	s
ʃ	S	무성 후치경 마찰음	guix	S
ʒ	Z	유성 후치경 마찰음	col·legi	S
͡tʃ	tS	무성 후치경 파찰음	cotxe	S
͡dʒ	dZ	유성 후치경 파찰음	platja	S
β	B	유성 양순 접근음	obert	B
ð	D	유성 치 접근음	bedoll	T

IPA	X-SAMPA	설명	예	Viseme
j	j	유성 경구개 접근음	noia	i
ɣ	G	유성 연구개 접근음	pega	k
v	v	유성 순치 마찰음	afgà	f
w	w	유성 순구개 접근음	aigua	u
x	x	무성 연구개 마찰음	Jiménez	k
ʝ	j\	유성 경구개 마찰음	yeso	J
l	l	유성 치경 설측 접근음	alondra	t
θ	T	무성 치 마찰음	González	T
모음				
a	a	후설 모음	casa	a
e	e	전설 비원순 중고모음	llenya	e
ɛ	E	전설 비원순 중저모음	xec	E
i	i	전설 비원순 모음	visca	i
o	o	후설 원순 중고모음	gos	o
ɔ	O	후설 원순 중저모음	joc	O
u	u	후설 원순 모음	un	u
ə	@	중설 중모음	casa	@
추가 기호				
ˈ	"	제1강세	Alabama	

IPA	X-SAMPA	설명	예	Viseme
ˈ	%	제2강세	Alabama	
.	.	음절 경계	A.la.ba.ma	

중국어(광동어)(yue-CN)

다음 표에는 Amazon Polly에서 지원하는 중국어(광동어) 음성에 대한 병음 및 IPA(International Phonetic Alphabet) 음소가 나열되어 있습니다. 병음은 학계와 광동어 사용자 사이에서 일반적으로 사용되는 광동어의 로마자 표기법입니다. IPA와 X-SAMPA는 일반적으로 사용되지는 않지만 영어 지원에 사용 가능합니다. 표에 있는 IPA 및 X-SAMPA 기호는 참조용이며 중국어 표기에 사용하면 안 됩니다. 병음의 예와 그에 해당하는 viseme도 표시됩니다.

Amazon Polly가 병음과 함께 철자 발음을 사용하도록 하려면 `phoneme alphabet="x-amazon-jyutping"` 태그를 사용하세요.

다음 예에서는 각 표준과 함께 이 작업을 수행하는 방법을 보여 줍니다.

병음:

```
<speak>
     ## <phoneme alphabet="x-amazon-jyutping" ph="sing2">#</phoneme>#
     ## <phoneme alphabet="x-amazon-jyutping" ph="seng2">#</phoneme>#
</speak>
```

IPA:

```
<speak>
     ## <phoneme alphabet="ipa" ph="p##k##n">pecan</phoneme>#
     ## <phoneme alphabet="ipa" ph="#pi.kæn">pecan</phoneme>#
</speak>
```

X-SAMPA:

```
<speak>
     ## <phoneme alphabet='x-sampa' ph='pI"kA:n'>pecan</phoneme>#
     ## <phoneme alphabet='x-sampa' ph='"pi.k{n'>pecan</phoneme>#
</speak>
```

> **Note**
>
> Amazon Polly는 UTF-8 형식으로 인코딩된 광동어 입력만 허용합니다.

음소/Viseme 표

병음	IPA	X-SAMPA	설명	병음 예제	Viseme
자음					
b	p	p	무성 양순 파열음	巴, baa1	p
c	ts^h	ts_h	기식 무성 치경 파찰음	叉, caa1	s
d	t	t	무성 치경 파열음	打, daa2	t
f	f	f	무성 순치 마찰음	花, faa1	f
g	k	k	무성 연구개 파열음	家, gaa1	k
gw	k^w	k_w	순음 무성 연구개 파열음	瓜, gwaa1	u
h	h	h	무성 성문 마찰음	哈, haa1	k
k	k^h	k_h	기식 무성 연구개 파열음	卡, kaa1	k
kw	k^{wh}	k_wh	순음 흡기식 무성 연구개 파열음	誇, kwaa1	u
l	l	l	치경 설측 접근음	啦, laa1	t
m	m	m	양순 비음	媽, maa1	p
m	m	m=	성절 양순 비음	唔, m4	p
ng	ŋ	N	연구개 비음	牙, ngaa4	k
ng	ŋ	N=	성절 연구개 비음	吳, ng4	k
n	n	n	치경 비음	拿, naa4	t

병음	IPA	X-SAMPA	설명	병음 예제	Viseme
p	pʰ	p_h	기식 무성 양순 파열음	趴, paa1	p
s	s	s	무성 치경 마찰음	沙, saa1	s
t	tʰ	t_h	기식 무성 치경 파열음	他, taa1	t
w	w	w	양순 연구개 접근음	娃, waa1	u
y	j	j	경구개 접근음	也, jaa5	i
z	ts	ts	무성 치경 파찰음	渣, zaa1	s
모음					
a	ɐ	6	중설 근저모음	吉, gat1	a
aa	ɑ	A	후설 비원순 저모음	家, gaa1	a
aai	ɑi	Ai	이중모음	街, gaai1	a
aau	ɑu	Au	이중모음	交, gaau1	a
ai	ɐi	6i	이중모음	雞, gai1	a
au	ɐu	6u	이중모음	溝, kau1	a
e	ɛ	E	전설 비원순 중저모음	爹, de1	E
ei	ei	ei	이중모음	基, gei1	e
eo	ɵ	8	중설 원순 중고모음	春, ceon1	o
eoi	ɵy	8y	이중 모음	居, geoi1	o
eu	ɛu	Eu	이중 모음	掉垃圾의 掉, deu6	E
i	i	i	전설 비원순 고모음	斯, si1	i

병음	IPA	X-SAMPA	설명	병음 예제	Viseme
i	I	I	근전설 비원순 근고모음	激, gik1	i
iu	iu	iu	이중 모음	驕, giu1	i
o	ɔ	O	후설 원순 중저모음	哥, go1	O
oe	œ	9	전설 원순 중저모음	鋸, goe3	O
oi	ɔi	Oi	이중모음	該, goi1	O
ou	ou	ou	이중모음	高, gou1	o
u	u	u	후설 원순 고모음	姑, gu1	u
u	ʊ	U	근후설 원순 근고모음	谷, guk5	u
ui	ui	ui	이중모음	劾, gui6	u
yu	y	y	전설 원순 고모음	於, jyu1	u
톤 표시 및 추가 기호					
1			높은 수준	詩, si1	
2			중간으로 올라감	史, si2	
3			중간 수준	試, si3	
4			매우 낮은 수준	時, si4	
5			낮게 올라감	市, si5	
6			낮은 수준	是, si6	
-	.	.	음절 경계	語音 jyu5-jam1	

표준 중국어(cmn-CN)

다음 표에는 Amazon Polly에서 지원하는 표준 중국어 음성에 대한 병음 및 IPA(International Phonetic Alphabet) 음소가 나열되어 있습니다. 병음은 표준 중국어 로마자 표기법의 국제 표준입니다. IPA와 X-SAMPA는 일반적으로 사용되지는 않지만 영어 지원에 사용 가능합니다. 표에 있는 IPA 및 X-SAMPA 기호는 참조용이며 중국어 표기에 사용하면 안 됩니다. 병음의 예와 그에 해당하는 viseme도 표시됩니다.

Amazon Polly에서 음운 발음과 병음을 함께 사용하게 하려면 `phoneme alphabet="x-amazon-`*`phonetic standard used`*`"` 태그를 사용하세요.

다음 예에서는 각 표준과 함께 이 작업을 수행하는 방법을 보여 줍니다.

병음:

```
<speak>
     ## <phoneme alphabet="x-amazon-pinyin" ph="bo2">#</phoneme>#
     ## <phoneme alphabet="x-amazon-pinyin" ph="bao2">#</phoneme>#
</speak>
```

IPA:

```
<speak>
     ## <phoneme alphabet="ipa" ph="p##k##n">pecan</phoneme>#
     ## <phoneme alphabet="ipa" ph="#pi.kæn">pecan</phoneme>#
</speak>
```

X-SAMPA:

```
<speak>
     ## <phoneme alphabet='x-sampa' ph='pI"kA:n'>pecan</phoneme>#
     ## <phoneme alphabet='x-sampa' ph='"pi.k{n'>pecan</phoneme>#
</speak>
```

> **Note**
>
> Amazon Polly는 UTF-8로 인코딩된 표준 중국어 입력만 허용합니다. GB 18030 인코딩 표준은 현재 Amazon Polly에서 지원되지 않습니다.

음소/Viseme 표

병음	IPA	X-SAMPA	설명	병음 예	Viseme
자음					
f	f	f	무성 순치 마찰음	发, fa1	f
h	h	h	무성 성문 마찰음	和, he2	k
g	k	k	무성 연구개 파열음	古, gu3	k
k	kʰ	k_h	기식 무성 연구개 파열음	苦, ku3	k
l	l	l	치경 설측 접근음	拉, la1	t
m	m	m	양순 비음	骂, ma4	p
n	n	n	치경 비음	那, na4	t
ng	ŋ	N	연구개 비음	正, zheng4	k
b	p	p	무성 양순 파열음	爸, ba4	p
p	pʰ	p_h	기식 무성 양순 파열음	怕, pa4	p
s	s	s	무성 치경 마찰음	四, si4	s
x	ɕ	s\	무성 치경구개 마찰음	西, xi1	J
sh	ʂ	s`	무성 권설 마찰음	是, shi4	S
d	t	t	무성 치경 파열음	打, da3	t
t	tʰ	t_h	기식 무성 치경 파열음	他, ta1	t
zh	‿tʂ	t`s`	무성 권설 파찰음	之, zhi1	S
ch	‿tʂʰ	t`s`_h	기식 무성 권설 파찰음	吃, chi1	S
s	‿ts	ts	무성 치경 파찰음	字, zi4	s

병음	IPA	X-SAMPA	설명	병음 예	Viseme
j	͡tɕ	ts\	무성 치경구개 파찰음	鸡, ji1	J
q	͡tɕʰ	ts_h	기식 무성 치경구개 경파찰음	七, qi1	J
c	͡tsʰ	ts_h	기식 무성 치경 파찰음	次, ci4	s
w	w	w	양순 연구개 접근음	我, wo3	u
r	ʐ	z`	유성 권설 마찰음	日, ri4	S
"er" 및 "r" 음색을 띤 음절					
er	ɚ	@`	r 음색을 띤 중설 중모음	二, er4	@
-r			r 음색을 띤 음절	馅儿, xianr4	@
모음					
e	ɤ	7	후설 비원순 중고모음	恶, e4	e
e	ə	@	중설 중모음	恩, en1	@
a	a	a	전설 비원순 저모음	安, an1	a
ai	aɪ	aI	이중 모음	爱, ai4	a
ao	aʊ	aU	이중 모음	奥, ao4	a
ei	eɪ	e	이중 모음	诶, ei4	e
e	ɛ	E	전설 비원순 중저모음	姐, jie3	E
i	i	i	전설 비원순 고모음	鸡, ji1	i
ou	oʊ	oU	이중 모음	欧, ou1	o
o	ɔ	O	후설 원순 중저모음	哦, o4	o

병음	IPA	X-SAMPA	설명	병음 예	Viseme
u	u	u	후설 원순 고모음	主, zhu3	u
yu	y	y	전설 원순 고모음	于, yu2	u
톤 표시 및 추가 기호					
1			높은 톤	淤, yu1	
2			올라가는 톤	鱼, yu2	
3			낮은(떨어지는-올라가는) 톤	语, yu3	
4			떨어지는 톤	育, yu4	
0			중간 톤	的, de0	
-	.	.	음절 경계	语音 yu3-yin1	

덴마크어(da-DK)

다음 표에는 Amazon Polly에서 지원하는 덴마크어 음성에 대해 IPA(International Phonetic Alphabet) 음소와 X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet) 기호 및 해당하는 viseme이 나와 있습니다.

음소/Viseme 표

IPA	X-SAMPA	설명	예	Viseme
자음				
b	b	유성 양순 파열음	bat	p
d	d	유성 치경 파열음	da	t
ð	D	유성 치 마찰음	mad, thriller	T

IPA	X-SAMPA	설명	예	Viseme
f	f	무성 순치 마찰음	fat	f
g	g	유성 연구개 파열음	gat	k
h	h	무성 성문 마찰음	hat	k
j	j	경구개 접근음	jo	i
k	k	무성 연구개 파열음	kat	k
l	l	치경 설측 접근음	ladt	t
m	m	양순 비음	mat	p
n	n	치경 비음	nay	t
ŋ	N	연구개 비음	lang	k
p	p	무성 양순 파열음	pande	p
r	r	치경 전동음	thriller, story	r
ʁ	R	유성 구개수 마찰음	rat	k
s	s	무성 치경 마찰음	sat	s
t	t	무성 치경 파열음	tal	t
v	v	유성 순치 마찰음	vat	f
w	w	양순 연구개 접근음	hav, weekend	u
모음				
ø	2	전설 원순 중고모음	øst	o
ø:	2:	전설 원순 중고장모음	øse	o
ɐ	6	중설 근저모음	mor	a

IPA	X-SAMPA	설명	예	Viseme
œ	9	전설 원순 중저모음	skøn, grønt	O
œ:	9:	전설 원순 중저장모음	høne, gøre	O
ə	@	중설 중모음	ane	@
æ:	{:	전설 비원순 근저장모음	male	a
a	a	전설 비원순 저모음	man	a
æ	{	전설 비원순 근저모음	adresse	a
ɑ	A	후설 비원순 저모음	lak, tak	a
ɑ:	A:	후설 비원순 저장모음	rase	a
e	e	전설 비원순 중고모음	midt	e
e:	e:	전설 비원순 중고장모음	mele	e
ɛ	E	전설 비원순 중저모음	mæt	E
ɛ:	E:	전설 비원순 중저장모음	mæle	E
i	i	전설 비원순 고모음	mit	i
i:	i:	전설 비원순 고장모음	mile	i
o	o	후설 원순 중고모음	foto	o

IPA	X-SAMPA	설명	예	Viseme
o:	o:	후설 원순 중고장모음	mole	o
ɔ	O	후설 원순 중저모음	mund	O
ɔ:	O:	후설 원순 중저장모음	måle	O
ɒ:	Q:	후설 원순 저장모음	morse	O
u	u	후설 원순 고모음	lusk	u
u:	u:	후설 원순 고장모음	mule	u
ʌ	V	후설 평순 중저	kører	E
y	y	전설 원순 고모음	yt	u
y:	y:	전설 원순 고장모음	hyle	u
추가 기호				
ˈ	"	제1강세	Alabama	
ˌ	%	제2강세	Alabama	
.	.	음절 경계	A.la.ba.ma	

네덜란드어(벨기에)(nl-BE)

다음 표에는 Amazon Polly에서 지원하는 벨기에 네덜란드어(플랑드르) 음성에 대해 IPA(International Phonetic Alphabet) 음소와 X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet) 기호 및 해당하는 viseme이 나와 있습니다.

음소/Viseme 표

IPA	X-SAMPA	설명	예	Viseme
자음				
b	b	유성 양순 파열음	bak	p
d	d	유성 치경 파열음	dak	t
ʤ	dZ	유성 후치경 파찰음	manager	S
f	f	무성 순치 마찰음	fel	f
g	g	유성 연구개 파열음	goal	k
ɣ	G	유성 연구개 마찰음	hoed	k
ɦ	h\	유성 성문 마찰음	hand	k
j	j	경구개 접근음	ja	i
k	k	무성 연구개 파열음	kap	k
l	l	치경 설측 접근음	land	t
m	m	양순 비음	met	p
n	n	치경 비음	net	t
ŋ	N	연구개 비음	bang	k
p	p	무성 양순 파열음	pak	p
r	r	치경 전동음	rand	r
s	s	무성 치경 마찰음	sein	s
ʃ	S	무성 후치경 마찰음	show	S
t	t	무성 치경 파열음	tak	t
v	v	유성 순치 마찰음	vel	f

IPA	X-SAMPA	설명	예	Viseme
ʋ	v\	순치 접근음	wit	f
x	x	무성 연구개 마찰음	toch	k
z	z	유성 치경 마찰음	ziin	s
ʒ	Z	유성 후치경 마찰음	bagage	S
모음				
øː	2:	전설 원순 중고장모음	neus	o
œy	9y	이중모음	buit	O
ə	@	중설 중모음	de	@
aː	a:	전설 비원순 저장모음	baad	a
ɑː	A	후설 비원순 저모음	bad	a
eː	e:	전설 비원순 중고장모음	beet	e
ɜː	3:	중설 비원순 중저장모음	barrière	E
ɛ	E	전설 비원순 중저모음	bed	E
ɛi	Ei	이중모음	beet	E
i	i	전설 비원순 고모음	vier	i
ɪ	I	근전설 비원순 근고모음	pit	i

IPA	X-SAMPA	설명	예	Viseme
oː	o:	후설 원순 중고장모음	boot	o
ɔ	O	후설 원순 중저모음	pot	O
u	u	후설 원순 고모음	hoed	u
ʌu	Vu	이중모음	fout	E
yː	y:	전설 원순 고장모음	fuut	u
ʏ	Y	근전설 원순 근고모음	hut	u
추가 기호				
ˈ	"	제1강세	Alabama	
ˌ	%	제2강세	Alabama	
.	.	음절 경계	A.la.ba.ma	

네덜란드어(nl-NL)

다음 표에는 Amazon Polly에서 지원하는 네덜란드어 음성에 대해 IPA(International Phonetic Alphabet) 음소와 X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet) 기호 및 해당하는 viseme이 나와 있습니다.

음소/Viseme 표

IPA	X-SAMPA	설명	예	Viseme
자음				
b	b	유성 양순 파열음	bak	p
d	d	유성 치경 파열음	dak	t

IPA	X-SAMPA	설명	예	Viseme
ʤ	dZ	유성 후치경 파찰음	manager	S
f	f	무성 순치 마찰음	fel	f
g	g	유성 연구개 파열음	goal	k
ɣ	G	유성 연구개 마찰음	hoed	k
ɦ	h\	유성 성문 마찰음	hand	k
j	j	경구개 접근음	ja	i
k	k	무성 연구개 파열음	kap	k
l	l	치경 설측 접근음	land	t
m	m	양순 비음	met	p
n	n	치경 비음	net	t
ŋ	N	연구개 비음	bang	k
p	p	무성 양순 파열음	pak	p
r	r	치경 전동음	rand	r
s	s	무성 치경 마찰음	sein	s
ʃ	S	무성 후치경 마찰음	show	S
t	t	무성 치경 파열음	tak	t
v	v	유성 순치 마찰음	vel	f
ʋ	v\	순치 접근음	wit	f
x	x	무성 연구개 마찰음	toch	k
z	z	유성 치경 마찰음	ziin	s

IPA	X-SAMPA	설명	예	Viseme
ʒ	Z	유성 후치경 마찰음	bagage	S
모음				
øː	2:	전설 원순 중고장모음	neus	o
œy	9y	이중모음	buit	O
ə	@	중설 중모음	de	@
aː	a:	전설 비원순 저장모음	baad	a
ɑː	A	후설 비원순 저모음	bad	a
eː	e:	전설 비원순 중고장모음	beet	e
ɜː	3:	중설 비원순 중저장모음	barrière	E
ɛ	E	전설 비원순 중저모음	bed	E
ɛi	Ei	이중모음	beet	E
i	i	전설 비원순 고모음	vier	i
ɪ	I	근전설 비원순 근고모음	pit	i
oː	o:	후설 원순 중고장모음	boot	o
ɔ	O	후설 원순 중저모음	pot	O
u	u	후설 원순 고모음	hoed	u

IPA	X-SAMPA	설명	예	Viseme
ʌu	Vu	이중모음	fout	E
y:	y:	전설 원순 고장모음	fuut	u
ʏ	Y	근전설 원순 근고모음	hut	u
추가 기호				
ˈ	"	제1강세	Alabama	
ˌ	%	제2강세	Alabama	
.	.	음절 경계	A.la.ba.ma	

영어(미국)(en-US)

다음 표에는 Amazon Polly에서 지원하는 미국 영어 음성에 대해 IPA(International Phonetic Alphabet) 음소와 X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet) 기호 및 해당하는 viseme이 나와 있습니다.

음소/Viseme 표

IPA	X-SAMPA	설명	예	Viseme
자음				
b	b	유성 양순 파열음	bed	p
d	d	유성 치경 파열음	dig	t
ʤ	dZ	유성 후치경 파찰음	jump	S
ð	D	유성 치 마찰음	then	T
f	f	무성 순치 마찰음	five	f
g	g	유성 연구개 파열음	game	k

IPA	X-SAMPA	설명	예	Viseme
h	h	무성 성문 마찰음	house	k
j	j	경구개 접근음	yes	i
k	k	무성 연구개 파열음	cat	k
l	l	치경 설측 접근음	lay	l
m	m	양순 비음	mouse	p
n	n	치경 비음	nap	t
ŋ	N	연구개 비음	thing	k
p	p	무성 양순 파열음	speak	p
ɹ	r\	치경 접근음	red	r
s	s	무성 치경 마찰음	seem	s
ʃ	S	무성 후치경 마찰음	ship	S
t	t	무성 치경 파열음	trap	t
͡tʃ	tS	무성 후치경 파찰음	chart	S
θ	T	무성 치 마찰음	thin	T
v	v	유성 순치 마찰음	vest	f
w	w	양순 연구개 접근음	west	u
z	z	유성 치경 마찰음	zero	s
ʒ	Z	유성 후치경 마찰음	vision	S
모음				
ə	@	중설 중모음	arena	@

IPA	X-SAMPA	설명	예	Viseme
ɚ	@`	중설 치경 접근 수반음	reader	@
æ	{	전설 비원순 근저모음	trap	a
aɪ	aI	이중 모음	price	a
aʊ	aU	이중 모음	mouth	a
ɑ	A	후설 비원순 저장모음	father	a
eɪ	eI	이중 모음	face	e
ɝ	3`	중설 비원순 중저 치경 접근 수반음	nurse	E
ɛ	E	전설 비원순 중저모음	dress	E
i	i	전설 비원순 고장모음	fleece	i
ɪ	I	근전설 비원순 근고모음	kit	i
oʊ	oU	이중 모음	goat	o
ɔ	O	후설 원순 중저장모음	thought	O
ɔɪ	OI	이중 모음	choice	O
u	u	후설 원순 고장모음	goose	u
ʊ	U	근후설 원순 근고모음	foot	u

IPA	X-SAMPA	설명	예	Viseme
ʌ	V	open-mid-back 반올림되지 않은 모음	strut	E
추가 기호				
ˈ	"	제1강세	Alabama	
ˌ	%	제2강세	Alabama	
.	.	음절 경계	A.la.ba.ma	

영어(호주)(en-AU)

다음 표에는 Amazon Polly에서 지원하는 호주 영어 음성에 대해 IPA(International Phonetic Alphabet) 음소와 X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet) 기호 및 해당하는 viseme이 나와 있습니다.

음소/Viseme 표

IPA	X-SAMPA	설명	예	Viseme
자음				
b	b	유성 양순 파열음	bed	p
d	d	유성 치경 파열음	dig	t
ʤ	dZ	유성 후치경 파찰음	jump	S
ð	D	유성 치 마찰음	then	T
f	f	무성 순치 마찰음	five	f
g	g	유성 연구개 파열음	game	k
h	h	무성 성문 마찰음	house	k
j	j	경구개 접근음	yes	i

IPA	X-SAMPA	설명	예	Viseme
k	k	무성 연구개 파열음	cat	k
l	l	치경 설측 접근음	lay	t
l̩	l=	성절 치경 설측 접근음	battle	t
m	m	양순 비음	mouse	p
m̩	m=	성절 양순 비음	anthem	p
n	n	치경 비음	nap	t
n̩	n=	성절 치경 비음	button	t
ŋ	N	연구개 비음	thing	k
p	p	무성 양순 파열음	pin	p
ɹ	r\	치경 접근음	red	r
s	s	무성 치경 마찰음	seem	s
ʃ	S	무성 후치경 마찰음	ship	S
t	t	무성 치경 파열음	task	t
t͡ʃ	tS	무성 후치경 파찰음	chart	S
Θ	T	무성 치 마찰음	thin	T
v	v	유성 순치 마찰음	vest	f
w	w	양순 연구개 접근음	west	u
z	z	유성 치경 마찰음	zero	s
ʒ	Z	유성 후치경 마찰음	vision	S
모음				

IPA	X-SAMPA	설명	예	Viseme
ə	@	중설 중모음	arena	@
əʊ	@U	이중 모음	goat	@
æ	{	전설 비원순 근저모음	trap	a
aɪ	aI	이중 모음	price	a
aʊ	aU	이중 모음	mouth	a
ɑː	A:	후설 비원순 저장모음	father	a
eɪ	eI	이중 모음	face	e
ɜː	3:	중설 비원순 중저장모음	nurse	E
ɛ	E	전설 비원순 중저모음	dress	E
ɛə	E@	이중 모음	square	E
iː	i	전설 비원순 고장모음	fleece	i
ɪ	I	근전설 비원순 근고모음	kit	i
ɪə	I@	이중 모음	near	i
ɔː	OI	후설 원순 중저장모음	thought	O
ɔɪ	OI	이중 모음	choice	O
ɒ	Q	후설 원순 저모음	lot	O

IPA	X-SAMPA	설명	예	Viseme
uː	u:	후설 원순 고장모음	goose	u
ʊ	U	근후설 원순 근고모음	foot	u
ʊə	U@	이중 모음	cure	u
ʌ	V	O pen-mid-back 언라운드된 모음	strut	E
추가 기호				
ˈ	"	제1강세	Alabama	
ˌ	%	제2강세	Alabama	
.	.	음절 경계	A.la.ba.ma	

영어(영국)(en-GB)

다음 표에는 Amazon Polly에서 지원하는 영국 영어 음성에 대해 IPA(International Phonetic Alphabet) 음소와 X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet) 기호 및 해당하는 viseme이 나와 있습니다.

음소/Viseme 표

IPA	X-SAMPA	설명	예	Viseme
자음				
b	b	유성 양순 파열음	bed	p
d	d	유성 치경 파열음	dig	t
ʤ	dZ	유성 후치경 파찰음	jump	S
ð	D	유성 치 마찰음	then	T

IPA	X-SAMPA	설명	예	Viseme
f	f	무성 순치 마찰음	five	f
g	g	유성 연구개 파열음	game	k
h	h	무성 성문 마찰음	house	k
j	j	경구개 접근음	yes	i
k	k	무성 연구개 파열음	cat	k
l	l	치경 설측 접근음	lay	t
l̩	l=	성절 치경 설측 접근음	battle	t
m	m	양순 비음	mouse	p
m̩	m=	성절 양순 비음	anthem	p
n	n	치경 비음	nap	t
n̩	n=	성절 치경 비음	button	t
ŋ	N	연구개 비음	thing	k
p	p	무성 양순 파열음	pin	p
ɹ	r\	치경 접근음	red	r
s	s	무성 치경 마찰음	seem	s
ʃ	S	무성 후치경 마찰음	ship	S
t	t	무성 치경 파열음	task	t
t͡ʃ	tS	무성 후치경 파찰음	chart	S
Θ	T	무성 치 마찰음	thin	T
v	v	유성 순치 마찰음	vest	f

IPA	X-SAMPA	설명	예	Viseme
w	w	양순 연구개 접근음	west	u
z	z	유성 치경 마찰음	zero	s
ʒ	Z	유성 후치경 마찰음	vision	S
모음				
ə	@	중설 중모음	arena	@
əʊ	@U	이중 모음	goat	@
æ	{	전설 비원순 근저모음	trap	a
aɪ	aI	이중 모음	price	a
aʊ	aU	이중 모음	mouth	a
ɑː	A:	후설 비원순 저장모음	father	a
eɪ	eI	이중 모음	face	e
ɜː	3:	중설 비원순 중저장모음	nurse	E
ɛ	E	전설 비원순 중저모음	dress	E
ɛə	E@	이중 모음	square	E
iː	i	전설 비원순 고장모음	fleece	i
ɪ	I	근전설 비원순 근고모음	kit	i
ɪə	I@	이중 모음	near	i

IPA	X-SAMPA	설명	예	Viseme
ɔː	O:	후설 원순 중저장모음	thought	O
ɔɪ	OI	이중 모음	choice	O
ɒ	Q	후설 원순 저모음	lot	O
uː	u:	후설 원순 고장모음	goose	u
ʊ	U	근후설 원순 근고모음	foot	u
ʊə	U@	이중 모음	cure	u
ʌ	V	O pen-mid-back 언라운드된 모음	strut	E
추가 기호				
ˈ	"	제1강세	Alabama	
ˌ	%	제2강세	Alabama	
.	.	음절 경계	A.la.ba.ma	

영어(인도)(en-IN)

다음 표에는 Amazon Polly에서 지원하는 인도 영어 음성에 대해 IPA(International Phonetic Alphabet) 음소와 X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet) 기호 및 해당하는 viseme이 나와 있습니다.

인도 영어와 함께 사용하는 그 밖의 음소는 힌디어(hi-IN)를 참조하십시오.

음소/Viseme 표

IPA	X-SAMPA	설명	예	Viseme
자음				

IPA	X-SAMPA	설명	예	Viseme
b	b	유성 양순 파열음	bed	p
d	d	유성 치경 파열음	dig	t
ʤ	dZ	유성 후치경 파찰음	jump	S
ð	D	유성 치 마찰음	then	T
f	f	무성 순치 마찰음	five	f
g	g	유성 연구개 파열음	game	k
h	h	무성 성문 마찰음	house	k
j	j	경구개 접근음	yes	i
k	k	무성 연구개 파열음	cat	k
l	l	치경 설측 접근음	lay	t
l̩	l=	성절 치경 설측 접근음	battle	t
m	m	양순 비음	mouse	p
m̩	m=	성절 양순 비음	anthem	p
n	n	치경 비음	nap	t
n̩	n=	성절 치경 비음	nap	t
ŋ	N	연구개 비음	thing	k
p	p	무성 양순 파열음	pin	p
ɹ	r\	치경 접근음	red	r
s	s	무성 치경 마찰음	seem	s
ʃ	S	무성 후치경 마찰음	ship	S

IPA	X-SAMPA	설명	예	Viseme
t	t	무성 치경 파열음	task	t
͡tʃ	tS	무성 후치경 파찰음	chart	S
Θ	T	무성 치 마찰음	thin	T
v	v	유성 순치 마찰음	vest	f
w	w	양순 연구개 접근음	west	u
z	z	유성 치경 마찰음	zero	s
ʒ	Z	유성 후치경 마찰음	vision	S
모음				
ə	@	중설 중모음	arena	@
əʊ	@U	이중 모음	goat	@
æ	{	전설 비원순 근저모음	trap	a
aɪ	aI	이중 모음	price	a
aʊ	aU	이중 모음	mouth	a
ɑː	A:	후설 비원순 저장모음	father	a
eɪ	eI	이중 모음	face	e
ɜː	3:	중설 비원순 중저장모음	nurse	E
ɛ	E	전설 비원순 중저모음	dress	E
ɛə	E@	이중 모음	square	E

IPA	X-SAMPA	설명	예	Viseme
iː	i	전설 비원순 고장모음	fleece	i
ɪ	I	근전설 비원순 근고모음	kit	i
ɪə	I@	이중 모음	near	i
ɔː	OI	후설 원순 중저장모음	thought	O
ɔɪ	OI	이중 모음	choice	O
ɒ	Q	후설 원순 저모음	lot	O
uː	uː	후설 원순 고장모음	goose	u
ʊ	U	근후설 원순 근고모음	foot	u
ʊə	U@	이중 모음	cure	u
ʌ	V	O pen-mid-back 언라운드된 모음	strut	E
추가 기호				
ˈ	"	제1강세	Alabama	
ˌ	%	제2강세	Alabama	
.	.	음절 경계	A.la.ba.ma	

영어(아일랜드)(en-IE)

다음 표에는 Amazon Polly에서 지원하는 아일랜드 영어 음성에 대해 IPA(International Phonetic Alphabet) 음소와 X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet) 기호 및 해당하는 viseme이 나와 있습니다.

음소/Viseme 표

IPA	X-SAMPA	설명	예	Viseme
자음				
b	b	유성 양순 파열음	bed	p
d	d	유성 치경 파열음	dig	t
ʤ	dZ	유성 후치경 파찰음	jump	S
ð	D	유성 치 마찰음	then	T
f	f	무성 순치 마찰음	five	f
g	g	유성 연구개 파열음	game	k
h	h	무성 성문 마찰음	house	k
j	j	경구개 접근음	yes	i
k	k	무성 연구개 파열음	cat	k
l	l	치경 설측 접근음	lay	t
m	m	양순 비음	mouse	p
n	n	치경 비음	nap	t
ŋ	N	연구개 비음	thing	k
p	p	무성 양순 파열음	speak	p
ɹ	r\	치경 접근음	red	r
s	s	무성 치경 마찰음	seem	s
ʃ	S	무성 후치경 마찰음	ship	S
t	t	무성 치경 파열음	trap	t
ʧ	tS	무성 후치경 파찰음	chart	S

IPA	X-SAMPA	설명	예	Viseme
θ	T	무성 치 마찰음	thin	T
v	v	유성 순치 마찰음	vest	f
w	w	양순 연구개 접근음	west	u
z	z	유성 치경 마찰음	zero	s
ʒ	Z	유성 후치경 마찰음	vision	S
모음				
ə	@	중설 중모음	arena	@
ɚ	@`	중설 치경 접근 수반음	reader	@
æ	{	전설 비원순 근저모음	trap	a
aɪ	aI	이중 모음	price	a
aʊ	aU	이중 모음	mouth	a
ɑ	A	후설 비원순 저장모음	father	a
eɪ	eI	이중 모음	face	e
ɝ	3`	중설 비원순 중저 치경 접근 수반음	nurse	E
ɛ	E	전설 비원순 중저모음	dress	E
i	i	전설 비원순 고장모음	fleece	i

IPA	X-SAMPA	설명	예	Viseme
ɪ	I	근전설 비원순 근고모음	kit	i
oʊ	oU	이중 모음	goat	o
ɔ	O	후설 원순 중저장모음	thought	O
ɔɪ	OI	이중 모음	choice	O
u	u	후설 원순 고장모음	goose	u
ʊ	U	근후설 원순 근고모음	foot	u
ʌ	V	open-mid-back 언라운드된 모음	strut	E
추가 기호				
ˈ	"	제1강세	Alabama	
ˌ	%	제2강세	Alabama	
.	.	음절 경계	A.la.ba.ma	

영어(뉴질랜드)(en-NZ)

다음 표에는 Amazon Polly에서 지원하는 뉴질랜드 영어 음성에 대해 IPA(International Phonetic Alphabet) 음소와 X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet) 기호 및 해당하는 viseme이 나와 있습니다.

음소/Viseme 표

IPA	X-SAMPA	설명	예	Viseme
자음				

IPA	X-SAMPA	설명	예	Viseme
b	b	유성 양순 파열음	bed	p
d	d	유성 치경 파열음	dig	t
͡dʒ	dZ	유성 후치경 파찰음	jump	S
ð	D	유성 치 마찰음	then	T
f	f	무성 순치 마찰음	five	f
g	g	유성 연구개 파열음	game	k
h	h	무성 성문 마찰음	house	k
j	j	경구개 접근음	yes	i
k	k	무성 연구개 파열음	cat	k
l	l	치경 설측 접근음	lay	t
l̩	l=	성절 치경 설측 접근음	battle	t
m	m	양순 비음	mouse	p
m̩	m=	성절 양순 비음	anthem	p
n	n	치경 비음	nap	t
n̩	n=	성절 치경 비음	button	t
ŋ	N	연구개 비음	thing	k
p	p	무성 양순 파열음	pin	p
ɹ	r\	치경 접근음	red	r
s	s	무성 치경 마찰음	seem	s
ʃ	S	무성 후치경 마찰음	ship	S

IPA	X-SAMPA	설명	예	Viseme
t	t	무성 치경 파열음	task	t
͡tʃ	tS	무성 후치경 파찰음	chart	S
Θ	T	무성 치 마찰음	thin	T
v	v	유성 순치 마찰음	vest	f
w	w	양순 연구개 접근음	west	u
z	z	유성 치경 마찰음	zero	s
ʒ	Z	유성 후치경 마찰음	vision	S
모음				
ə	@	중설 중모음	arena	@
əʊ	@U	이중 모음	goat	@
æ	{	전설 비원순 근저모음	trap	a
aɪ	aI	이중 모음	price	a
aʊ	aU	이중 모음	mouth	a
ɑː	A:	후설 비원순 저장모음	father	a
eɪ	eI	이중 모음	face	e
ɜː	3:	중설 비원순 중저장모음	nurse	E
ɛ	E	전설 비원순 중저모음	dress	E
ɛə	E@	이중 모음	square	E

IPA	X-SAMPA	설명	예	Viseme
iː	i	전설 비원순 고장모음	fleece	i
ɪ	I	근전설 비원순 근고모음	kit	i
ɪə	I@	이중 모음	near	i
ɔː	O:	후설 원순 중저장모음	thought	O
ɔɪ	OI	이중 모음	choice	O
ɒ	Q	후설 원순 저모음	lot	O
uː	u:	후설 원순 고장모음	goose	u
ʊ	U	근후설 원순 근고모음	foot	u
ʊə	U@	이중 모음	cure	u
ʌ	V	O pen-mid-back 언라운드된 모음	strut	E
추가 기호				
ˈ	"	제1강세	Alabama	
ˌ	%	제2강세	Alabama	
.	.	음절 경계	A.la.ba.ma	

Aria 음성은 뉴질랜드 영어를 구사하며 마오리어를 제한적으로 지원합니다. Aria 음성은 다음 마오리 단어와 구문을 발음할 수 있습니다. 마오리 문구는 대/소문자를 구분합니다.

영어	마오리
인사/환호	Kia ora
환영	Nau mai (ki)
인사(한 명)/감사	Tēnā koe
인사(세 명 이상)/감사	Tēnā koutou
아침 인사	Ata mārie
아침 인사	Mōrena
감사	Ngā mihi
작별 인사	Ngā manaakitanga
작별 인사 2	Ka kite
나중에 봐요	Mā te wā
좋은 하루 보내세요	Kia pai tō rā
메리 크리스마스	Meri Kirihimete
마오리	마오리
마오리어	te reo Māori
마오리어 주간	Te wiki o te reo Māori
뉴질랜드	Aotearoa
마오리 새해	Mātariki
뉴질랜드 마을, 와이탕이 데이는 뉴질랜드 국경일	Waitangi
1개	tahi
2	rua

영어	마오리
3	toru
4	whā
5	rima
6	ono
7	whitu
8	waru
9	iwa
10	tekau
20	rua tekau
30	Toru tekau

영어(남아프리카)(en-ZA)

다음 표에는 Amazon Polly에서 지원하는 남아프리카 영어 음성에 대해 IPA(International Phonetic Alphabet) 음소와 X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet) 기호 및 해당하는 viseme이 나와 있습니다.

음소/Viseme 표

IPA	X-SAMPA	설명	예	Viseme
자음				
b	b	유성 양순 파열음	bed	p
d	d	유성 치경 파열음	dig	t
ʤ	dZ	유성 후치경 파찰음	jump	S
ð	D	유성 치 마찰음	then	T

IPA	X-SAMPA	설명	예	Viseme
f	f	무성 순치 마찰음	five	f
g	g	유성 연구개 파열음	game	k
h	h	무성 성문 마찰음	house	k
j	j	경구개 접근음	yes	i
k	k	무성 연구개 파열음	cat	k
l	l	치경 설측 접근음	lay	t
l̩	l=	성절 치경 설측 접근음	battle	t
ɬ̩	K	무성 설측 마찰음	umhlanga	t
m	m	양순 비음	mouse	p
m̩	m=	성절 양순 비음	anthem	p
n	n	치경 비음	nap	t
n̩	n=	성절 치경 비음	button	t
ŋ	N	연구개 비음	thing	k
p	p	무성 양순 파열음	pin	p
ɹ	r\	치경 접근음	red	r
r	r	치경 전동음	pareis	r
s	s	무성 치경 마찰음	seem	s
ʃ	S	무성 후치경 마찰음	ship	S
t	t	무성 치경 파열음	task	t
t͡ʃ	tS	무성 후치경 파찰음	chart	S

IPA	X-SAMPA	설명	예	Viseme
θ	T	무성 치 마찰음	thin	T
v	v	유성 순치 마찰음	vest	f
w	w	양순 연구개 접근음	west	u
x	x	무성 연구개 마찰음	gauteng	k
z	z	유성 치경 마찰음	zero	s
!	!\	후치경 클릭음	gqeberha	k
|	|\	치 클릭음	ncube	t
||	||\	설측 클릭음	xhosa	t
모음				
ə	@	중설 중모음	arena	@
əi	@i	이중 모음	넬스퍼 ui t	i
əʊ	@U	이중 모음	goat	@
æ	{	전설 비원순 근저모음	trap	a
aɪ	aI	이중 모음	price	a
aʊ	aU	이중 모음	mouth	a
ɑː	A:	후설 비원순 저장모음	father	a
eɪ	eI	이중 모음	face	e
ɜː	3:	중설 비원순 중저장모음	nurse	E

IPA	X-SAMPA	설명	예	Viseme
ɛ	E	전설 비원순 중저모음	dress	E
ɛə	E@	이중 모음	square	E
iː	i	전설 비원순 고장모음	fleece	i
iə	I@	이중 모음	du preez	i
ɪ	I	근전설 비원순 근고모음	kit	i
ɪə	I@	이중 모음	near	i
ɔː	O:	후설 원순 중저장모음	thought	O
ɔɪ	OI	이중 모음	choice	O
ɒ	Q	후설 원순 저모음	lot	O
uː	u:	후설 원순 고장모음	goose	u
ʊ	U	근후설 원순 근고모음	foot	u
ʊə	U@	이중 모음	cure	u
ʌ	V	pen-mid-back 반올림되지 않은 모음	strut	E
y	y	전설 원순 고모음	van vuuren	u
추가 기호				
ˈ	"	제1강세	Alabama	
ˌ	%	제2강세	Alabama	

IPA	X-SAMPA	설명	예	Viseme
.	.	음절 경계	A.la.ba.ma	

영어(웨일스)(en-GB-WLS)

다음 표에는 Amazon Polly에서 지원하는 웨일스 영어 음성에 대해 IPA(International Phonetic Alphabet) 음소와 X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet) 기호 및 해당하는 viseme이 나와 있습니다.

음소/Viseme 표

IPA	X-SAMPA	설명	예	Viseme
자음				
b	b	유성 양순 파열음	bed	p
d	d	유성 치경 파열음	dig	t
ʤ	dZ	유성 후치경 파찰음	jump	S
ð	D	유성 치 마찰음	then	T
f	f	무성 순치 마찰음	five	f
g	g	유성 연구개 파열음	game	k
h	h	무성 성문 마찰음	house	k
j	j	경구개 접근음	yes	i
k	k	무성 연구개 파열음	cat	k
l	l	치경 설측 접근음	lay	t
l̩	l=	성절 치경 설측 접근음	battle	t
m	m	양순 비음	mouse	p

IPA	X-SAMPA	설명	예	Viseme
m̩	m=	성절 양순 비음	anthem	p
n	n	치경 비음	nap	t
n̩	n=	성절 치경 비음	nap	t
ŋ	N	연구개 비음	thing	k
p	p	무성 양순 파열음	pin	p
ɹ	r\	치경 접근음	red	r
s	s	무성 치경 마찰음	seem	s
ʃ	S	무성 후치경 마찰음	ship	S
t	t	무성 치경 파열음	task	t
͡tʃ	tS	무성 후치경 파찰음	chart	S
Θ	T	무성 치 마찰음	thin	T
v	v	유성 순치 마찰음	vest	f
w	w	양순 연구개 접근음	west	u
z	z	유성 치경 마찰음	zero	s
ʒ	Z	유성 후치경 마찰음	vision	S
모음				
ə	@	중설 중모음	arena	@
əʊ	@U	이중 모음	goat	@
æ	{	전설 비원순 근저모음	trap	a
aɪ	aI	이중 모음	price	a

IPA	X-SAMPA	설명	예	Viseme
aʊ	aU	이중 모음	mouth	a
ɑː	A:	후설 비원순 저장모음	father	a
eɪ	eI	이중 모음	face	e
ɜː	3:	중설 비원순 중저장모음	nurse	E
ɛ	E	전설 비원순 중저모음	dress	E
ɛə	E@	이중 모음	square	E
iː	i	전설 비원순 고장모음	fleece	i
ɪ	I	근전설 비원순 근고모음	kit	i
ɪə	I@	이중 모음	near	i
ɔː	OI	후설 원순 중저장모음	thought	O
ɔɪ	OI	이중 모음	choice	O
ɒ	Q	후설 원순 저모음	lot	O
uː	u:	후설 원순 고장모음	goose	u
ʊ	U	근후설 원순 근고모음	foot	u
ʊə	U@	이중 모음	cure	u
ʌ	V	O pen-mid-back 언라운드된 모음	strut	E

IPA	X-SAMPA	설명	예	Viseme
추가 기호				
ˈ	"	제1강세	Alabama	
ˌ	%	제2강세	Alabama	
.	.	음절 경계	A.la.ba.ma	

핀란드어(fi-FI)

다음 표에는 Amazon Polly에서 지원하는 핀란드어 음성에 대해 IPA(International Phonetic Alphabet) 음소와 X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet) 기호 및 해당하는 viseme이 나와 있습니다.

음소/Viseme 표

IPA	X-SAMPA	설명	예	Viseme
핀란드 자음				
p	p	무성 양순 파열음	[p]ankki	p
t	t	무성 치경 파열음	[t]alo	t
k	k	무성 연구개 파열음	[k]aali	k
d	d	유성 치경 파열음	[d]ata	t
s	s	무성 치경 마찰음	[s]ali	s
h	h	무성 성문 마찰음	[h]attu	k
ʋ	v\	유성 순치 접근음	[v]aiv́a	v
j	j	경구개 접근음	[j]oki	i
l	l	치경 설측 접근음	[l]oma	t

IPA	X-SAMPA	설명	예	Viseme
r	r	유성 치경 전동음	[r]iita	r
m	m	양순 비음	[m]ato	p
n	n	치경 비음	[n]enäa	t
ŋ	N	연구개 비음	he[n]ki	k
외래 단어에서 발견되는 자음				
b	b	유성 양순 파열음	[b]ussi	p
f	f	무성 순치 마찰음	[f]irma	v
w	w	양순 연구개 접근음	[w]iki	u
z	z	유성 치경 마찰음	[z]ulu	s
g	g	유성 연구개 파열음	[g]aala	k
ʃ	S	무성 후치경 마찰음	[sh]akki	S
ʒ	Z	유성 후치경 마찰음	[g]enre	S
θ	T	무성 치 마찰음	ear[th]	T
ð	D	유성 치 마찰음	ei[th]er	T
단모음				
i	i	전설 비원순 고모음	k[i]lo	i
ɛ	E	전설 비원순 중저모음	k[e]sä	E
æ	{	전설 비원순 근저모음	k[ä]ly	A
y	y	전설 원순 고모음	k[y]lä	u

IPA	X-SAMPA	설명	예	Viseme
ø	2	전중설 원순 모음	p[ö]ly	O
u	u	후설 원순 고모음	k[u]lo	u
ɔ	O	후중설 원순 모음	k[o]lo	O
ɑ	A	후설 비원순 저모음	k[a]la	A
장모음				
iː	iː	전설 비원순 고장모음	s[ii]li	i
ɛː	E:	전중설 원순 장모음	[ee]tu	E
æː	{:	전경설 비원순 장모음	t[ää]llä	A
yː	yː	전설 비원순 고장모음	t[yy]li	u
øː	2:	전중설 원순 모음	t[öö]lö	O
uː	uː	후설 원순 고장모음	t[uu]li	u
ɔː	O:	후설 원순 중저장모음	r[oo]li	O
ɑː	A:	후설 비원순 저장모음	k[aa]su	A
이중모음				
ɛi	Ei	이중모음	l[ei]pä	E
æi	{i	이중모음	[äi]ti	A
ui	ui	이중모음	k[ui]n	u

IPA	X-SAMPA	설명	예	Viseme
ɑi	Ai	이중모음	k[ai]kki	A
ɔi	Oi	이중모음	p[oi]ka	O
øi	2i	이중모음	s[öi]n	O
yi	yi	이중모음	l[yi]jy	u
ɑu	Au	이중모음	s[au]na	A
ɔu	Ou	이중모음	k[ou]lu	O
ɛu	Eu	이중모음	r[eu]na	E
iu	iu	이중모음	v[iu]lu	i
æy	{y	이중모음	t[äy]nnä	A
øy	2y	이중모음	k[öy]hä	O
ɛy	Ey	이중모음	pes[ey]tyä	E
iy	iy	이중모음	käär[iy]tyä	i
iɛ	iE	이중모음	t[ie]	i
yø	y2	이중모음	[yö]	u
uɔ	uO	이중모음	t[uo]	u
영어 외래어에서 발견되는 모음				
ɪ	I	근전설 비원순 근고모음	b[i]t	i
ʊ	U	근후설 원순 근고모음	b[oo]k	u
ə	@	중설 중모음	[a]bout	@

IPA	X-SAMPA	설명	예	Viseme
ʌ	V	open-mid-back 언라운드된 모음	c[u]t	E

프랑스어(fr-FR)

다음 표에는 Amazon Polly에서 지원하는 프랑스어 음성에 대해 IPA(International Phonetic Alphabet) 음소와 X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet) 기호 및 해당하는 viseme이 나와 있습니다.

음소/Viseme 표

IPA	X-SAMPA	설명	예	Viseme
자음				
b	b	유성 양순 파열음	boire	p
d	d	유성 치경 파열음	madame	t
f	f	무성 순치 마찰음	femme	f
g	g	유성 연구개 파열음	grand	k
ɥ	H	양순 경구개 접근음	bruit	u
j	j	경구개 접근음	meilleur	i
k	k	무성 연구개 파열음	quatre	k
l	l	치경 설측 접근음	malade	t
m	m	양순 비음	maison	p
n	n	치경 비음	astronome	t
ɲ	J	경구개 비음	baigner	J
ŋ	N	연구개 비음	parking	k

IPA	X-SAMPA	설명	예	Viseme
p	p	무성 양순 파열음	pomme	p
ʁ	R	유성 구개수 마찰음	amoureux	k
s	s	무성 치경 마찰음	santé	s
ʃ	S	무성 후치경 마찰음	chat	S
t	t	무성 치경 파열음	téléphone	t
v	v	유성 순치 마찰음	vrai	f
w	w	양순 연구개 접근음	soir	u
z	z	유성 치경 마찰음	raison	s
ʒ	Z	유성 후치경 마찰음	aubergine	S
모음				
ø	2	전설 원순 중고모음	deux	o
œ	9	전설 원순 중저모음	neuf	O
œ̃	9~	전설 원순 비(nasal) 중저모음	brun	O
ə	@	중설 중모음	je	@
a	a	전설 비원순 저모음	table	a
ɑ̃	A~	후설 비원순 비(nasal) 저모음	camembert	a
e	e	전설 비원순 중고모음	marché	e
ɛ	E	전설 비원순 중저모음	neige	E

IPA	X-SAMPA	설명	예	Viseme
ɛ̃	E~	전설 비원순 비(nasal) 중저모음	sapin	E
i	i	전설 비원순 고모음	mille	i
o	o	후설 원순 중고모음	hôpital	o
ɔ	O	후설 원순 중저모음	homme	O
ɔ̃	O~	후설 원순 비(nasal) 중저모음	bon	O
u	u	후설 원순 고모음	sous	u
y	y	전설 원순 고모음	dur	u
추가 기호				
ˈ	"	제1강세	Alabama	
ˌ	%	제2강세	Alabama	
.	.	음절 경계	A.la.ba.ma	

프랑스어(벨기에)(fr-BE)

다음 표에는 Amazon Polly에서 지원하는 벨기에 프랑스어 음성에 대해 IPA(International Phonetic Alphabet) 음소와 X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet) 기호 및 해당하는 viseme이 나와 있습니다.

음소/Viseme 표

IPA	X-SAMPA	설명	예	Viseme
자음				
b	b	유성 양순 파열음	boire	p

IPA	X-SAMPA	설명	예	Viseme
d	d	유성 치경 파열음	madame	t
f	f	무성 순치 마찰음	femme	f
g	g	유성 연구개 파열음	grand	k
ɥ	H	양순 경구개 접근음	bruit	u
j	j	경구개 접근음	meilleur	i
k	k	무성 연구개 파열음	quatre	k
l	l	치경 설측 접근음	malade	t
m	m	양순 비음	maison	p
n	n	치경 비음	astronome	t
ɲ	J	경구개 비음	baigner	J
ŋ	N	연구개 비음	parking	k
p	p	무성 양순 파열음	pomme	p
ʁ	R	유성 구개수 마찰음	amoureux	k
s	s	무성 치경 마찰음	santé	s
ʃ	S	무성 후치경 마찰음	chat	S
t	t	무성 치경 파열음	téléphone	t
v	v	유성 순치 마찰음	vrai	f
w	w	양순 연구개 접근음	soir	u
z	z	유성 치경 마찰음	raison	s
ʒ	Z	유성 후치경 마찰음	aubergine	S

IPA	X-SAMPA	설명	예	Viseme
모음				
ø	2	전설 원순 중고모음	deux	o
œ	9	전설 원순 중저모음	neuf	O
œ̃	9~	전설 원순 비(nasal) 중저모음	brun	O
ə	@	중설 중모음	je	@
a	a	전설 비원순 저모음	table	a
ɑ̃	A~	후설 비원순 비(nasal) 저모음	camembert	a
e	e	전설 비원순 중고모음	marché	e
ɛ	E	전설 비원순 중저모음	neige	E
ɛ̃	E~	전설 비원순 비(nasal) 중저모음	sapin	E
i	i	전설 비원순 고모음	mille	i
o	o	후설 원순 중고모음	hôpital	o
ɔ	O	후설 원순 중저모음	homme	O
ɔ̃	O~	후설 원순 비(nasal) 중저모음	bon	O
u	u	후설 원순 고모음	sous	u
y	y	전설 원순 고모음	dur	u
추가 기호				

IPA	X-SAMPA	설명	예	Viseme
ˈ	"	제1강세	Alabama	
ˌ	%	제2강세	Alabama	
.	.	음절 경계	A.la.ba.ma	

프랑스어(캐나다)(fr-CA)

다음 표에는 Amazon Polly에서 지원하는 캐나다 프랑스어 음성에 대해 IPA(International Phonetic Alphabet) 음소와 X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet) 기호 및 해당하는 viseme이 나와 있습니다.

음소/Viseme 표

IPA	X-SAMPA	설명	예	Viseme
자음				
b	b	유성 양순 파열음	boire	p
d	d	유성 치경 파열음	madame	t
f	f	무성 순치 마찰음	femme	f
g	g	유성 연구개 파열음	grand	k
ɥ	H	양순 경구개 접근음	bruit	u
j	j	경구개 접근음	meilleur	i
k	k	무성 연구개 파열음	quatre	k
l	l	치경 설측 접근음	malade	t
m	m	양순 비음	maison	p
n	n	치경 비음	astronome	t
ɲ	J	경구개 비음	baigner	J

IPA	X-SAMPA	설명	예	Viseme
ŋ	N	연구개 비음	parking	k
p	p	무성 양순 파열음	pomme	p
ʁ	R	유성 구개수 마찰음	amoureux	k
s	s	무성 치경 마찰음	santé	s
ʃ	S	무성 후치경 마찰음	chat	S
t	t	무성 치경 파열음	téléphone	t
v	v	유성 순치 마찰음	vrai	f
w	w	양순 연구개 접근음	soir	u
z	z	유성 치경 마찰음	raison	s
ʒ	Z	유성 후치경 마찰음	aubergine	S
모음				
ø	2	전설 원순 중고모음	deux	o
œ	9	전설 원순 중저모음	neuf	O
œ̃	9~	전설 원순 비(nasal) 중저모음	brun	O
ə	@	중설 중모음	je	@
a	a	전설 비원순 저모음	table	a
ɑ̃	A~	후설 비원순 비(nasal) 저모음	camembert	a
e	e	전설 비원순 중고모음	marché	e

IPA	X-SAMPA	설명	예	Viseme
ɛ	E	전설 비원순 중저모음	neige	E
ɛ̃	E~	전설 비원순 비(nasal) 중저모음	sapin	E
i	i	전설 비원순 고모음	mille	i
o	o	후설 원순 중고모음	hôpital	o
ɔ	O	후설 원순 중저모음	homme	O
ɔ̃	O~	후설 원순 비(nasal) 중저모음	bon	O
u	u	후설 원순 고모음	sous	u
y	y	전설 원순 고모음	dur	u
추가 기호				
ˈ	"	제1강세	Alabama	
ˌ	%	제2강세	Alabama	
.	.	음절 경계	A.la.ba.ma	

독일어(de-DE)

다음 표에는 Amazon Polly에서 지원하는 독일어 음성에 대해 IPA(International Phonetic Alphabet) 음소와 X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet) 기호 및 해당하는 viseme이 나와 있습니다.

음소/Viseme 표

IPA	X-SAMPA	설명	예	Viseme
자음				

IPA	X-SAMPA	설명	예	Viseme
ʔ	?	성문 폐쇄음		
b	b	유성 양순 파열음	Bier	p
d	d	유성 치경 파열음	Dach	t
ç	C	무성 경구개 마찰음	ich	k
͡dʒ	dZ	유성 후치경 파찰음	Dschungel	S
f	f	무성 순치 마찰음	Vogel	f
g	g	유성 연구개 파열음	Gabel	k
h	h	무성 성문 마찰음	Haus	k
j	j	무성 성문 마찰음	jemand	i
k	k	무성 연구개 파열음	Kleid	k
l	l	치경 설측 접근음	Loch	t
m	m	양순 비음	Milch	p
n	n	치경 비음	Natur	t
ŋ	N	연구개 비음	klingen	k
p	p	무성 양순 파열음	Park	p
͡pf	pf	무성 순치 파찰음	Apfel	
ʁ	R	구개수 전동음	Regen	
s	s	무성 치경 마찰음	Messer	s
ʃ	S	무성 후치경 마찰음	Fischer	S
t	t	무성 치경 파열음	Topf	T

IPA	X-SAMPA	설명	예	Viseme
͡ts	Ts	무성 치경 파찰음	Zahl	
͡tʃ	tS	무성 후치경 파찰음	deutsch	S
v	v	유성 순치 마찰음	Wasser	f
x	x	무성 연구개 마찰음	kochen	k
z	z	유성 치경 마찰음	See	s
ʒ	Z	유성 후치경 마찰음	Orange	S
모음				
øː	2:	전설 원순 중고장모음	böse	o
ɐ	6	중설 근저모음	besser	a
ɐ̯	6_^	비성절 중설 근저모음	Klar	a
œ	9	전설 원순 중저모음	können	O
ə	@	중설 중모음	Rede	@
a	a	전설 비원순 저모음	Salz	a
aː	a:	전설 비원순 저장모음	Sahne	a
aɪ	aI	이중 모음	nein	a
aʊ	aU	이중 모음	Augen	a
ɑ̃	A~	후설 비원순 비(nasal) 저모음	Restaurant	a

IPA	X-SAMPA	설명	예	Viseme
eː	e:	전설 비원순 중고장모음	Rede	e
ɛ	E	전설 비원순 중저모음	Keller	E
ɛ̃	E~	전설 비원순 비(nasal) 중저모음	Terrain	E
iː	i:	전설 비원순 고장모음	Lied	i
ɪ	I	근전설 비원순 근고모음	bitte	i
oː	o:	후설 원순 중고장모음	Kohl	o
ɔ	O	후설 원순 중저모음	Koffer	O
ɔ̃	O~	후설 원순 비(nasal) 중저모음	Annonce	O
ɔʏ	OY	이중 모음	neu	O
uː	u:	후설 원순 고장모음	Bruder	u
ʊ	U	근후설 원순 근고모음	Wunder	u
yː	y:	전설 원순 고장모음	kühl	u
ʏ	Y	근전설 원순 근고모음	Küche	u
추가 기호				
ˈ	"	제1강세	Alabama	

IPA	X-SAMPA	설명	예	Viseme
ˌ	%	제2강세	Alabama	
.	.	음절 경계	A.la.ba.ma	

독일어(오스트리아)(de-AT)

다음 표에는 Amazon Polly에서 지원하는 오스트리아 독일어 음성에 대해 IPA(International Phonetic Alphabet) 음소와 X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet) 기호 및 해당하는 viseme이 나와 있습니다.

음소/Viseme 표

IPA	X-SAMPA	설명	예	Viseme
자음				
ʔ	?	성문 폐쇄음		
b	b	유성 양순 파열음	Bier	p
d	d	유성 치경 파열음	Dach	t
ç	C	무성 경구개 마찰음	ich	k
͡dʒ	dZ	유성 후치경 파찰음	Dschungel	S
f	f	무성 순치 마찰음	Vogel	f
g	g	유성 연구개 파열음	Gabel	k
h	h	무성 성문 마찰음	Haus	k
j	j	무성 성문 마찰음	jemand	i
k	k	무성 연구개 파열음	Kleid	k
l	l	치경 설측 접근음	Loch	t
m	m	양순 비음	Milch	p

IPA	X-SAMPA	설명	예	Viseme
n	n	치경 비음	Natur	t
ŋ	N	연구개 비음	klingen	k
p	p	무성 양순 파열음	Park	p
͡pf	pf	무성 순치 파찰음	Apfel	
ʁ	R	구개수 전동음	Regen	
s	s	무성 치경 마찰음	Messer	s
ʃ	S	무성 후치경 마찰음	Fischer	S
t	t	무성 치경 파열음	Topf	T
͡ts	Ts	무성 치경 파찰음	Zahl	
͡tʃ	tS	무성 후치경 파찰음	deutsch	S
v	v	유성 순치 마찰음	Wasser	f
x	x	무성 연구개 마찰음	kochen	k
z	z	유성 치경 마찰음	See	s
ʒ	Z	유성 후치경 마찰음	Orange	S
모음				
øː	2:	전설 원순 중고장모음	böse	o
ɐ	6	중설 근저모음	besser	a
ɐ̯	6_^	비성절 중설 근저모음	Klar	a
œ	9	전설 원순 중저모음	können	O

IPA	X-SAMPA	설명	예	Viseme
ə	@	중설 중모음	Rede	@
a	a	전설 비원순 저모음	Salz	a
a:	a:	전설 비원순 저장모음	Sahne	a
aɪ	aI	이중 모음	nein	a
aʊ	aU	이중 모음	Augen	a
ɑ̃	A~	후설 비원순 비(nasal) 저모음	Restaurant	a
e:	e:	전설 비원순 중고장모음	Rede	e
ɛ	E	전설 비원순 중저모음	Keller	E
ɛ̃	E~	전설 비원순 비(nasal) 중저모음	Terrain	E
i:	i:	전설 비원순 고장모음	Lied	i
ɪ	I	근전설 비원순 근고모음	bitte	i
o:	o:	후설 원순 중고장모음	Kohl	o
ɔ	O	후설 원순 중저모음	Koffer	O
ɔ̃	O~	후설 원순 비(nasal) 중저모음	Annonce	O
ɔʏ	OY	이중 모음	neu	O

IPA	X-SAMPA	설명	예	Viseme
u:	u:	후설 원순 고장모음	Bruder	u
ʊ	U	근후설 원순 근고모음	Wunder	u
y:	y:	전설 원순 고장모음	kühl	u
ʏ	Y	근전설 원순 근고모음	Küche	u
추가 기호				
ˈ	"	제1강세	Alabama	
ˌ	%	제2강세	Alabama	
.	.	음절 경계	A.la.ba.ma	

힌디어(hi-IN)

다음 표에는 Amazon Polly에서 지원하는 힌디어 음성에 대해 IPA(International Phonetic Alphabet) 음소와 X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet) 기호 및 음소의 소리 유형이 나와 있습니다.

힌디어와 함께 사용하는 그 밖의 음소는 영어(인도)(en-IN)를 참조하십시오.

음소/Viseme 표

IPA	X-SAMPA	설명	예
자음			
pʰ	p_h	무성 기식 양순 파열음	फूल(phool)
bʱ	b_h	유성 기식 양순 파열음	भारी(bhaari)
t̪	t_d	무성 치 파열음	तापमान(taapmaan)
t̪ʰ	t_d_h	무성 기식 치 파열음	थोड़ा(thoda)

IPA	X-SAMPA	설명	예
d̪	d_d	유성 치 파열음	दिल्ली(dilli)
d̪ʰ	d_d_h	유성 기식 치 파열음	धोबी(dhobi)
ʈ	t`	무성 권설 파열음	कटोरा(katora)
ʈʰ	t`_h	무성 기식 권설 파열음	ठंड(thand)
ɖ	d`	유성 권설 파열음	डर(darr)
ɖʰ	d`_h	유성 기식 권설 파열음	ढाल(dhal)
tʃʰ	tS_h	무성 기식 구개 파찰음	छाल(chaal)
dʒʰ	dZ_h	유성 기식 구개 파찰음	झाल(jhaal)
kʰ	k_h	무성 기식 연구개 파열음	खान(khan)
gʱ	g_h	유성 기식 연구개 파열음	घान(ghaan)
ɳ	n`	권설 비음	क्षण(kshan)
ɾ	4	치경 탄음	राम(ram)
ɽ	r`	평권설 단전음	बड़ा(bada)
ɽʱ	r`_h	유성 기식 권설 단전음	बढ़ी(barhi)
ʋ	v\	양순 접근음	वसूल(wasool)
모음			
ə	@_o	중설 중모음	अच्छा(achhaa)
ə̃	@~	중설 비음화 중모음	हँसना(hansnaa)
a	A_o	전설 비원순 저모음	आग(aag)
ã	A~	전설 비원순 비음화 저모음	घ़डयिाँ (ghariyaan)

IPA	X-SAMPA	설명	예
ɪ	I_o	근전설 비원순 근고모음	इक्कीस(ikkees)
ɪ̃	I~	근전설 비원순 비음화 근고모음	संचिाई(sinchai)
i	i_o	전설 비원순 고모음	बल्लिी(billee)
ĩ	i~	전설 비원순 비음화 고모음	नहीं(nahin)
ʊ	U_o	근후설 원순 근고모음	उलूल(ullu)
ʊ̃	U~	근후설 원순 비음화 근고모음	मुँह(munh)
u	u_o	후설 원순 고모음	फूल(phool)
ũ	u~	후설 원순 비음화 고모음	ऊँट(oont)
ɔ	O_o	후설 원순 중저모음	कौन(kaun)
ɔ̃	O~	후설 원순 비음화 중저모음	भौं(bhaun)
o	o	후설 원순 중고모음	सोना(sona)
õ	o~	후설 원순 비음화 중고모음	क्यों(kyon)
ɛ	E_o	전설 비원순 중저모음	पैसा(paisa)
ɛ̃	E~	전설 비원순 비음화 중저모음	मैं(main)
e	e	전설 비원순 중고모음	एक(ek)
ẽ	e~	전설 비원순 비음화 중고모음	किताबें (kitabein)

아이슬란드어(is-IS)

다음 표에는 Amazon Polly에서 지원하는 아이슬란드어 음성에 대해 IPA(International Phonetic Alphabet) 음소와 X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet) 기호 및 해당하는 viseme이 나와 있습니다.

음소/Viseme 표

IPA	X-SAMPA	설명	예	Viseme
자음				
b	b	유성 양순 파열음	grasbakkanum	0
c	c	무성 경구개 파열음	pakkin	k
cʰ	c_h	기식 무성 경구개 파열음	anarkistai	k
ç	C	무성 경구개 마찰음	héðan	k
d	d	유성 치경 파열음	bóndi	t
ð	D	유성 치 마찰음	borð	T
f	f	무성 순치 마찰음	duft	f
g	g	유성 연구개 파열음	holgóma	k
ɣ	G	유성 연구개 마찰음	hugur	k
h	h	무성 성문 마찰음	heili	k
j	j	경구개 접근음	jökull	i
kʰ	k_h	기식 무성 연구개 파열음	ósköpunum	k
l	l	치경 설측 접근음	gólf	t
l̥	l_0	무성 치경 설측 접근음	fólk	t

IPA	X-SAMPA	설명	예	Viseme
m	m	양순 비음	september	p
m̥	m_0	무성 양순 비음	kompa	p
n	n	치경 비음	númer	t
n̥	n_0	무성 치경 비음	pöntun	t
ɲ	J	경구개 비음	pælingar	J
ŋ	N	연구개 비음	söngvarann	k
ŋ̊	N_0	무성 연구개 비음	frænka	k
pʰ	p_h	기식 무성 양순 파열음	afplánun	p
r	r	치경 전동음	afskrifta	r
r̥	r_0	무성 치경 전동음	andvörpum	r
s	s	무성 치경 마찰음	baðhús	s
tʰ	t_h	기식 무성 치경 파열음	tanki	t
θ	T	무성 치 마찰음	þeldökki	T
v	v	유성 순치 마찰음	silfur	f
w	w	양순 연구개 접근음		u
x	x	무성 연구개 마찰음	samfélags	k
모음				
œ	9	전설 원순 중저모음	þröskuldinum	O
œː	9:	전설 원순 중저장모음	tvö	O

IPA	X-SAMPA	설명	예	Viseme
a	a	전설 비원순 저모음	nefna	a
a:	a:	전설 비원순 저장모음	fara	a
au	au	이중 모음	átta	a
au:	au:	이중 모음	átján	a
ɛ	E	전설 비원순 중저모음	kennari	E
ɛ:	E:	전설 비원순 중저장모음	dreka	E
i	i	전설 비원순 고모음	Gúlíver	i
i:	i:	전설 비원순 고장모음	þrír	i
ɪ	I	근전설 비원순 근고모음	samspil	i
ɪ:	I:	근전설 비원순 근고장모음	stig	i
ɔ	O	후설 원순 중저모음	regndropar	O
ɔ:	O:	후설 원순 중저장모음	ullarbolur	O
ɔu	Ou	이중 모음	tólf	O
ɔu:	Ou:	이중 모음	fjórir	O
u	u	후설 원순 고모음	stúlkan	u
u:	u:	후설 원순 고장모음	frú	u

IPA	X-SAMPA	설명	예	Viseme
ʏ	Y	근전설 원순 근고모음	tíu	u
ʏː	Y	근전설 원순 근고장모음	gruninn	u
추가 기호				
ˈ	"	제1강세	Alabama	
ˌ	%	제2강세	Alabama	
.	.	음절 경계	A.la.ba.ma	

이탈리아어(it-IT)

다음 표에는 Amazon Polly에서 지원하는 이탈리아어 음성에 대해 IPA(International Phonetic Alphabet) 음소와 X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet) 기호 및 해당하는 viseme이 나와 있습니다.

음소/Viseme 표

IPA	X-SAMPA	설명	예	Viseme
자음				
b	b	유성 양순 파열음	bacca	p
d	d	유성 치경 파열음	dama	t
͡dz	dz	유성 치경 파찰음	zero	s
͡dʒ	dZ	유성 후치경 파찰음	giro	S
f	f	무성 순치 마찰음	famiglia	f
g	g	유성 연구개 파열음	gatto	k

IPA	X-SAMPA	설명	예	Viseme
h	h	무성 성문 마찰음	horror	k
j	j	경구개 접근음	dieci	i
k	k	무성 연구개 파열음	campo	k
l	l	치경 설측 접근음	lido	t
ʎ	L	경구개 설측 접근음	aglio	J
m	m	양순 비음	mille	p
n	n	치경 비음	nove	t
ɲ	J	경구개 비음	lasagne	J
p	p	무성 양순 파열음	pizza	p
r	r	치경 전동음	risata	r
s	s	무성 치경 마찰음	sei	s
ʃ	S	무성 후치경 마찰음	scienza	S
t	t	무성 치경 파열음	tavola	t
͡ts	ts	무성 치경 파찰음	forza	s
͡tʃ	tS	무성 후치경 파찰음	cielo	S
v	v	유성 순치 마찰음	venti	f
w	w	양순 연구개 접근음	quattro	u
z	z	유성 치경 마찰음	bisogno	s
ʒ	Z	유성 후치경 마찰음	bijou	S
모음				

IPA	X-SAMPA	설명	예	Viseme
a	a	전설 비원순 저모음	arco	a
e	e	전설 비원순 중고모음	tre	e
ɛ	E	전설 비원순 중저모음	ettaro	E
i	i	전설 비원순 고모음	impero	i
o	o	후설 원순 중고모음	cento	o
ɔ	O	후설 원순 중저모음	otto	O
u	u	후설 원순 고모음	uno	u
추가 기호				
ˈ	"	제1강세	Alabama	
ˌ	%	제2강세	Alabama	
.	.	음절 경계	A.la.ba.ma	

일본어(ja-JP)

Amazon Polly는 일본어에 대해 가나 및 요미가나 알파벳 발음을 지원합니다. Amazon Polly가 이러한 알파벳의 발음 발음을 사용하도록 하려면 `alphabet="x-amazon-`*`phonetic standard used`*`"` 음소 속성을 사용하세요.

- `x-amazon-pron-kana` – 가나 발음이 사용됨을 나타냅니다. 가나 발음은 음성 표기에 사용되는 특수 가타카나 문자이며 음높이 악센트를 인코딩할 수 있습니다.
- `x-amazon-yomigana` – 요미가나가 사용되었음을 나타냅니다. 요미가나는 햅번 로마자 표기법으로 해석되는 일반적인 가타카나, 히라가나, 라틴 알파벳일 수 있습니다.

다음 예제는 이들을 사용하는 방법을 보여줍니다.

발음: 가나

```
<speak>
     ###<phoneme alphabet="x-amazon-pron-kana" ph="###'#">##</phoneme>###
</speak>
```

요미가나

```
<speak>
     ###<phoneme alphabet="x-amazon-yomigana" ph="####">##</phoneme>###
     ###<phoneme alphabet="x-amazon-yomigana" ph="####">##</phoneme>###
     ###<phoneme alphabet="x-amazon-yomigana" ph="Hirokazu">##</phoneme>###
</speak>
```

다음 표에는 Amazon Polly에서 지원하는 일본어 음성에 대해 IPA(International Phonetic Alphabet) 음소와 X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet) 기호 및 해당하는 viseme이 나와 있습니다.

IPA	X-SAMPA	설명	예	Viseme
자음				
ɾ	4	치경 탄음	練習, renshuu	t
ʔ	?	성문 폐쇄음	あつっ, atsu'	
b	b	유성 양순 파열음	舞踊, buyou	p
β	B	유성 양순 마찰음	ヴィンテージ, vinteeji	B
c	c	무성 경구개 파열음	ききょう, kikyou	k
ç	C	무성 경구개 마찰음	人, hito	k
d	d	유성 치경 파열음	濁点, dakuten	t
͡dʑ	dz\	유성 치경구개 파찰음	純, jun	J
g	g	유성 연구개 파열음	ご飯, gohan	k

IPA	X-SAMPA	설명	예	Viseme
h	h	무성 성문 마찰음	本, hon	k
j	j	경구개 접근음	屋根, yane	i
ɟ	J\	유성 경구개 파열음	行儀, gyougi	J
k	k	무성 연구개 파열음	漢字, kanji	k
ɺ	l\	치경 설측 탄음	釣り, tsuri	r
ɺj	l\j	치경 설측 탄음, 경구개 접근음	流行, ryuukou	r
m	m	양순 비음	飯, meshi	p
n	n	치경 비음	猫, neko	t
ɲ	J	경구개 비음	日本, nippon	J
ɴ	N\	구개수 비음	缶, kan	k
p	p	무성 양순 파열음	パン, pan	p
ɸ	p\	무성 양순 마찰음	福, huku	f
s	s	무성 치경 마찰음	層, sou	s
ɕ	s\	무성 치경구개 마찰음	書簡, shokan	J
t	t	무성 치경 파열음	手紙, tegami	t
t͡s	ts	무성 치경 파찰음	釣り, tsuri	s
t͡ɕ	ts\	무성 치경구개 파찰음	吉, kichi	J
w	w	양순 연구개 접근음	電話, denwa	u
z	z	유성 치경 마찰음	座敷, zashiki	s

IPA	X-SAMPA	설명	예	Viseme
모음				
äː	a:_"	중설 비원순 저장모음	羽蟻, haari	a
ä	a_"	중설 비원순 저모음	仮名, kana	a
eː	e:_o	전설 비원순 중장모음	学生, gakusei	@
e	e_o	전설 비원순 중모음	歴, reki	@
i	i	전설 비원순 고모음	気, ki	i
iː	i:	전설 비원순 고장모음	詩歌, shiika	i
ɯ	M USD	후설 비원순 고모음	運, un	i
ɯː	M USD:	후설 비원순 고장모음	宗教, shuukyou	i
oː	o:_o	후설 원순 중장모음	購読, koodoku	o
o	o_o	후설 원순 중모음	読者, dokusha	o

한국어(ko-KR)

다음 표에는 Amazon Polly에서 지원하는 한국어 음성에 대해 IPA(International Phonetic Alphabet) 음소와 X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet) 기호 및 해당하는 viseme이 나와 있습니다.

IPA	X-SAMPA	설명	예	Viseme
자음				
k	k	무성 연구개 파열음	강, [g]ang	k

IPA	X-SAMPA	설명	예	Viseme
k#	k_t	무성 연구개 경파열음	깨, [kk]e	k
n	n	치경 비음	남, [n]am	t
t	t	무성 치경 파열음	도, [d]o	t
t#	t_t	무성 치경 경파열음	때, [tt]e	t
ɾ	4	치경 탄음	사랑, sa[r]ang	t
l	l	치경 설측 접근음	돌, do[l]	t
m	m	양순 비음	무, [m]u	p
p	p	무성 양순 파열음	봄, [b]om	p
p#	p_t	무성 양순 경파열음	뻘, [pp]eol	p
s	s	무성 치경 마찰음	새, [s]e	s
s#	s_t	무성 치경 경마찰음	씨, [ss]i	s
ŋ	N	연구개 비음	방, ba[ng]	k
t͡ɕ	ts\	무성 치경구개 파찰음	조, [j]o	J
t#͡ɕ	ts_t	무성 치경구개 경파찰음	찌, [jj]i	J
t͡ɕʰ	ts_h	기식 무성 치경구개 경파찰음	차, [ch]a	J
kʰ	k_h	기식 무성 연구개 파열음	코, [k]o	k
tʰ	t_h	기식 무성 치경 파열음	통, [t]ong	t

IPA	X-SAMPA	설명	예	Viseme
p^h	p_h	기식 무성 양순 파열음	패, [p]e	p
h	h	무성 성문 마찰음	힘, [h]im	k
j	j	경구개 접근음	양, [y]ang	i
w	w	양순 연구개 접근음	왕, [w]ang	u
ɯ̟	M\	연구개 접근음>	의, [wj]i	i
모음				
a	a	전설 비원순 저모음	밥, b[a]b	a
ʌ	V	후설 평순 중저모음	정, j[eo]ng	E
ɛ	E	전설 비원순 중저모음	배, b[e]	E
o	o	후설 원순 중고모음	노, n[o]	o
u	u	후설 원순 고모음	둘, d[u]l	u
ɯ	M	후설 비원순 고모음	은, [eu]n	i
i	i	전설 비원순 고모음	김, k[i]m	i

노르웨이어(nb-NO)

다음 차트에는 Amazon Polly에서 지원하는 노르웨이어 음성에 대해 IPA(International Phonetic Alphabet) 음소와 X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet) 기호 세트 및 해당하는 viseme이 나와 있습니다.

IPA	X-SAMPA	설명	예	Viseme
자음				

IPA	X-SAMPA	설명	예	Viseme
ɾ	4	치경 탄음	prøv	t
b	b	유성 양순 파열음	labb	p
ç	C	무성 경구개 마찰음	kino	k
d	d	유성 치경 파열음	ladd	t
ɖ	d`	유성 권설 파열음	verdi	t
f	f	무성 순치 마찰음	fot	f
g	g	유성 연구개 파열음	tagg	k
h	h	무성 성문 마찰음	ha	k
j	j	경구개 접근음	gi	i
k	k	무성 연구개 파열음	takk	k
l	l	치경 설측 접근음	fall, ball	t
ɭ	l`	권설 설측 접근음	ærlig	t
m	m	양순 비음	lam	p
n	n	치경 비음	vann	t
ɳ	n`	권설 비음	garn	t
ŋ	N	연구개 비음	sang	k
p	p	무성 양순 파열음	hopp	p
s	s	무성 치경 마찰음	lass	s
ʂ	s`	무성 권설 마찰음	års	S
ʃ	S	무성 후치경 마찰음	skyt	S

IPA	X-SAMPA	설명	예	Viseme
t	t	무성 치경 파열음	lat	t
ʈ	t`	무성 권설 파열음	hardt	t
ʋ	v\	순치 접근음	vin	f
w	w	양순 연구개 접근음	will	x
모음				
øː	2:	전설 원순 중고장모음	søt	o
œ	9	전설 원순 중저모음	søtt	O
ə	@	중설 중모음	ape	@
æː	{:	전설 비원순 근저장모음	vær	a
ʉ	}	중설 원순 고모음	lund	u
ʉː	}:	중설 원순 고장모음	lun	u
æ	{	전설 비원순 근저모음	vært	a
ɑ	A	후설 비원순 저모음	hatt	a
ɑː	A:	후설 비원순 저장모음	hat	a
eː	e:	전설 비원순 중고장모음	sen	e
ɛ	E	전설 비원순 중저모음	send	E

IPA	X-SAMPA	설명	예	Viseme
iː	i:	전설 비원순 고장모음	vin	i
ɪ	I	근전설 비원순 근고모음	vind	i
oː	o:	후설 원순 중고장모음	våt	o
ɔ	O	후설 원순 중저모음	vått	O
uː	u:	후설 원순 고장모음	bok	u
ʊ	U	근후설 원순 근고모음	bukk	u
yː	y:	전설 원순 고장모음	lyn	u
ʏ	Y	근전설 원순 근고모음	lynne	u
추가 기호				
ˈ	"	제1강세	Alabama	
ˌ	%	제2강세	Alabama	
.	.	음절 경계	A.la.ba.ma	

폴란드어(pl-PL)

다음 표에는 Amazon Polly에서 지원하는 폴란드어 음성에 대해 IPA(International Phonetic Alphabet) 음소와 X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet) 기호 및 해당하는 viseme이 나와 있습니다.

음소/Viseme 표

IPA	X-SAMPA	설명	예	Viseme
자음				
b	b	유성 양순 파열음	bobas, belka	p
d	d	유성 치경 파열음	dar, do	t
ʣ	dz	유성 치경 파찰음	dzwon, widzowie	s
ʥ	dz\	유성 치경구개 파찰음	dźwięk	J
ʤ	dz`	유성 권설 파찰음	dżem, dżungla	S
f	f	무성 순치 마찰음	furtka, film	f
g	g	유성 연구개 파열음	gazeta, waga	k
h	h	무성 성문 마찰음	chleb, handel	k
j	j	경구개 접근음	jak, maja	i
k	k	무성 연구개 파열음	kura, marek	k
l	l	치경 설측 접근음	lipa, alicja	t
m	m	양순 비음	matka, molo	p
n	n	치경 비음	norka	t
ɲ	J	경구개 비음	koń, toruń	J
p	p	무성 양순 파열음	pora, stop	p
r	r	치경 전동음	rok, park	r
s	s	무성 치경 마찰음	sum, pas	s

IPA	X-SAMPA	설명	예	Viseme
ɕ	s\	무성 치경구개 마찰음	śruba, śnieg	J
ʂ	s`	무성 권설 마찰음	szum, masz	S
t	t	무성 치경 파열음	tok, stół	t
t͡s	ts	무성 치경 파찰음	car, co	s
t͡ɕ	ts\	무성 치경구개 파찰음	ćma, mieć	J
t͡ʂ	ts`	무성 권설 파찰음	czas, raczej	S
v	v	유성 순치 마찰음	worek, mewa	f
w	w	양순 연구개 접근음	łaska, mało	u
z	z	유성 치경 마찰음	zero	s
ʑ	z\	유성 치경구개 마찰음	źrebię, bieliźnie	J
ʐ	z`	유성 권설 마찰음	żar, żona	S
모음				
a	a	전설 비원순 저모음	ja	a
ɛ	E	전설 비원순 중저모음	echo	E
ɛ̃	E~	전설 비원순 비(nasal) 중저모음	węże	E
i	i	전설 비원순 고모음	ile	i
ɔ	O	후설 원순 중저모음	oczy	O

IPA	X-SAMPA	설명	예	Viseme
ɔ̃	O~	후설 원순 비(nasal) 중저모음	wąż	O
u	u	후설 원순 고모음	uczta	u
ɨ	1	중설 비원순 고모음	byk	i
추가 기호				
ˈ	"	제1강세	Alabama	
ˌ	%	제2강세	Alabama	
.	.	음절 경계	A.la.ba.ma	

포르투갈어(pt-PT)

다음 표에는 Amazon Polly에서 지원하는 포르투갈어 음성에 대해 IPA(International Phonetic Alphabet) 음소와 X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet) 기호 및 해당하는 viseme이 나와 있습니다.

음소/Viseme 표

IPA	X-SAMPA	설명	예	Viseme
자음				
ɾ	4	치경 탄음	pira	t
b	b	유성 양순 파열음	dato	p
d	d	유성 치경 파열음	dato	t
f	f	무성 순치 마찰음	facto	f
g	g	유성 연구개 파열음	gato	k
j	j	경구개 접근음	paraguay	i

IPA	X-SAMPA	설명	예	Viseme
k	k	무성 연구개 파열음	cacto	k
l	l	치경 설측 접근음	galo	t
ʎ	L	경구개 설측 접근음	galho	J
m	m	양순 비음	mato	p
n	n	치경 비음	nato	t
ɲ	J	경구개 비음	pinha	J
p	p	무성 양순 파열음	pato	p
ʁ	R\	구개수 전동음	barroso	k
s	s	무성 치경 마찰음	saca	s
ʃ	S	무성 후치경 마찰음	chato	S
t	t	무성 치경 파열음	tacto	t
v	v	유성 순치 마찰음	vaca	f
w	w	양순 연구개 접근음	mau	u
z	z	유성 치경 마찰음	zaca	s
ʒ	Z	유성 후치경 마찰음	jacto	S
모음				
a	a	전설 비원순 저모음	parto	a
ã	a~	전설 비원순 비(nasal) 저모음	pega	a
e	e	전설 비원순 중고모음	pega	e

IPA	X-SAMPA	설명	예	Viseme
ẽ	e~	전설 비원순 비(nasal) 중고모음	movem	e
ɛ	E	전설 비원순 중저모음	café	E
i	i	전설 비원순 고모음	lingueta	i
ĩ	i~	전설 비원순 비(nasal) 고모음	cinto	i
o	o	후설 원순 중고모음	poder	o
õ	o~	후설 원순 비(nasal) 중고모음	compra	o
ɔ	O	후설 원순 중저모음	cotó	O
u	u	후설 원순 고모음	fui	u
ũ	u~	후설 원순 비(nasal) 고모음	sunto	u
추가 기호				
ˈ	"	제1강세	Alabama	
ˌ	%	제2강세	Alabama	
.	.	음절 경계	A.la.ba.ma	

포르투갈어(브라질)(pt-BR)

다음 표에는 Amazon Polly에서 지원하는 브라질 포르투갈어 음성에 대해 IPA(International Phonetic Alphabet) 음소와 X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet) 기호 및 해당하는 viseme이 나와 있습니다.

음소/Viseme 표

IPA	X-SAMPA	설명	예	Viseme
자음				
ɾ	4	치경 탄음	pira	t
b	b	유성 양순 파열음	bato	p
d	d	유성 치경 파열음	dato	t
͡dʒ	dZ	유성 후치경 파찰음	idade	S
f	f	무성 순치 마찰음	facto	f
g	g	유성 연구개 파열음	gato	k
j	j	경구개 접근음	paraguay	i
k	k	무성 연구개 파열음	cacto	k
l	l	치경 설측 접근음	galo	t
ʎ	L	경구개 설측 접근음	galho	J
m	m	양순 비음	mato	p
n	n	치경 비음	nato	t
ɲ	J	경구개 비음	pinha	J
p	p	무성 양순 파열음	pato	p
s	s	무성 치경 마찰음	saca	s
ʃ	S	무성 후치경 마찰음	chato	S
t	t	무성 치경 파열음	tacto	t
͡tʃ	tS	무성 후치경 파찰음	noite	S
v	v	유성 순치 마찰음	vaca	f

IPA	X-SAMPA	설명	예	Viseme
w	w	양순 연구개 접근음	mau	u
χ	X	무성 구개수 마찰음	carro	k
z	z	유성 치경 마찰음	zaca	s
ʒ	Z	유성 후치경 마찰음	jacto	S
모음				
a	a	전설 비원순 저모음	parto	a
ã	a~	전설 비원순 비(nasal) 저모음	pensamos	a
e	e	전설 비원순 중고모음	pega	e
ẽ	e~	전설 비원순 비(nasal) 중고모음	movem	e
ɛ	E	전설 비원순 중저모음	café	E
i	i	전설 비원순 고모음	lingueta	i
ĩ	i~	전설 비원순 비(nasal) 고모음	cinto	i
o	o	후설 원순 중고모음	poder	o
õ	o~	후설 원순 비(nasal) 중고모음	compra	o
ɔ	O	후설 원순 중저모음	cotó	O
u	u	후설 원순 고모음	fui	u

IPA	X-SAMPA	설명	예	Viseme
ũ	u~	후설 원순 비(nasal) 고모음	sunto	u
추가 기호				
ˈ	"	제1강세	Alabama	
ˌ	%	제2강세	Alabama	
.	.	음절 경계	A.la.ba.ma	

루마니아어(ro-RO)

다음 표에는 Amazon Polly에서 지원하는 루마니아어 음성에 대해 IPA(International Phonetic Alphabet) 음소와 X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet) 기호 및 해당하는 viseme이 나와 있습니다.

음소/Viseme 표

IPA	X-SAMPA	설명	예	Viseme
자음				
b	b	유성 양순 파열음	bubă	p
d	d	유성 치경 파열음	după	t
ʤ	dZ	유성 후치경 파찰음	george	S
f	f	무성 순치 마찰음	afacere	f
g	g	유성 연구개 파열음	agri#	k
h	h	무성 성문 마찰음	harpă	k
j	j	경구개 접근음	baie	i
k	k	무성 연구개 파열음	co#	k

IPA	X-SAMPA	설명	예	Viseme
l	l	치경 설측 접근음	lampa	t
m	m	양순 비음	mama	p
n	n	치경 비음	nor	t
p	p	무성 양순 파열음	pilă	p
r	r	치경 전동음	rampă	r
s	s	무성 치경 마찰음	soare	s
ʃ	S	무성 후치경 마찰음	ma#ină	S
t	t	무성 치경 파열음	tata	t
͡ts	ts	무성 치경 파찰음	#ară	s
͡tʃ	tS	무성 후치경 파찰음	ceai	S
v	v	유성 순치 마찰음	via#ă	f
w	w	양순 연구개 접근음	beau	u
z	z	유성 치경 마찰음	mozol	s
ʒ	Z	유성 후치경 마찰음	joacă	S
모음				
ə	@	중설 중모음	babă	@
a	a	전설 비원순 저모음	casa	a
e	e	전설 비원순 중고모음	elan	e
e̯	e_^	비성절 전설 비원순 중고모음	beau	e

IPA	X-SAMPA	설명	예	Viseme
i	i	전설 비원순 고모음	mie	i
o	o	후설 원순 중고모음	ură	o
oa	o_^a	이중 모음	oare	o
u	u	후설 원순 고모음	unde	u
ɨ	1	중설 비원순 고모음	România	i
추가 기호				
ˈ	"	제1강세	Alabama	
ˌ	%	제2강세	Alabama	
.	.	음절 경계	A.la.ba.ma	

러시아어(ru-RU)

다음 표에는 Amazon Polly에서 지원하는 러시아어 음성에 대해 IPA(International Phonetic Alphabet) 음소와 X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet) 기호 및 해당하는 viseme이 나와 있습니다.

음소/Viseme 표

IPA	X-SAMPA	설명	예	Viseme
자음				
b	b	유성 양순 파열음	борт	p
bʲ	b'	경구개음화 유성 양순 파열음	бюро	p
d	d	유성 치경 파열음	дом	t

IPA	X-SAMPA	설명	예	Viseme
d^j	d'	경구개음화 유성 치경 파열음	дядя	t
f	f	무성 순치 마찰음	флаг	f
f^j	f'	경구개음화 무성 순치 마찰음	февраль	f
g	g	유성 연구개 파열음	нога	k
g^j	g'	경구개음화 유성 연구개 파열음	герой	k
j	j	경구개 접근음	дизайн, ящик	i
k	k	무성 연구개 파열음	кот	k
k^j	k'	경구개음화 무성 연구개 파열음	кино	k
l	l	치경 설측 접근음	лампа	t
l^j	l'	경구개음화 치경 설측 접근음	лес	t
m	m	양순 비음	мама	p
m^j	m'	경구개음화 양순 비음	мяч	p
n	n	치경 비음	нос	t
n^j	n'	경구개음화 치경 비음	няня	t
p	p	무성 양순 파열음	папа	p
p^j	p'	경구개음화 무성 양순 파열음	перо	p

IPA	X-SAMPA	설명	예	Viseme
r	r	치경 전동음	роза	r
rʲ	r'	경구개음화 치경 전동음	рюмка	r
s	s	무성 치경 마찰음	сыр	s
sʲ	s'	경구개음화 무성 치경 마찰음	сердце, русь	s
ɕː	s\:	무성 치경구개 마찰장음	щека	J
ʂ	s`	무성 권설 마찰음	шум	S
t	t	무성 치경 파열음	точка	t
tʲ	t'	경구개음화 무성 치경 파열음	тётя	t
t͡s	ts	무성 치경 파찰음	царь	s
t͡ɕ	ts\	무성 치경구개 파찰음	час	J
v	v	유성 순치 마찰음	вор	f
vʲ	v'	경구개음화 유성 순치 마찰음	верфь	f
x	x	무성 연구개 마찰음	хор	k
xʲ	x'	경구개음화 무성 연구개 마찰음	химия	k
z	z	유성 치경 마찰음	зуб	s
zʲ	z'	경구개음화 유성 치경 마찰음	зима	s

IPA	X-SAMPA	설명	예	Viseme
ʑː	z\:	유성 치경구개 마찰장음	уезжать	J
ʐ	z`	유성 권설 마찰음	жена	S
모음				
ə	@	중설 중모음	канарейка	@
a	a	전설 비원순 저모음	два, яблоко	a
e	e	전설 비원순 중고모음	печь	e
ɛ	E	전설 비원순 중저모음	это	E
i	i	전설 비원순 고모음	один, четыре	i
o	o	후설 원순 중고모음	кот	o
u	u	후설 원순 고모음	муж, вьюга	u
ɨ	1	중설 비원순 고모음	мышь	i

스페인어(es-ES)

다음 표에는 Amazon Polly에서 지원하는 스페인어 음성에 대해 IPA(International Phonetic Alphabet) 음소와 X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet) 기호 및 해당하는 viseme이 나와 있습니다.

음소/Viseme 표

IPA	X-SAMPA	설명	예	Viseme
자음				

IPA	X-SAMPA	설명	예	Viseme
ɾ	4	치경 탄음	pero, bravo, amor, eterno	t
b	b	유성 양순 파열음	bestia	p
β	B	유성 양순 마찰음	bebé	B
d	d	유성 치경 파열음	cuando	t
ð	D	유성 치 마찰음	arder	T
f	f	무성 순치 마찰음	fase, café	f
g	g	유성 연구개 파열음	gato, lengua, guerra	k
ɣ	G	유성 연구개 마찰음	trigo, Argos	k
j	j	경구개 접근음	hacia, tierra, radio, viuda	i
ʝ	j\	유성 경구개 마찰음	enhielar, sayo, inyectado, desyerba	J
k	k	무성 연구개 파열음	caña, laca, quisimos	k
l	l	치경 설측 접근음	lino, calor, principal	t
ʎ	L	경구개 설측 접근음	llave, pollo	J
m	m	양순 비음	madre, comer, anfibio	p
n	n	치경 비음	nido, anillo, sin	t
ɲ	J	경구개 비음	cabaña, ñoquis	J

IPA	X-SAMPA	설명	예	Viseme
ŋ	N	연구개 비음	cinco, venga	k
p	p	무성 양순 파열음	pozo, topo	p
r	r	치경 전동음	perro, enrachado	r
s	s	무성 치경 마찰음	saco, casa, puertas	s
t	t	무성 치경 파열음	tamiz, átomo	t
ʧ	tS	무성 후치경 파찰음	chubasco	S
θ	T	무성 치 마찰음	cereza, zorro, lacero, paz	T
w	w	양순 연구개 접근음	fuego, fuimos, cuota, cuadro	u
x	x	무성 연구개 마찰음	jamón, general, suje, reloj	k
z	z	유성 치경 마찰음	rasgo, mismo	s
모음				
a	a	전설 비원순 저모음	tanque	a
e	e	전설 비원순 중고모음	peso	e
i	i	전설 비원순 고모음	cinco	i
o	o	후설 원순 중고모음	bosque	o
u	u	전설 비원순 중고모음	publicar	u
추가 기호				

IPA	X-SAMPA	설명	예	Viseme
ˈ	"	제1강세	Alabama	
ˌ	%	제2강세	Alabama	
.	.	음절 경계	A.la.ba.ma	

스페인어(멕시코)(es-MX)

다음 표에는 Amazon Polly에서 지원하는 멕시코 스페인어 음성에 대해 IPA(International Phonetic Alphabet) 음소와 X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet) 기호 및 해당하는 viseme이 나와 있습니다.

음소/Viseme 표

IPA	X-SAMPA	설명	예	Viseme
자음				
ɾ	4	치경 탄음	pero, bravo, amor, eterno	t
b	b	유성 양순 파열음	bestia	p
β	B	유성 양순 마찰음	bebé	B
d	d	유성 치경 파열음	cuando	t
ð	D	유성 치 마찰음	arder	T
f	f	무성 순치 마찰음	fase, café	f
g	g	유성 연구개 파열음	gato, lengua, guerra	k
ɣ	G	유성 연구개 마찰음	trigo, Argos	k
j	j	경구개 접근음	hacia, tierra, radio, viuda	i

IPA	X-SAMPA	설명	예	Viseme
ʝ	j\	유성 경구개 마찰음	enhielar, sayo, inyectado, desyerba	J
k	k	무성 연구개 파열음	caña, laca, quisimos	k
l	l	설측 치경 접근음	lino, calor, principal	t
m	m	양순 비음	madre, comer, anfibio	p
n	n	치경 비음	nido, anillo, sin	t
ɲ	J	경구개 비음	cabaña, ñoquis	J
ŋ	N	연구개 비음	angosto, increíble	k
p	p	무성 양순 파열음	pozo, topo	p
r	r	치경 전동음	perro, enrachado	r
s	s	무성 치경 마찰음	saco, casa, puertas	s
ʃ	S	무성 후치경 마찰음	표시, 플래시	S
t	t	무성 치경 파열음	tamiz, átomo	t
͡tʃ	tS	무성 후치경 파찰음	chubasco	S
w	w	양순 연구개 접근음	fuego, fuimos, cuota, cuadro	u
x	x	무성 연구개 마찰음	jamón, general, suje, reloj	k
z	z	유성 치경 마찰음	rasgo, mismo	s

모음

IPA	X-SAMPA	설명	예	Viseme
a	a	중앙의 열린 비원순 저모음	tanque	a
e	e	전설 비원순 중고모음	peso	e
i	i	전설 비원순 고모음	cinco	i
o	o	후설 원순 중고모음	bosque	o
u	u	후설 원순 고모음	publicar	u
추가 기호				
ˈ	"	제1강세	Alabama	
ˌ	%	제2강세	Alabama	
.	.	음절 경계	A.la.ba.ma	

스페인어(미국)(es-US)

다음 표에는 Amazon Polly에서 지원하는 미국 스페인어 음성에 대해 IPA(International Phonetic Alphabet) 음소와 X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet) 기호 및 해당하는 viseme이 나와 있습니다.

음소/Viseme 표

IPA	X-SAMPA	설명	예	Viseme
자음				
ɾ	4	치경 탄음	pero, bravo, amor, eterno	t
b	b	유성 양순 파열음	bestia	p
β	B	유성 양순 마찰음	bebé	B

IPA	X-SAMPA	설명	예	Viseme
d	d	유성 치경 파열음	cuando	t
ð	D	유성 치 마찰음	arder	T
f	f	무성 순치 마찰음	fase, café	f
g	g	유성 연구개 파열음	gato, lengua, guerra	k
ɣ	G	유성 연구개 마찰음	trigo, Argos	k
j	j	경구개 접근음	hacia, tierra, radio, viuda	i
ʝ	j\	유성 경구개 마찰음	enhielar, sayo, inyectado, desyerba	J
k	k	무성 연구개 파열음	caña, laca, quisimos	k
l	l	설측 치경 접근음	lino, calor, principal	t
m	m	양순 비음	madre, comer, anfibio	p
n	n	치경 비음	nido, anillo, sin	t
ɲ	J	경구개 비음	cabaña, ñoquis	J
ŋ	N	연구개 비음	angosto, increíble	k
p	p	무성 양순 파열음	pozo, topo	p
r	r	치경 전동음	perro, enrachado	r
s	s	무성 치경 마찰음	saco, casa, puertas	s
ʃ	S	무성 후치경 마찰음	표시, 플래시	S

IPA	X-SAMPA	설명	예	Viseme
t	t	무성 치경 파열음	tamiz, átomo	t
ʧ	tS	무성 후치경 파찰음	chubasco	S
w	w	양순 연구개 접근음	fuego, fuimos, cuota, cuadro	u
x	x	무성 연구개 마찰음	jamón, general, suje, reloj	k
z	z	유성 치경 마찰음	rasgo, mismo	s
모음				
a	a	중앙의 열린 비원순 저모음	tanque	a
e	e	전설 비원순 중고모음	peso	e
i	i	전설 비원순 고모음	cinco	i
o	o	후설 원순 중고모음	bosque	o
u	u	후설 원순 고모음	publicar	u
추가 기호				
ˈ	"	제1강세	Alabama	
ˌ	%	제2강세	Alabama	
.	.	음절 경계	A.la.ba.ma	

스웨덴어(sv-SE)

다음 표에는 Amazon Polly에서 지원하는 스웨덴어 음성에 대해 IPA(International Phonetic Alphabet) 음소와 X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet) 기호 및 해당하는 viseme이 나와 있습니다.

음소/Viseme 표

IPA	X-SAMPA	설명	예	Viseme
자음				
b	b	유성 양순 파열음	bil	p
d	d	유성 치경 파열음	dal	t
ɖ	d`	유성 권설 파열음	bord	t
f	f	무성 순치 마찰음	fil	f
g	g	유성 연구개 파열음	gås	k
h	h	무성 성문 마찰음	hal	k
j	j	경구개 접근음	jag	i
k	k	무성 연구개 파열음	kal	k
l	l	치경 설측 접근음	lös	t
ɭ	l`	권설 설측 접근음	härlig	t
m	m	양순 비음	mil	p
n	n	치경 비음	nålar	t
ɳ	n`	권설 비음	barn	t
ŋ	N	연구개 비음	ring	k
p	p	무성 양순 파열음	pil	p

IPA	X-SAMPA	설명	예	Viseme
r	r	치경 전동음	ris	r
s	s	무성 치경 마찰음	sil	s
ɕ	s\	무성 치경구개 마찰음	tjock	J
ʂ	s`	무성 권설 마찰음	fors, schlager	S
t	t	무성 치경 파열음	tal	t
ʈ	t`	무성 권설 파열음	hjort	t
v	v	유성 순치 마찰음	vår	f
w	w	양순 연구개 접근음	aula, airways	u
ɧ	x\	무성 경구개-연구개 마찰음	sjuk	k
모음				
ø	2	전설 원순 중고모음	föll, förr	o
ø	2:	전설 원순 중고장모음	föl, nöt, för	o
ɵ	8	중설 원순 중고모음	buss, full	o
ə	@	중설 중모음	pojken	@
ʉː	}:	중설 원순 고장모음	hus, ful	u
a	a	전설 비원순 저모음	hall, matt	a
æ	{	전설 비원순 근저모음	herr	a

IPA	X-SAMPA	설명	예	Viseme
ɑ:	A:	후설 비원순 저장모음	hal, mat	a
e:	e:	전설 비원순 중고장모음	vet, hel	e
ɛ	E	전설 비원순 중저모음	vett, rätt, hetta, häll	E
ɛ:	E:	전설 비원순 중저장모음	säl, häl, här	E:
i:	i:	전설 비원순 고장모음	vit, sil	i:
ɪ	I	근전설 비원순 근고모음	vitt, sill	i
o:	o:	후설 원순 중고장모음	hål, mål	o
ɔ	O	후설 원순 중저모음	håll, moll	O
u:	u:	후설 원순 고장모음	sol, bot	u
ʊ	U	근후설 원순 근고모음	bott	u
y	y	전설 원순 고모음	bytt	u
y:	y:	전설 원순 고장모음	syl, syl	u
추가 기호				
ˈ	"	제1강세	Alabama	
ˌ	%	제2강세	Alabama	
.	.	음절 경계	A.la.ba.ma	

터키어(tr-TR)

다음 표에는 Amazon Polly에서 지원하는 터키어 음성에 대해 IPA(International Phonetic Alphabet) 음소와 X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet) 기호 및 해당하는 viseme이 나와 있습니다.

음소/Viseme 표

IPA	X-SAMPA	설명	예	Viseme
자음				
ɾ	4	치경 탄음	durum	t
ɾ̥	4_0_r	무성 마찰 치경 탄음	bir	t
ɾ̞	4_r	마찰 치경 탄음	raf	t
b	b	유성 양순 파열음	raf	p
c	c	무성 경구개 파열음	kedi	k
d	d	유성 치경 파열음	dede	t
ʤ	dZ	유성 후치경 파찰음	cam	S
f	f	무성 순치 마찰음	fare	f
g	g	유성 연구개 파열음	galibi	k
h	h	무성 성문 마찰음	hasta	k
j	j	경구개 접근음	yat	i
ɟ	J\	유성 경구개 파열음	genç	J
k	k	무성 연구개 파열음	akıl	k
l	l	치경 설측 접근음	lale	t

IPA	X-SAMPA	설명	예	Viseme
ɬ	5	연구개음화 치경 설측 접근음	labirent	t
m	m	양순 비음	maaş	p
n	n	치경 비음	anı	t
p	p	무성 양순 파열음	ip	p
s	s	무성 치경 마찰음	ses	s
ʃ	S	무성 후치경 마찰음	aşı	S
t	t	무성 치경 파열음	ütü	t
͡tʃ	tS	무성 후치경 파찰음	çaba	S
v	v	유성 순치 마찰음	ekvator, kahveci, akvaryum, isveçli, teşviki, cetvel	f
z	z	유성 치경 마찰음	ver	s
ʒ	Z	유성 후치경 마찰음	azık	S
모음				
ø	2	전설 원순 중고모음	göl	0
œ	9	전설 원순 중저모음	banliyö	O
a	a	전설 비원순 저모음	kal	a
a:	a:	전설 비원순 저장모음	davacı	a
æ	{	전설 비원순 근저모음	özlem, güvenlik, gürel, somersault	a

IPA	X-SAMPA	설명	예	Viseme
e	e	전설 비원순 중고모음	keçi	e
ɛ	E	전설 비원순 중저모음	dede	E
i	i	전설 비원순 고모음	bir	i
i:	i:	전설 비원순 고장모음	izah	i
ɪ	I	근전설 비원순 근고모음	keçi	i
ɯ	M USD	후설 비원순 고모음	kıl	i
o	o	후설 원순 중고모음	kol	o
o:	o:	후설 원순 중고장모음	dolar	o
u	u	후설 원순 고모음	durum	u
u:	u:	후설 원순 고장모음	ruhum	u
ʊ	U	근후설 원순 근고모음	dolu	u
y	y	전설 원순 고모음	güvenlik	u
ʏ	Y	근전설 원순 근고모음	aşı	u
추가 기호				
ˈ	"	제1강세	Alabama	
ˌ	%	제2강세	Alabama	

IPA	X-SAMPA	설명	예	Viseme
.	.	음절 경계	A.la.ba.ma	

웨일스어(cy-GB)

다음 표에는 Amazon Polly에서 지원하는 웨일스어 음성에 대해 IPA(International Phonetic Alphabet) 음소와 X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet) 기호 및 해당하는 viseme이 나와 있습니다.

음소/Viseme 표

IPA	X-SAMPA	설명	예	Viseme
자음				
b	b	유성 양순 파열음	baban	p
d	d	유성 치경 파열음	deg	t
ʤ	dZ	유성 후치경 파찰음	garej	S
ð	D	유성 치 마찰음	deuddeg	T
f	f	무성 순치 마찰음	ffacs	f
g	g	유성 연구개 파열음	gadael	k
h	h	무성 성문 마찰음	haearn	k
j	j	경구개 접근음	astudio	i
k	k	무성 연구개 파열음	cant	k
l	l	치경 설측 접근음	lan	t
ɬ	K USD	무성 치경 설측 마찰음	llan	t
m	m	양순 비음	mae	p

IPA	X-SAMPA	설명	예	Viseme
m̥	m_0	무성 양순 비음	ymhen	p
n	n	치경 비음	naw	t
n̥	n_0	무성 치경 비음	anhawster	t
ŋ	N	연구개 비음	argyfwng	k
ŋ̊	N_0	무성 연구개 비음	anghenion	k
p	p	무성 양순 파열음	pump	p
r	r	치경 전동음	rhoi	r
r̥	r_0	무성 치경 전동음	garw	r
s	s	무성 치경 마찰음	saith	s
ʃ	S	무성 후치경 마찰음	siawns	S
t	t	무성 치경 파열음	tegan	t
t͡ʃ	tS	무성 후치경 파찰음	cytsain	S
θ	T	무성 치 마찰음	aberth	T
v	v	유성 순치 마찰음	prawf	f
w	w	양순 연구개 접근음	rhagweld	u
χ	X	무성 구개수 마찰음	chwech	k
z	z	유성 치경 마찰음	aids	s
ʒ	Z	유성 후치경 마찰음	rouge	S
모음				
ə	@	중설 중모음	ychwanega	@

IPA	X-SAMPA	설명	예	Viseme
a	a	전설 비원순 저모음	acen	a
ai	ai	이중 모음	dau	a
au	au	이중 모음	awdur	a
ɑː	A:	후설 비원순 저장모음	mab	a
ɑːɨ	A:1	이중 모음	aelod	a
eː	e:	전설 비원순 중고장모음	peth	e
ɛ	E	전설 비원순 중저모음	pedwar	E
ɛi	Ei	이중 모음	beic	E
iː	i:	전설 비원순 고장모음	tri	i
ɪ	I	근전설 비원순 근고모음	miliwn	i
ɨu	1u	이중 모음	unigryw	i
oː	o:	후설 원순 중고장모음	oddi	o
ɔ	O	후설 원순 중저모음	oddieithr	O
ɔi	Oi	이중 모음	troi	O
ɔu	Ou	이중 모음	rownd	O
uː	u:	후설 원순 고장모음	cwch	u

IPA	X-SAMPA	설명	예	Viseme
ʊ	U	근후설 원순 근고모음	acwstig	u
ʊi	Ui	이중 모음	wyth	u
추가 기호				
ˈ	"	제1강세	Alabama	
ˌ	%	제2강세	Alabama	
.	.	음절 경계	A.la.ba.ma	

아마존 Polly 보이스 엔진

Amazon Polly에는 입력 텍스트를 실제와 같은 음성으로 변환하는 네 가지 음성 엔진이 있습니다. 여기에는 제너레이티브, 롱폼, 뉴럴, 스탠다드 등이 포함됩니다. Amazon Polly 음성을 사용하려면 엔진 및 음성 합성 API 작업을 선택합니다. 그런 다음 엔진이 합성할 입력 텍스트를 제공하고 오디오 출력 형식을 선택합니다. Amazon Polly는 이러한 입력이 주어지면 제공된 텍스트를 고품질 음성 오디오 스트림으로 합성합니다.

다음 섹션에는 Amazon Polly에서 제공하는 음성 엔진에 대한 세부 정보가 포함되어 있습니다.

주제

- 제너레이티브 보이스
- 롱폼 음성
- 뉴럴 보이스
- 표준 음성

제너레이티브 보이스

Amazon Polly의 제너레이티브 text-to-speech (TTS) 엔진은 Amazon Polly 콘솔을 통해 사용할 수 있는 가장 인간적이고 감정적이며 적응력이 뛰어난 대화 음성을 제공합니다.

제너레이티브 엔진은 현재까지 출시된 Amazon Polly TTS 모델 중 가장 큰 모델입니다. 원시 텍스트를 음성 코드로 변환하는 10억 파라미터 변환기를 배포한 다음, 이러한 음성 코드를 점진적이고 스트리밍 가능한 방식으로 파형으로 변환하는 컨볼루션 기반 디코더를 배포합니다. 이 방법은 다양한 음성, 언어 및 스타일로 구성된 공개적으로 사용 가능한 독점 데이터의 양이 증가함에 따라 학습했을 때 널리 보고된 LLM (Large Language Model) 의 새로운 기능을 보여줍니다.

제너레이티브 엔진은 사람의 목소리와 매우 유사한 방식으로 감정적으로 몰입되고 자기 주장이 강하며 매우 구어적인 합성 음성을 생성합니다. 이러한 음성을 지식이 풍부한 고객 도우미, 가상 트레이너 또는 인간에 가까운 합성 음성을 사용하는 광고주로 사용할 수 있습니다.

> **Note**
>
> 이러한 음성의 기반이 되는 state-of-the-art 기술은 언어 및 음성 모델링을 위한 제너레이티브 AI 패러다임에 속합니다. 이 기술의 부작용은 훈련 데이터와 모델이 업데이트되면 음성이 약간 다르게 들릴 수 있다는 것입니다. 모델 업데이트로 전반적인 음질이 개선되더라도 말이죠. 이

는 오랜 기간 동안 다양한 콘텐츠 부분을 합성하는 사용 사례 (예: 팟캐스트 시즌) 에 영향을 미칠 수 있습니다.

사용 가능한 제너레이티브 보이스

Amazon Polly는 현재 생성형 변형으로 여성 2명과 남성 1명의 영어 음성을 제공합니다. 이러한 제너레이티브 음성은 대화형 NTTS 변형으로도 사용할 수 있습니다.

	언어	언어 코드	이름/ID	Gender
1	영어 (영국)	en-GB	Amy	여성
2	영어(미국)	en-US	Matthew	남성
			Ruth	여성

> **Note**
>
> 제너레이티브 보이스 비용은 Amazon Polly 요금 정보 페이지에 명시되어 있습니다.

기능 및 지역 호환성

Amazon Polly 생성 음성은 다음 지역에서 사용할 수 있습니다.

- 미국 동부(버지니아 북부) 리전
- 다른 지역은 사용할 수 없습니다.

제너레이티브 보이스에는 다음 기능이 지원됩니다.

- 실시간 및 비동기 스피치 합성 작업
- 제너레이티브 엔진에서는 뉴스캐스터 발언 스타일이 지원되지 않습니다.
- 전부는 아니지만 많은 SSML 태그가 Amazon Polly에서 지원됩니다. NTTS를 지원하는 SSML 태그에 대한 자세한 내용은 지원되는 SSML 태그를 참조하세요.
- 표준 음성의 경우와 마찬가지로, 다양한 샘플링 속도 중에서 선택하여 애플리케이션의 대역폭과 오디오 품질을 최적화할 수 있습니다. 표준 음성과 신경 음성에 맞는 샘플링 속도는 8kHz, 16kHz,

22kHz 또는 24kHz입니다. 표준 음성의 기본값은 22kHz이고 제너레이티브 보이스의 기본값은 24kHz입니다. Amazon Polly는 MP3, OGG(Vorbis) 및 원시 PCM 오디오 스트림 형식을 지원합니다.

- 새로운 Amazon Polly 제너레이티브 보이스의 지연 시간은 100ms입니다.

스피치 마크 생성에 대한 지원은 현재 제공되지 않습니다.

> **Note**
>
> 드물게 모델 환각 현상이 발생하고 토큰별로 음성 토큰을 렌더링하는 제너레이티브 엔진의 모델 동작이 발생하는 경우 비상 정지 메커니즘이 적용됩니다. 내장된 메커니즘은 모델이 더 이상 음성을 렌더링하지 못하게 합니다. 이 안전 기능은 모델이 환각을 일으킬 가능성이 있는 데이터 분석 (보통 문장 끝부분) 을 기반으로 합니다.
> 모델이 환각을 일으킬 것이라고 생각하다가 생성 단계에서 단어를 잘라서 단어의 절반을 렌더링하는 경우가 있을 수 있습니다. 이로 인해 부적절한 결과가 나올 수 있습니다.

콘솔의 제너레이티브 엔진 사용

Amazon Polly 콘솔 또는 을 통해 Amazon Polly 제너레이티브 음성에 액세스할 수 있습니다. AWS CLI 콘솔에서 제너레이티브 엔진을 선택한 다음 목록에서 해당하는 제너레이티브 음성을 선택하면 해당 음성으로 합성된 음성을 들을 수 있습니다. 및 API 작업을 사용하여 제너레이티브 보이스를 탐색할 수도 있습니다. `SynthesizeSpeech` `StartSpeechSynthesisTask` API 작업의 경우 API 요청에서 엔진 및 음성 이름을 지정할 수 있습니다. Python을 사용하여 빠르게 시작하는 코드 예제는 Python 예제를 참조하십시오.

콘솔에서 제너레이티브 엔진을 사용하려면

1. https://console.aws.amazon.com/polly/에서 Amazon Polly 콘솔을 엽니다.
2. Amazon Polly 콘솔에서 제너레이티브 엔진을 선택합니다.
3. 음성 드롭다운 메뉴에서 원하는 음성을 선택합니다.
4. 원하는 텍스트로 TTS 오디오를 생성하세요.

> **Note**
>
> **SynthesizeSpeech**및 **StartSpeechSynthesisTask**API 작업과 함께 제너레이티브 음성을 사용할 수도 있습니다. API 작업의 경우 고객은 API 요청에서 엔진 및 음성 이름을 지정할 수 있습니다. 여기에서 더 많은 퀵 스타트 코드 샘플을 찾을 수 있습니다.

롱폼 음성

Amazon Polly에는 사람과 비슷하고 표현력이 뛰어나며 감정적으로 능숙한 음성을 생성하는 장편 엔진이 있습니다. 롱폼 음성은 뉴스 기사, 교육 자료 또는 마케팅 비디오와 같은 긴 콘텐츠에서 청취자의 관심을 끌도록 설계되었습니다.

Amazon Polly 장편 음성은 최첨단 딥 러닝 TTS 기술로 개발되었습니다. 이 모델은 인간 언어의 음소, 운율, 억양 및 기타 음성 및 음향 측면을 따라하도록 학습하여 매우 자연스러운 음성 출력을 제공합니다.

Long-Form 엔진은 텍스트 임베딩을 사용하여 텍스트의 의미를 해석합니다. 롱폼 엔진은 텍스트 임베딩을 사용하여 자연스러운 목소리의 정확한 강조, 일시 중지 및 어조를 생성할 수 있습니다. 그 결과 인간의 의사소통에 존재하는 다양한 감정적 요소를 모두 결합한 목소리가 탄생했습니다. 여기에는 놀라움을 흉내내거나 대화를 내레이션과 구별하는 것도 포함됩니다. 이 모든 것이 합쳐져 살아 있는 사람처럼 들리는 프리미엄 스피치 산물이 생성됩니다.

> **Note**
>
> 이러한 음성의 기반이 되는 state-of-the-art 기술은 언어 및 음성 모델링을 위한 제너레이티브 AI 패러다임에 속합니다. 이 기술의 부작용은 훈련 데이터와 모델이 업데이트되면 음성이 약간 다르게 들릴 수 있다는 것입니다. 모델 업데이트로 전반적인 음질이 개선되더라도 말이죠. 이는 오랜 기간 동안 다양한 콘텐츠 부분을 합성하는 사용 사례 (예: 팟캐스트 시즌) 에 영향을 미칠 수 있습니다.

사용 가능한 장편 음색

Amazon Polly는 현재 미국 내 장편 음성 2명과 여성 1명의 남성 음성을 제공합니다. 이러한 롱폼 음성은 대화형 NTTS 변형으로도 사용할 수 있습니다.

	언어	언어 코드	이름/ID	Gender
1	영어(미국)	en-US	Danielle	여성
			Gregory	남성
			Ruth	여성

기능 및 지역 호환성

Amazon Polly 장편 음성은 다음 지역에서 사용할 수 있습니다.

- 미국 동부(버지니아 북부) 리전
- 기타 지역은 사용할 수 없습니다.

Amazon Polly 롱폼 엔진은 다음과 같은 기능을 지원합니다.

- 실시간 및 비동기 스피치 합성 작업
- 모든 스피치 마크
- 전부는 아니지만 많은 SSML 태그가 Amazon Polly에서 지원됩니다. NTTS를 지원하는 SSML 태그에 대한 자세한 내용은 지원되는 SSML 태그를 참조하세요.
- 100밀리초의 지연 시간
- 표준 음성의 경우와 마찬가지로, 다양한 샘플링 속도 중에서 선택하여 애플리케이션의 대역폭과 오디오 품질을 최적화할 수 있습니다. 표준, 롱폼 및 신경 음성에 맞는 샘플링 속도는 8kHz, 16kHz, 22kHz 또는 24kHz입니다. 표준 음성의 기본값은 22kHz이고 롱폼 및 신경 음성의 기본값은 24kHz입니다. Amazon Polly는 MP3, OGG(Vorbis) 및 원시 PCM 오디오 스트림 형식을 지원합니다.

> **Note**
>
> 긴 형식의 음성 요금은 Amazon Polly 가격 정보 페이지에 명시되어 있습니다.

콘솔에서 롱폼 엔진 사용

Amazon Polly 콘솔 또는 AWS CLI를 통해 Amazon Polly 롱폼 음성에 액세스할 수 있습니다.

콘솔에서 롱폼 엔진을 사용하려면

1. https://console.aws.amazon.com/polly/에서 Amazon Polly 콘솔을 엽니다.
2. Amazon Polly 콘솔에서 롱 폼 엔진을 선택합니다.
3. 음성 드롭다운 메뉴에서 원하는 음성을 선택합니다.
4. 원하는 텍스트로 TTS 오디오를 생성하세요.

> **Note**
>
> 긴 형식의 음성을 **SynthesizeSpeech**및 **StartSpeechSynthesisTask**API 작업과 함께 사용할 수도 있습니다. API 작업의 경우 고객은 API 요청에서 엔진 및 음성 이름을 지정할 수 있습니다. 여기에서 더 많은 퀵 스타트 코드 샘플을 찾을 수 있습니다.

뉴럴 보이스

Amazon Polly에는 표준 음성보다 훨씬 더 높은 품질의 음성을 생성할 수 있는 신경망 text-to-speech (NTTS) 엔진이 있습니다. 표준 TTS 음성은 연결 합성을 사용합니다. 표준 엔진은 녹음된 음성의 음소를 연결하여 매우 자연스러운 합성 음성을 생성합니다. 하지만 말은 변형될 수밖에 없고 파형 분절에 사용되는 기술로 인해 스피치 품질이 제한됩니다. Amazon Polly NTTS 엔진은 표준 연결 합성을 사용하여 음성을 생성하지 않습니다. 다음 두 부분으로 이루어져 있습니다.

- 신경망 — 일련의 음소 (가장 기본적인 언어 단위) 를 스펙트로그램 시퀀스로 변환하는 신경망. (스펙트그램은 다양한 주파수 대역의 에너지 준위를 보여주는 스냅샷입니다.)
- 보코더 — 스펙트로그램을 거의 연속적인 오디오 신호로 변환합니다.

신경망 TTS 시스템의 첫 번째 구성요소는 모델입니다. sequence-to-sequence 이 모델은 해당하는 입력에서만 결과를 만들 뿐만 아니라 입력 요소의 시퀀스가 함께 작동하는 방식도 고려합니다. 모델이 출력되는 스펙트로그램을 선택하면 주파수 대역이 사람의 뇌가 말을 처리할 때 사용하는 음향 기능을 강조합니다.

그런 다음 이 모델의 출력이 신경 보코더로 전달되고 스펙트로그램이 스피치 파형으로 변환됩니다. 범용 연쇄 합성 시스템을 구축하는 데 사용되는 대규모 데이터 세트를 대상으로 학습하면 이 sequence-to-sequence 접근 방식을 통해 더 고품질의 자연스러운 음성을 얻을 수 있습니다.

사용 가능한 뉴럴 보이스

뉴럴 음성은 33개 언어 및 언어 변형으로 제공됩니다. 다음 표에 음성이 나와 있습니다.

	언어 및 사투리	언어 코드	이름/ID	Gender
1	아랍어(걸프)	ar-AE	할라	여성
			자이드	남성
2	벨기에 네덜란드어(플랑드르)	nl-BE	Lisa	여성
3	카탈루냐어	ca-ES	Arlet	여성
4	중국어(광동어)	yue-CN	Hiujin	여성
5	표준 중국어	cmn-CN	Zhiyu	여성
6	덴마크어	da-DK	Sofie	여성
7	네덜란드어	nl-NL	Laura	여성
8	영어(호주)	en-AU	Olivia	여성
9	영어(영국)	en-GB	Amy*	여성
			Emma	여성
			Brian	남성
			Arthur	남성
10	영어(인도)	en-IN	카잘	여성
11	영어(아일랜드)	en-IE	Niamh	여성
12	영어(뉴질랜드)	en-NZ	Aria	여성
13	영어(남아프리카)	en-ZA	Ayanda	여성
14	영어(미국)	en-US	Danielle	여성

	언어 및 사투리	언어 코드	이름/ID	Gender
			Gregory	남성
			Ivy	여성(어린이)
			Joanna*	여성
			Kendra	여성
			Kimberly	여성
			Salli	여성
			Joey	남성
			Justin	남성(어린이)
			Kevin	남성(어린이)
			Matthew*	남성
			Ruth	여성
			Stephen	남성
15	핀란드어	fi-FI	Suvi	여성
16	프랑스어(벨기에)	fr-BE	Isabelle	여성
17	프랑스어(캐나다)	fr-CA	Gabrielle	여성
			Liam	남성
18	프랑스어	fr-FR	Léa	여성
			Rémi	남성
19	독일어	de-DE	Vicki	여성
			Daniel	남성

	언어 및 사투리	언어 코드	이름/ID	Gender
20	독일어(오스트리아)	de-AT	Hannah	여성
21	힌디어	hi-IN	카잘	여성
22	이탈리아어	it-IT	Bianca	여성
			Adriano	남성
23	일본어	ja-JP	Takumi	남성
			Kazuha	여성
			Tomoko	여성
24	한국어	ko-KR	Seoyeon	여성
25	노르웨이어	nb-NO	Ida	여성
26	폴란드어	pl-PL	Ola	여성
27	포르투갈어(브라질)	pt-BR	Camila	여성
			Vitória/Vitoria	여성
			Thiago	남성
28	포르투갈어(유럽)	pt-PT	Inês/Ines	여성
29	스페인어(유럽)	es-ES	Lucia	여성
			Sergio	남성
30	스페인어(멕시코)	es-MX	Mia	여성
			Andrés	남성
31	스페인어(미국)	es-US	Lupe*	여성
			Pedro	남성

	언어 및 사투리	언어 코드	이름/ID	Gender
32	스웨덴어	sv-SE	Elin	여성
33	터키어	tr-TR	버치	여성

*Amy, Joanna, Lupe 및 Matthew 음성은 뉴스 진행자 말투로 사용할 수 있습니다. 자세한 정보는 뉴스 캐스터 목소리을 참조하세요.

주제

- 기능 및 지역 호환성
- 콘솔에서 뉴럴 엔진 사용하기

기능 및 지역 호환성

뉴럴 보이스는 모든 AWS 지역에서 사용할 수 없으며 모든 Amazon Polly 기능을 지원하지도 않습니다.

신경 음성이 지원되는 리전은 다음과 같습니다.

- 미국 동부(버지니아 북부): us-east-1
- 미국 서부(오레곤): us-west-2
- 아프리카(케이프타운): af-south-1
- 아시아 태평양(도쿄): ap-northeast-1
- 아시아 태평양(서울): ap-northeast-2
- 아시아 태평양(오사카): ap-northeast-3
- 아시아 태평양(뭄바이): ap-south-1
- 아시아 태평양(싱가포르): ap-southeast-1
- 아시아 태평양(시드니): ap-southeast-2
- 캐나다(중부): ca-central-1
- 유럽(프랑크푸르트): eu-central-1
- 유럽(아일랜드): eu-west-1
- 유럽(런던): eu-west-2
- 유럽(파리): eu-west-3
- AWS GovCloud (미국 서부): -1 us-gov-west

이 리전의 엔드포인트와 프로토콜은 표준 음성에 사용되는 것과 같습니다. 자세한 내용은 Amazon Polly 엔드포인트 및 할당량을 참조하세요.

신경 음성에 지원되는 기능은 다음과 같습니다.

- 실시간 및 비동기 스피치 합성 작업
- 뉴스 진행자 말투입니다. 진행자 말투에 대한 자세한 내용은 뉴스캐스터 목소리을 참조하세요.
- 모든 스피치 마크
- Amazon Polly에서 지원하는 상당수의(전부 아님) SSML 태그입니다. NTTS 지원 SSML 태그에 대한 자세한 내용은 지원되는 태그를 참조하십시오.

표준 음성의 경우와 마찬가지로, 다양한 샘플링 속도 중에서 선택하여 애플리케이션의 대역폭과 오디오 품질을 최적화할 수 있습니다. 표준 음성과 신경 음성에 맞는 샘플링 속도는 8kHz, 16kHz, 22kHz 또는 24kHz입니다. 표준 음성의 기본값은 22kHz이고 신경 음성의 기본값은 24kHz입니다. Amazon Polly는 MP3, OGG(Vorbis) 및 원시 PCM 오디오 스트림 형식을 지원합니다.

콘솔에서 뉴럴 엔진 사용하기

Amazon Polly 콘솔 또는 을 통해 Amazon Polly Neural 음성에 액세스할 수 있습니다. AWS CLI

콘솔에서 뉴럴 엔진을 사용하려면

1. https://console.aws.amazon.com/polly/에서 Amazon Polly 콘솔을 엽니다.
2. 콘솔에서 Neural 엔진을 선택합니다.
3. 음성 드롭다운 메뉴에서 원하는 음성을 선택합니다.
4. 원하는 텍스트로 TTS 오디오를 생성하세요.

표준 음성

Amazon Polly에는 연결 합성을 사용하는 표준 엔진이 있습니다. 표준 엔진은 녹음된 음성의 음소를 연결하여 매우 자연스러운 합성 음성을 생성합니다.

사용 가능한 표준 음성

Amazon Polly는 현재 29개 언어 및 언어 변형으로 40개의 여성 및 20개의 남성 표준 음성을 제공합니다.

	언어	언어 코드	이름/ID	Gender
1	아랍어	arb	Zeina	여성
2	표준 중국어	cmn-CN	Zhiyu	여성
3	덴마크어	da-DK	Naja	여성
			Mads	남성
4	네덜란드어	nl-NL	Lotte	여성
			Ruben	남성
5	영어(호주)	en-AU	Nicole	여성
			Russell	남성
6	영어(영국)	en-GB	Amy	여성
			Emma	여성
			Brian	남성
7	영어(인도)	en-IN	Aditi	여성
			Raveena	여성
8	영어(미국)	en-US	Ivy	여성
			Joanna	여성
			Kendra	여성
			Kimberly	여성
			Salli	여성
			Joey	남성
			Kevin	남성

	언어	언어 코드	이름/ID	Gender
9	영어(웨일스)	en-GB-WLS	Geraint	남성
10	프랑스어	fr-FR	Céline/Celine	여성
			Léa	여성
			Mathieu	남성
11	프랑스어(캐나다)	fr-CA	Chantal	여성
12	독일어	de-DE	Marlene	여성
			Vicki	여성
			Hans	남성
13	힌디어	hi-IN	아디티	여성
14	아이슬란드어	is-IS	Dóra/Dora	여성
			Karl	남성
15	이탈리아어	it-IT	Carla	여성
			Bianca	여성
			Giorgio	남성
16	일본어	ja-JP	Mizuki	여성
			Takumi	남성
17	한국어	ko-KR	Seoyeon	여성
18	노르웨이어	nb-NO	Liv	여성

	언어	언어 코드	이름/ID	Gender
19	폴란드어	pl-PL	Ewa	여성
			Maja	여성
			Jacek	남성
			Jan	남성
20	포르투갈어(브라질)	pt-BR	Camila	여성
			Vitória/Vitoria	여성
			Ricardo	남성
21	포르투갈어(유럽)	pt-PT	Inês/Ines	여성
			Cristiano	남성
22	루마니아어	ro-RO	Carmen	여성
23	러시아어	ru-RU	Tatyana	여성
			Maxim	남성
24	스페인어(유럽)	es-ES	Conchita	여성
			Lucia	여성
			Enrique	남성
25	스페인어(멕시코)	es-MX	Mia	여성
26	스페인어(미국)	es-US	Lupe	여성
			Penélope/Penelope	여성
			Miguel	남성
27	스웨덴어	sv-SE	Astrid	여성

	언어	언어 코드	이름/ID	Gender
28	터키어	tr-TR	Filiz	남성
29	웨일스어	cy-GB	Gwyneth	여성

기능 및 지역 호환성

Amazon Polly 표준 음성은 22개 아마존 폴리 지역 모두에서 사용할 수 있습니다.

- 미국 동부(버지니아 북부): us-east-1
- 미국 서부(오레곤): us-west-2
- 아프리카(케이프타운): af-south-1
- 아시아 태평양(도쿄): ap-northeast-1
- 아시아 태평양(서울): ap-northeast-2
- 아시아 태평양(오사카): ap-northeast-3
- 아시아 태평양(뭄바이): ap-south-1
- 중국(닝샤) 리전
- 아시아 태평양(싱가포르): ap-southeast-1
- 아시아 태평양(시드니): ap-southeast-2
- 캐나다(중부): ca-central-1
- 유럽(프랑크푸르트): eu-central-1
- 유럽(아일랜드): eu-west-1
- 유럽(런던): eu-west-2
- 유럽(파리): eu-west-3
- AWS GovCloud (미국 서부): -1 us-gov-west

이 지역의 엔드포인트 및 프로토콜은 뉴럴 음성에 사용되는 것과 동일합니다. 자세한 내용은 Amazon Polly 엔드포인트 및 할당량을 참조하세요.

Amazon Polly 표준 엔진은 다음 기능 (TBD) 을 지원합니다.

- 실시간 및 비동기 스피치 합성 작업
- 모든 스피치 마크

- 전부는 아니지만 많은 SSML 태그가 Amazon Polly에서 지원됩니다. NTTS가 지원하는 SSML 태그에 대한 자세한 내용은 지원되는 SSML 태그를 참조하십시오.
- 100밀리초의 지연 시간
- 다양한 샘플링 속도 중에서 선택하여 애플리케이션의 대역폭과 오디오 품질을 최적화할 수 있습니다. 표준 음성의 기본 샘플링 속도는 22kHz입니다. Amazon Polly는 MP3, OGG(Vorbis) 및 원시 PCM 오디오 스트림 형식을 지원합니다.

> **Note**
>
> 표준 음성 비용은 Amazon Polly 요금 정보 페이지에 명시되어 있습니다.

콘솔에서 스탠다드 보이스 사용하기

Amazon Polly 콘솔 또는 을 통해 Amazon Polly 표준 음성에 액세스할 수 있습니다. AWS CLI

콘솔에서 표준 음성을 사용하려면

1. https://console.aws.amazon.com/polly/에서 Amazon Polly 콘솔을 엽니다.
2. Amazon Polly 콘솔에서 표준 엔진을 선택합니다.
3. 음성 드롭다운 메뉴에서 원하는 음성을 선택합니다.
4. 원하는 텍스트로 TTS 오디오를 생성하세요.

> **Note**
>
> 표준 음성은 **SynthesizeSpeech**및 **StartSpeechSynthesisTask**API 작업과 함께 사용할 수도 있습니다. API 작업의 경우 고객은 API 요청에서 엔진 및 음성 이름을 지정할 수 있습니다. 퀵 스타트 코드 샘플을 더 많이 찾을 수 있습니다.

스피치 마크

스피치 마크는 문장이나 단어가 오디오 스트림에서 시작과 끝의 위치에서 합성하는 스피치를 설명하는 메타데이터입니다. 텍스트에 스피치 마크를 요청하면 Amazon Polly에서 합성된 스피치 대신 이 메타데이터를 반환합니다. 합성 스피치 오디오 스트림과 함께 스피치 마크를 사용하면 애플리케이션에 향상된 시각적 경험을 제공할 수 있습니다.

예를 들어, 메타데이터를 텍스트의 오디오 스트림과 결합하면 스피치를 얼굴 애니메이션(립싱크)과 동기화하거나 쓰여진 단어를 강조해서 표시할 수 있습니다.

음성 표시는 신경망 형식 또는 표준 text-to-speech 형식을 사용할 때 사용할 수 있습니다.

주제

- 스피치 마크 유형
- 스피치 부호 사용
- 콘솔에서 스피치 마크 요청하기

스피치 마크 유형

SynthesizeSpeech또는 StartSpeechSynthesisTask명령의 SpeechMarkTypes옵션을 사용하여 스피치 마크를 요청합니다. 입력 텍스트에서 반환할 메타데이터 요소를 지정합니다. 최대 네 가지 형식의 메타데이터를 요청할 수 있지만, 요청당 반드시 하나 이상을 지정해야 합니다. 요청 시 오디오 출력이 생성되지 않습니다.

예를 AWS CLI들면 다음과 같습니다.

```
--speech-mark-types='["sentence", "word", "viseme", "ssml"]'
```

Amazon Polly는 다음 요소를 사용하여 스피치 마크를 생성합니다.

- sentence – 입력 텍스트의 문장 요소를 나타냅니다.
- word – 텍스트의 단어 요소를 나타냅니다.
- viseme – 말하는 각 음소에 대응하는 얼굴과 입의 움직임을 설명합니다. 자세한 내용은 Viseme과 Amazon Polly을(를) 참조하세요.
- ssml – SSML 입력 텍스트의 <mark> 요소를 설명합니다. 자세한 내용은 SSML 문서에서 음성 생성을(를) 참조하세요.

Viseme과 Amazon Polly

viseme은 단어를 말할 때 얼굴과 입의 위치를 나타냅니다. 이것은 단어가 구성되는 기본적인 음향 단위인 음소에 대한 시각적인 효과와 동등한 것을 나타냅니다. Viseme은 스피치의 기본 시각적 빌딩 블록입니다.

각 언어에는 특정 음소에 해당하는 비짐 세트가 있습니다. 언어에서 각 음소는 소리를 형성할 때 입이 만드는 모양을 나타내는 해당 viseme을 가지고 있습니다. 그러나 소리가 달라도 수많은 음소가 동일하게 나타나므로 모든 viseme을 특정 음소에 매핑할 수 있는 것은 아닙니다. 예를 들어 영어에서는 "pet"과 "bet"이라는 단어가 음향적으로 다릅니다. 그러나 소리없이 시각적으로 관찰했을 때 두 단어는 동일한 것처럼 보입니다.

다음 차트에서는 미국 영어 음성에 대한 IPA(International Phonetic Alphabet) 음소와 X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet) 기호 및 그에 해당하는 viseme의 일부 목록을 보여줍니다.

사용 가능한 모든 언어에 관한 전체적인 표는 지원되는 언어의 음소 및 Viseme 표을(를) 참조하세요.

IPA	X-SAMPA	설명	예	Viseme
자음				
b	b	유성 양순 파열음	bed	p
d	d	유성 치경 파열음	dig	t
ʧ	dZ	유성 후치경 파찰음	jump	S
ð	D	유성 치 마찰음	then	T
f	f	무성 순치 마찰음	five	f
g	g	유성 연구개 파열음	game	k
h	h	무성 성문 마찰음	house	k
...	...	...	...	...

스피치 부호 사용

스피치 마크 요청

입력 텍스트의 스피치 마크를 요청하려면 synthesize-speech 명령을 사용하세요. 입력 텍스트 외에 이 메타데이터를 반환하려면 다음 요소가 필요합니다.

- output-format

  Amazon Polly에서는 스피치 마크를 반환할 때 JSON 형식만 지원합니다.

  ```
  --output-format json
  ```

  지원되지 않는 출력 형식을 사용하면 Amazon Polly에서 예외가 발생합니다.

- voice-id

  메타데이터가 관련 오디오 스트림과 일치하는지 확인하려면 합성 스피치 오디오 스트림을 생성하는 데 사용되는 것과 동일한 음성을 지정하세요. 사용 가능한 음성은 동일한 스피치 속도를 아닙니다. 스피치를 생성하는 데 사용된 것과 다른 음성을 사용하는 경우 메타데이터가 오디오 스트림과 일치하지 않습니다.

  ```
  --voice-id Joanna
  ```

- speech-mark-types

  원하는 스피치 마크 유형을 지정합니다. 스피치 마크 형식의 일부 또는 전부를 요청할 수 있지만 적어도 하나의 형식을 지정해야 합니다.

  ```
  --speech-mark-types='["sentence", "word", "viseme", "ssml"]'
  ```

- text-type

  일반 텍스트는 Amazon Polly의 기본 입력 텍스트이므로 SSML 스피치 마크를 반환하려면 text-type ssml을 사용해야 합니다.

- outfile

  메타데이터가 기록되는 출력 파일을 지정합니다.

  ```
  MaryLamb.txt
  ```

다음 AWS CLI 예제는 유닉스, 리눅스, macOS용으로 포맷되었습니다. Windows의 경우 각 줄 끝에 있는 백슬래시(\) Unix 연속 문자를 캐럿(^)으로 바꿉니다. 입력 텍스트는 큰 따옴표(")로 감싸고 내부 태그에는 작은 따옴표(')를 사용합니다.

```
aws polly synthesize-speech \
  --output-format json \
  --voice-id Voice ID \
  --text 'Input text' \
  --speech-mark-types='["sentence", "word", "viseme"]' \
  outfile
```

스피치 마크 출력

Amazon Polly에서는 줄로 구분된 JSON 스트림에서 스피치 마크 객체를 반환합니다. 스피치 마크 객체는 다음과 같은 필드를 포함합니다.

- time – 해당하는 오디오 스트림 시작 부분의 타임스탬프(밀리초)
- type – 스피치 마크의 형식(sentence, word, viseme, ssml)
- start – 입력 텍스트에서 객체 시작의 오프셋(바이트)(viseme 마크 포함하지 않음)
- end – 입력 텍스트에서 객체 끝의 오프셋(바이트)(viseme 마크 포함하지 않음)
- value – 스피치 마크 형식에 따라 다양
  - SSML: <mark> SSML 태그
  - viseme: viseme 이름
  - word 또는 sentence: 시작 및 끝 필드로 구분된 입력 텍스트의 하위 문자열

예를 들어, Amazon Polly에 "Mary have little lamb" 텍스트를 제공하면 다음 `word` 스피치 마크 객체를 생성합니다.

```
{"time":373,"type":"word","start":5,"end":8,"value":"had"}
```

설명된 단어 "had"는 오디오 스트림이 시작된 후 373밀리초부터 시작하여 입력 텍스트의 5바이트에서 시작하여 8바이트에서 끝납니다.

> **Note**
>
> 이 메타데이터는Joanna 음성 ID용입니다. 동일한 입력 텍스트로 다른 음성을 사용하는 경우 메타데이터가 다를 수 있습니다.

스피치 마크 예제

다음과 같은 스피치 마크 요청의 예제는 일반적인 요청을 만드는 방법과 요청이 생성하는 출력을 보여줍니다.

예제 1 : SSML이 없는 스피치 마크

다음 예제는 "Mary have little lamb."이라는 간단한 문장에 대해 화면에서 요청된 메타데이터가 어떻게 표시되는지 보여줍니다. 이 예제에서는 간소화를 위해 SSML 스피치 마크를 포함하지 않습니다.

다음 AWS CLI 예제는 유닉스, 리눅스, macOS용으로 포맷되었습니다. Windows의 경우 각 줄 끝에 있는 백슬래시(\) Unix 연속 문자를 캐럿(^)으로 바꿉니다. 입력 텍스트는 큰 따옴표(")로 감싸고 내부 태그에는 작은 따옴표(')를 사용합니다.

```
aws polly synthesize-speech \
  --output-format json \
  --voice-id Joanna \
  --text 'Mary had a little lamb.' \
  --speech-mark-types='["viseme", "word", "sentence"]' \
  MaryLamb.txt
```

이 요청을 하면 Amazon Polly에서 .txt 파일에 다음을 반환합니다.

```
{"time":0,"type":"sentence","start":0,"end":23,"value":"Mary had a little lamb."}
{"time":6,"type":"word","start":0,"end":4,"value":"Mary"}
{"time":6,"type":"viseme","value":"p"}
{"time":73,"type":"viseme","value":"E"}
{"time":180,"type":"viseme","value":"r"}
{"time":292,"type":"viseme","value":"i"}
{"time":373,"type":"word","start":5,"end":8,"value":"had"}
{"time":373,"type":"viseme","value":"k"}
{"time":460,"type":"viseme","value":"a"}
{"time":521,"type":"viseme","value":"t"}
```

```
{"time":604,"type":"word","start":9,"end":10,"value":"a"}
{"time":604,"type":"viseme","value":"@"}
{"time":643,"type":"word","start":11,"end":17,"value":"little"}
{"time":643,"type":"viseme","value":"t"}
{"time":739,"type":"viseme","value":"i"}
{"time":769,"type":"viseme","value":"t"}
{"time":799,"type":"viseme","value":"t"}
{"time":882,"type":"word","start":18,"end":22,"value":"lamb"}
{"time":882,"type":"viseme","value":"t"}
{"time":964,"type":"viseme","value":"a"}
{"time":1082,"type":"viseme","value":"p"}
```

이 출력에서 텍스트의 각 부분은 스피치 마크 용어로 구분됩니다.

- sentence는 "Mary had a little lamb."입니다.
- 텍스트의 각 word는 "Mary", "had", "a", "little", and "lamb."입니다.
- 해당하는 오디오 스트림에서 각 사운드에 대한 viseme는 "p", "E", "r", "i" 등입니다. viseme에 대한 자세한 내용은 Viseme과 Amazon Polly을(를) 참조하세요.

예 2: SSML을 사용한 스피치 마크

SSML로 확장된 텍스트에서 스피치 마크를 생성하는 과정은 SSML이 없는 경우와 유사합니다. `synthesize-speech` 명령을 사용하여 다음 예제와 같이 원하는 SSML 확장 텍스트와 스피치 마크 형식을 지정하세요. 예제를 더 쉽게 읽을 수 있도록 viseme 스피치 마크는 포함하지 않지만 이러한 스피치 마크도 포함할 수 있습니다.

다음 AWS CLI 예제는 유닉스, 리눅스, macOS용으로 포맷되었습니다. Windows의 경우 각 줄 끝에 있는 백슬래시(\) Unix 연속 문자를 캐럿(^)으로 바꿉니다. 입력 텍스트는 큰 따옴표(")로 감싸고 내부 태그에는 작은 따옴표(')를 사용합니다.

```
aws polly synthesize-speech \
  --output-format json \
  --voice-id Joanna \
  --text-type ssml \
  --text '<speak><prosody volume="+20dB">Mary had <break time="300ms"/>a little <mark
 name="animal"/>lamb</prosody></speak>' \
  --speech-mark-types='["sentence", "word", "ssml"]' \
  output.txt
```

이 요청을 하면 Amazon Polly에서 .txt 파일에 다음을 반환합니다.

```
{"time":0,"type":"sentence","start":31,"end":95,"value":"Mary had <break time=\"300ms
\"\/>a little <mark name=\"animal\"\/>lamb"}
{"time":6,"type":"word","start":31,"end":35,"value":"Mary"}
{"time":325,"type":"word","start":36,"end":39,"value":"had"}
{"time":897,"type":"word","start":40,"end":61,"value":"<break time=\"300ms\"\/>"}
{"time":1291,"type":"word","start":61,"end":62,"value":"a"}
{"time":1373,"type":"word","start":63,"end":69,"value":"little"}
{"time":1635,"type":"ssml","start":70,"end":91,"value":"animal"}
{"time":1635,"type":"word","start":91,"end":95,"value":"lamb"}
```

콘솔에서 스피치 마크 요청하기

콘솔을 사용하여 Amazon Polly의 스피치 마크를 요청할 수 있습니다. 그런 다음 메타데이터를 보거나 파일에 저장할 수 있습니다.

스피치 마크를 생성하려면(콘솔)

1. AWS Management Console 로그인하고 https://console.aws.amazon.com/polly/ 에서 Amazon Polly 콘솔을 엽니다.
2. 텍스트 투 스피치 탭을 선택합니다.
3. SSML을 사용하기 위해 SSML을 활성화합니다.
4. 텍스트를 입력란에 입력하거나 붙여 넣습니다.
5. 언어 및 리전에서 텍스트 언어를 선택합니다.
6. 음성에서 텍스트에 사용하려는 음성을 선택합니다.
7. 텍스트 발음을 변경하려면 추가 설정을 확장하고 발음 사용자 지정을 켠 다음 어휘 적용에서 원하는 어휘를 선택합니다.
8. 듣기를 선택하여 스피치가 최종 형태인지 확인합니다.
9. 스피치 파일 형식 설정을 활성화합니다.

   > **Note**
   >
   > MP3, OGG, PCM 형식을 다운로드하면 스피치 마크가 생성되지 않습니다.

10. 파일 형식에서 스피치 마크를 선택합니다.
11. 스피치 마크 유형에서 생성할 스피치 마크 유형을 선택합니다. SSML 메타데이터를 선택하는 옵션은 SSML이 활성화된 상태에서만 사용할 수 있습니다. Amazon Polly에서 SSML을 사용하는 방법에 대한 자세한 내용은 SSML 문서에서 음성 생성을 참조하세요.

12. 다운로드를 선택합니다.

SSML 문서에서 음성 생성

Amazon Polly를 사용하여 일반 텍스트 또는 SSML(Speech Synthesis Markup Language)로 표시된 문서로부터 스피치를 생성할 수 있습니다. SSML로 확장된 텍스트를 사용하면 Amazon Polly가 제공된 텍스트에서 스피치를 생성하는 방식을 추가로 제어할 수 있습니다.

예를 들어, 텍스트에 일시 중지를 길게 포함하거나 스피치 속도 또는 피치를 변경할 수 있습니다. 그 밖에도 다음과 같은 옵션이 있습니다.

- 특정 단어나 어구 강조
- 철자대로의 발음 사용
- 숨소리 포함
- 속삭임 기능
- 뉴스 진행자 말투 사용

Amazon Polly에서 지원되는 SSML 태그에 대한 자세한 내용과 태그 사용 방법은 지원되는 SSML 태그를 참조하세요.

SSML을 사용할 때 특수한 처리가 필요한 예약 문자가 여러 개 있습니다. SSML에서 코드의 일부로 이 문자를 사용하기 때문입니다. 이 문자를 사용하려면 특정 개체를 사용하여 이스케이프합니다. 자세한 정보는 SSML의 예약된 문자을(를) 참조하세요.

Amazon Polly는 Speech Synthesis Markup Language (SSML) Version 1.1, W3C Recommendation에 정의된 SSML 마크업 태그의 하위 집합과 함께 이러한 유형의 제어 기능을 제공합니다.

SSML을 Amazon Polly 콘솔 내에서 사용하거나 AWS CLI를 통해 사용할 수 있습니다. 다음 주제는 사용자가 SSML을 사용하여 스피치를 생성하고 본인의 필요에 정확히 부합하도록 출력을 제어하는 방법을 보여줍니다.

주제

- SSML의 예약된 문자
- 콘솔에서 SSML 사용
- SSML을 사용하여 AWS CLI
- 지원되는 SSML 태그

SSML의 예약된 문자

일반적으로 SSML 문에서 사용할 수 없는 문자 5가지가 미리 정의되어 있습니다. 이 개체는 언어 사양에 의해 예약되며 다음과 같은 문자입니다.

명칭	문자
장	케이프코드
따옴표(큰 따옴표)	&qubt;
앰퍼샌드	&
아포스트로피 또는 작은 따	'

명칭	문자	이스케이프 코드
앰퍼샌드	&	&
작음 기호	<	<
큼 기호	>	>

이 문자가 SSML에서 코드의 일부로 사용되므로 SSML에서 이 기호를 사용하려면 이 문자를 사용할 때 이스케이프해야 합니다. 실제 문자 대신 이스케이프 코드를 사용하면 문자가 제대로 표시되는 유효한 SSML 문서를 만들 수 있습니다. 예를 들어 다음 문장은

```
We're using the lawyer at Peabody & Chambers, attorneys-at-law.
```

SSML에서 아래와 같이 렌더링됩니다.

```
<speak>
We&apos;re using the lawyer at Peabody &amp; Chambers, attorneys-at-law.
</speak>
```

이 경우 아포스트로피와 앰퍼샌드의 특수 문자가 이스케이프되어 SSML 문서가 계속 유효합니다.

&, < 및 > 기호는 SSML을 사용할 때 항상 이스케이프 코드가 필요합니다. 뿐만 아니라 아포스트로피/작은따옴표(')를 아포스트로피로 사용할 때도 이스케이프 코드를 사용해야 합니다.

하지만 큰따옴표(")나 아포스트로피/작은따옴표(')를 인용 부호로 사용할 때는 이스케이프 코드 사용 여부가 상황에 따라 달라집니다.

큰따옴표

- 큰따옴표로 구분되는 속성 값에서는 이스케이프해야 합니다. 예를 들어, 다음 AWS CLI 코드에서는

```
--text "Pete &quot;Maverick&quot; Mitchell"
```

- 텍스트 컨텍스트에서는 이스케이프할 필요가 없습니다. 예를 들어 다음과 같습니다.

```
He said, "Turn right at the corner."
```

- 작은따옴표로 구분되는 속성 값에서는 이스케이프할 필요가 없습니다. 예를 들어, 다음 AWS CLI 코드에서는

```
--text 'Pete "Maverick" Mitchell'
```

작은따옴표

- 아포스트로피로 사용할 때 이스케이프해야 합니다. 예를 들어 다음과 같습니다.

```
We&apos;ve got to leave quickly.
```

- 텍스트 컨텍스트에서는 이스케이프할 필요가 없습니다. 예를 들어 다음과 같습니다.

```
"And then I said, 'Don't quote me.'"
```

- 큰따옴표로 구분되는 코드 속성에서는 이스케이프할 필요가 없습니다. 예를 들어, 다음 AWS CLI 코드에서는

```
--text "Pete 'Maverick' Mitchell"
```

콘솔에서 SSML 사용

SSML 태그를 사용하면 스피치를 사용자 지정하고 발음, 볼륨 및 스피치 속도와 같은 음성 측면을 제어할 수 있습니다. 에서 오디오로 변환하려는 SSML 강화 텍스트는 텍스트 음성 변환 페이지의 SSML 탭에 입력됩니다. AWS Management Console일반 텍스트로 입력된 텍스트는 선택한 언어 및 음성에 대해 기본 설정을 사용하지만, SSML로 확장된 텍스트는 Amazon Polly에 말하고자 하는 내용뿐만 아니라 말하고자 하는 방법도 지시합니다. 추가된 SSML 태그를 제외하면, Amazon Polly는 일반 텍스트를 합성할 때와 동일한 방식을 통해 SSML로 확장된 텍스트를 합성합니다. 자세한 정보는 1.2단계: 콘솔의 일반 텍스트 입력으로 음성을 합성합니다.을(를) 참조하세요.

SSML을 사용할 때 SSML을 사용하고 있음을 Amazon Polly가 알 수 있도록 전체 텍스트를 <speak> 태그로 묶습니다. 예:

```
<speak>Hi! My name is Joanna. I will read any text you type here.</speak>
```

그런 다음 <speak> 태그 내부의 텍스트에서 특정 SSML 태그를 사용하여 텍스트를 소리 낼 방법을 사용자 지정할 수 있습니다. 일시 중지를 추가하거나, 스피치 속도를 변경하거나, 음성 볼륨을 낮추거나 높이거나, 텍스트 소리가 내게 적합하도록 다른 많은 사용자 지정을 추가할 수 있습니다. 사용할 수 있는 SSML 태그의 전체 목록은 지원되는 SSML 태그를 참조하세요.

다음 예에서는 SSML 태그를 사용하여 Amazon Polly에서 짧은 단락을 말할 때 "W3C"를 "월드 와이드 웹 컨소시엄"으로 대체하도록 지시합니다. 또한 태그를 사용하여 일시 중지를 도입하고 단어를 속삭입니다. 이 연습의 결과를 콘솔에서 어휘 적용 (음성 합성)의 결과와 비교합니다.

예에 사용된 SSML에 대한 자세한 내용은 지원되는 SSML 태그를 참조하세요.

SSML로 확장된 텍스트(콘솔)로부터 스피치를 합성하려면

1. AWS Management Console 로그인하고 https://console.aws.amazon.com/polly/ 에서 Amazon Polly 콘솔을 엽니다.
2. 아직 표시되지 않은 경우 텍스트 투 스피치 탭을 선택합니다.
3. SSML을 활성화합니다.
4. 다음 텍스트를 텍스트 상자에 입력하거나 붙여 넣습니다.

```
<speak>
     He was caught up in the game.<break time="1s"/> In the middle of the
     10/3/2014 <sub alias="World Wide Web Consortium">W3C</sub> meeting,
```

```
    he shouted, "Nice job!" quite loudly. When his boss stared at him, he
 repeated
    <amazon:effect name="whispered">"Nice job,"</amazon:effect> in a
    whisper.
</speak>
```

SSML 태그가 Amazon Polly에 텍스트를 렌더링하는 방법을 알려줍니다.

- `<break time="1s"/>`는 처음 두 문장 사이에서 1초 동안 일시 중지하도록 Amazon Polly에 지시합니다.
- `<sub alias="World Wide Web Consortium">W3C</sub>`는 두문자어 W3C를 월드 와이드 웹 컨소시엄으로 대체하도록 Amazon Polly에 지시합니다.
- `<amazon:effect name="whispered">Nice job</amazon:effect>`은 속삭이는 음성으로 "Nice job"의 두 번째 인스턴스를 말하도록 Amazon Polly에 지시합니다.

> **Note**
>
> 를 사용할 때는 입력 텍스트를 따옴표로 묶어 주변 코드와 구분합니다. AWS CLI Amazon Polly 콘솔은 사용자에게 코드를 보여주지 않으므로 이 콘솔을 사용할 때에는 입력 텍스트를 따옴표로 묶지 않습니다.

5. 언어의 경우 영어, 미국을 선택한 다음 음성을 선택합니다.
6. 스피치를 들으려면 듣기를 선택합니다.
7. 스피치 파일을 저장하려면 다운로드를 선택합니다. 다른 형식으로 저장하려면 추가 설정을 확장하고, 스피치 파일 형식 설정을 활성화하고, 원하는 형식을 선택한 다음 다운로드를 선택합니다.

SSML을 사용하여 AWS CLI

를 사용하여 SSML 입력 텍스트를 AWS CLI 합성할 수 있습니다. 다음 예제는 AWS CLI를 사용하여 일반적인 작업을 수행하는 방법을 보여줍니다.

주제

- SSML을 음성 합성 명령과 함께 사용
- SSML로 개선된 문서 합성하기
- 일반적인 Amazon Polly 작업에 SSML 사용

SSML을 음성 합성 명령과 함께 사용

이 예에서는 synthesize-speech 명령을 SSML 문자열과 함께 사용하는 방법을 보여줍니다. synthesize-speech 명령을 사용하는 경우 사용자는 일반적으로 다음 사항을 제공합니다.

- 입력 텍스트(필수)
- 태그 열기 및 닫기(필수)
- 출력 형식
- 음성

이 예에서는 필요한 열기 및 닫기 <speak></speak> 태그와 함께 따옴표로 묶여 있는 간단한 문자열을 지정합니다.

Important

Amazon Polly 콘솔에서 입력 텍스트 주위에 따옴표를 사용하지 않더라도 AWS CLI 를 사용하는 경우 반드시 따옴표를 사용해야 합니다. 입력 텍스트를 둘러싸는 따옴표와 각 태그에 필요한 따옴표를 구별하는 것 또한 중요합니다.
예를 들어, 입력 텍스트 주위에는 표준 따옴표(")를 사용하고 내부 태그에는 작은 따옴표(')를 사용할 수 있으며, 그 반대로도 사용할 수 있습니다. 두 옵션 모두 Unix, Linux 및 macOS에 사용됩니다. 그러나 Windows에서는 입력 텍스트 주위에 표준 따옴표(")를 사용하고 태그에는 작은 따옴표(')를 사용해야 합니다.
모든 운영 체제에서, 입력 텍스트 주위에는 표준 따옴표(")를 사용하고 내부 태그에는 작은 따옴표(')를 사용할 수 있습니다. 예:

```
--text "<speak>Hello <break time='300ms'/> World</speak>"
```

Unix, Linux 및 macOS에서는 그 반대로 입력 텍스트 주위에 작은 따옴표(')를 사용하고 내부 태그에는 표준 따옴표(")를 사용할 수도 있습니다.

```
--text '<speak>Hello <break time="300ms"/> World</speak>'
```

다음 AWS CLI 예제는 유닉스, 리눅스, macOS용으로 포맷되었습니다. Windows의 경우 각 줄 끝에 있는 백슬래시(\) Unix 연속 문자를 캐럿(^)으로 바꿉니다. 입력 텍스트는 큰 따옴표(")로 감싸고 내부 태그에는 작은 따옴표(')를 사용합니다.

```
aws polly synthesize-speech \
--text-type ssml \
--text '<speak>Hello world</speak>' \
--output-format mp3 \
--voice-id Joanna \
speech.mp3
```

합성된 스피치를 들으려면 어떤 종류로든 오디오 플레이어를 사용하여 speech.mp3 결과 파일을 재생합니다.

SSML로 개선된 문서 합성하기

비교적 긴 입력 텍스트에 대해서는 SSML 콘텐츠를 파일에 저장하고 synthesize-speech 명령에 파일 이름을 지정하는 단순한 방법이 편하게 느껴질 수 있습니다. 예를 들어 다음 사항을 파일 이름이 example.xml인 파일에 저장할 수 있습니다.

```
<?xml version="1.0"?>
<speak version="1.1"
       xmlns="http://www.w3.org/2001/10/synthesis"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.w3.org/2001/10/synthesis http://www.w3.org/TR/
speech-synthesis11/synthesis.xsd"
       xml:lang="en-US">Hello World</speak>
```

xml:lang 속성은 en-US(미국 영어)를 입력 텍스트의 언어로 지정합니다. 입력 텍스트의 언어와 선택한 음성의 언어가 SynthesizeSpeech 작업에 미치는 영향에 대한 자세한 내용은 외국어 발음 개선하기을 참조하세요.

SSML로 확장된 파일을 실행하려면

1. SSML을 파일(예: example.xml)에 저장합니다.
2. XML 파일이 저장된 경로에서 다음 synthesize-speech 명령을 실행하고 입력 데이터를 file:\\example.xml로 대체하여 SSML 파일을 입력으로 지정합니다. 이 명령은 실제의 입력 텍스트를 포함하는 대신 파일을 지정하므로 따옴표를 사용하지 않습니다.

> **Note**
>
> 다음 AWS CLI 예제는 유닉스, 리눅스, macOS용으로 포맷되었습니다. Windows의 경우 각 줄의 끝에 있는 백슬래시(\) Unix 연속 문자를 캐럿(^)으로 바꿉니다.

```
aws polly synthesize-speech \
--text-type ssml \
--text file://example.xml \
--output-format mp3 \
--voice-id Joanna \
speech.mp3
```

3. 합성된 스피치를 들으려면 어떤 종류로든 오디오 플레이어를 사용하여 speech.mp3 결과 파일을 재생합니다.

일반적인 Amazon Polly 작업에 SSML 사용

다음 예에서는 SSML 태그를 사용하여 일반적인 Amazon Polly 작업을 완료하는 방법을 보여줍니다. SSML 태그에 대한 자세한 내용은 지원되는 SSML 태그를 참조하세요.

다음 예를 테스트하려면 SSML로 확장된 적절한 텍스트로 다음 synthesize-speech 명령을 사용합니다.

다음 AWS CLI 예제는 유닉스, 리눅스, macOS용으로 포맷되었습니다. Windows의 경우 각 줄 끝에 있는 백슬래시(\) Unix 연속 문자를 캐럿(^)으로 바꿉니다. 입력 텍스트는 큰 따옴표(")로 감싸고 내부 태그에는 작은 따옴표(')를 사용합니다.

```
aws polly synthesize-speech \
--text-type ssml \
--text '<speak>Hello <break time="300ms"/> World</speak>' \
--output-format mp3 \
--voice-id Joanna \
speech.mp3
```

일시 중지 추가

단어 사이에 일시 중지를 추가하려면 <break> 요소를 사용합니다. 다음 SSML synthesize-speech 명령은 <break> 요소를 사용하여 "Hello"와 "World" 단어 사이에 300밀리초 지연을 추가합니다.

```
<speak>
     Hello <break time="300ms"/> World.
</speak>
```

음량, 피치, 속도 제어

피치, 말하기 속도 및 스피치 볼륨을 제어하려면 <prosody> 요소를 사용합니다.

- 다음 synthesize-speech 명령은 <prosody> 요소를 사용하여 볼륨을 제어합니다.

```
<speak>
     <prosody volume="+20dB">Hello world</prosody>
</speak>
```

- 다음 synthesize-speech 명령은 <prosody> 요소를 사용하여 피치를 제어합니다.

```
<speak>
     <prosody pitch="x-high">Hello world.</prosody>
</speak>
```

- 다음 synthesize-speech 명령은 <prosody> 요소를 사용하여 스피치 속도(말하기 속도)를 지정합니다.

```
<speak>
     <prosody rate="x-fast">Hello world.</prosody>
</speak>
```

- 다음 예와 같이 <prosody> 요소에서 속성을 여러 개 지정할 수 있습니다.

```
<speak>
     <prosody volume="x-loud" pitch="x-high" rate="x-fast">Hello world.</prosody>
</speak>
```

속삭임 기능

단어를 속삭이려면 <amazon:effect name="whispered"> 요소를 사용합니다. 다음 예에서는 <amazon:effect name="whispered"> 요소가 Amazon Polly에 "little lamb"을 속삭이도록 지시합니다.

```
<speak>
     Mary has a <amazon:effect name="whispered">little lamb.</amazon:effect>
</speak>
```

이 효과를 확장시키려면 <prosody> 요소를 사용하여 속삭이는 스피치의 속도를 약간 늦춥니다.

단어 강조

단어 또는 구절을 강조하려면 <emphasis> 요소를 사용합니다.

```
<speak>
     <emphasis level="strong">Hello</emphasis> world how are you?
</speak>
```

특정 단어를 말하는 방법 지정하기

말할 텍스트의 유형에 대한 정보를 제공하려면 <say-as> 요소를 사용합니다.

예를 들어 다음 SSML에서는 `<say-as>`가 텍스트 4/6이 날짜로 해석되어야 함을 나타냅니다. 속성 `interpret-as="date" format="dm"`는 텍스트 4/6이 월/일 형식의 날짜로 말해져야 함을 나타냅니다.

또한 <say-as> 요소를 사용하여 Amazon Polly에 숫자를 분수, 전화번호, 측정 단위 등으로 말하도록 지시할 수도 있습니다.

```
<speak>
     Today is <say-as interpret-as="date" format="md" >4/6</say-as>
</speak>
```

결과 스피치는 "Today is June 4th"입니다. `<say-as>` 태그는 `interpret-as` 속성을 통해 추가 컨텍스트를 제공하여 텍스트가 해석되어질 방식을 설명합니다.

합성된 스피치의 정확성을 확인하려면 `speech.mp3` 결과 파일을 재생합니다.

이 요소에 대한 자세한 내용은 특별한 유형의 단어를 말하는 방식 제어 을(를) 참조하세요.

외국어 발음 개선하기

Amazon Polly는 입력 텍스트가 사용자가 선택한 음성으로 말해지는 언어와 동일한 언어라고 가정합니다. 입력 텍스트 내의 외국어 단어의 발음을 개선하려면 `synthesize-speech` 호출에서 `xml:lang` 속성을 통해 대상 언어를 지정합니다. 이렇게 하면 Amazon Polly에 사용자가 태그한 외국어 단어에 다른 발음 규칙을 적용하도록 지시합니다.

다음 예에서는 입력 텍스트에서 다양한 언어 조합을 사용하는 방법과 외국어 단어에 음성 및 발음을 지정하는 방법을 보여줍니다. 사용 가능한 언어에 대한 전체 목록은 Amazon Polly의 언어을(를) 참조하세요.

다음 예에서 선택된 음성(Joanna)은 미국 영어 음성입니다. 기본적으로 Amazon Polly는 입력 텍스트가 해당 음성(이 경우 미국 영어)과 동일한 언어라고 가정합니다. `xml:lang` 태그를 사용하면 Amazon Polly가 텍스트를 스페인어로 해석하여 해당 텍스트를 해당 외국어의 발음 규칙에 따라 선택된 음성이 스페인어 단어를 발음하는 방식으로 읽습니다. 이 태그가 없으면 텍스트를 선택된 음성의 발음 규칙을 사용해 읽습니다.

```
<speak>
     That restaurant is terrific. <lang xml:lang="es-ES">Mucho gusto.</lang>
</speak>
```

입력 텍스트의 언어가 영어이므로 Amazon Polly는 스페인어 음소를 음향 거리가 가장 가까운 영어 음소로 매핑합니다. 결과적으로, Joanna는 스페인어로 정확히 발음할 줄 아는 미국 원어민으로서 해당 텍스트를 읽지만 미국 영어 억양으로 읽게 됩니다.

> **Note**
>
> 일부 언어들은 다른 언어들에 비해 서로 간에 더 큰 유사성을 지니므로 그러한 언어들의 조합은 다른 조합에 비해 더 좋은 결과를 만들어냅니다.

지원되는 SSML 태그

Amazon Polly는 다음과 같은 SSML 태그를 지원합니다.

작업	SSML 태그	뉴럴 보이스 사용 가능 여부	긴 형식의 음성을 사용할 수 있음	제너레이티브 보이스를 통한 가용성
일시 중지 추가	<break>	완전히 사용 가능	완전히 사용 가능	완전히 사용 가능
단어 강조	<emphasis>	사용할 수 없음	사용할 수 없음	사용할 수 없음
특정 단어에 대하여 다른 언어 지정	<lang>	완전히 사용 가능	완전히 사용 가능	완전히 사용 가능

작업	SSML 태그	뉴럴 보이스 사용 가능 여부	긴 형식의 음성을 사용할 수 있음	제너레이티브 보이스를 통한 가용성
텍스트에 사용자 지정 태그 배치	<mark>	완전히 사용 가능	완전히 사용 가능	완전히 사용 가능
단락 사이에 일시 중지 추가	<p>	완전히 사용 가능	완전히 사용 가능	완전히 사용 가능
철자대로의 발음 사용	<phoneme>	완전히 사용 가능	완전히 사용 가능	사용할 수 없음
볼륨, 말하기 속도 및 피치 제어	<prosody>	부분 사용 가능	부분 사용 가능	사용할 수 없음
합성된 스피치의 최대 기간 설정	<prosody amazon:max-duration>	사용할 수 없음	사용할 수 없음	사용할 수 없음
문장 사이에 일시 중지 추가	<s>	완전히 사용 가능	완전히 사용 가능	완전히 사용 가능
특별한 유형의 단어를 말하는 방식 제어	<say-as>	부분 사용 가능	부분 사용 가능	부분 사용 가능
SSML로 확장된 텍스트 식별	<speak>	완전히 사용 가능	완전히 사용 가능	완전히 사용 가능
두문자어 및 약어 발음	<sub>	완전히 사용 가능	완전히 사용 가능	완전히 사용 가능
스피치의 일부를 구체적으로 지정함으로써 발음 개선	<w>	완전히 사용 가능	완전히 사용 가능	완전히 사용 가능
숨소리 추가	<amazon:auto-breaths>	사용할 수 없음	사용할 수 없음	사용할 수 없음

작업	SSML 태그	뉴럴 보이스 사용 가능 여부	긴 형식의 음성을 사용할 수 있음	제너레이티브 보이스를 통한 가용성
뉴스 진행자 말투	<amazon:domain name="news">	신경 음성만 선택	사용할 수 없음	사용할 수 없음
동적 범위 압축 추가	<amazon:effect name="drc">	완전히 사용 가능	완전히 사용 가능	사용할 수 없음
부드러운 어투	<amazon:effect phonation="soft">	사용할 수 없음	사용할 수 없음	사용할 수 없음
음색 제어	vocal-tract-length<아마존: 이펙트 >	사용할 수 없음	사용할 수 없음	사용할 수 없음
속삭임 기능	<amazon:effect name="whispered">	사용할 수 없음	사용할 수 없음	사용할 수 없음

> **Note**
>
> 지원되지 않는 SSML 태그를 표준, 신경 또는 롱폼 형식으로 사용할 경우 오류가 발생합니다.

SSML로 확장된 텍스트 식별

<speak>

이 태그는 생성형, 장형, 신경망 및 표준 TTS 형식에서 지원됩니다.

<speak> 태그는 모든 Amazon Polly SSML 텍스트의 루트 요소입니다. SSML로 확장된 모든 텍스트는 한 쌍의 <speak> 태그로 묶여야 합니다.

```
<speak>Mary had a little lamb.</speak>
```

일시 중지 추가

<break>

이 태그는 제너레이티브, 롱폼, 뉴럴 및 표준 TTS 형식에서 지원됩니다.

텍스트에 일시 중지를 추가하려면 <break> 태그를 사용합니다. 강도를 기준으로 일시 중지를 설정하거나(쉼표, 문장 또는 단락 다음에 오는 일시 중지에 해당), 특정 시간 길이(초 또는 밀리초)로 설정할 수 있습니다. 일시 중지 길이를 결정할 속성을 지정하지 않을 경우, Amazon Polly는 기본값인 <break strength="medium"/>을 사용합니다. 이는 쉼표 다음에 일시 중지를 추가합니다.

strength 속성 값

- none: 일시 중지가 존재하지 않습니다. none을 사용하여 마침표 뒤와 같이 일반적으로 발생하는 일시 중지를 제거합니다.
- x-weak: none과 동일한 강도를 지니며 일시 중지가 없습니다.
- weak: 쉼표 다음에 오는 일시 중지와 동일한 지속 시간의 일시 중지를 설정합니다.
- medium: weak와 동일한 강도를 지닙니다.
- strong: 문장 다음에 오는 일시 중지와 동일한 지속 시간의 일시 중지를 설정합니다.
- x-strong: 단락 다음에 오는 일시 중지와 동일한 지속 시간의 일시 중지를 설정합니다.

time 속성 값

- *[number]*s: 일시 중시 기간(초)입니다. 최대 지속 시간은 10s입니다.
- *[number]*ms: 일시 중시 기간(밀리초)입니다. 최대 지속 시간은 10000ms입니다.

예:

```
<speak>
     Mary had a little lamb <break time="3s"/>Whose fleece was white as snow.
</speak>
```

속성을 break 태그와 함께 사용하지 않는 경우, 결과는 텍스트에 따라 달라집니다.

- break 태그 옆에 다른 문장 부호가 없는 경우, <break strength="medium"/>(쉼표 길이 일시 중지)이 생성됩니다.

- 태그가 쉼표 옆에 있는 경우, 태그가 <break strength="strong"/>(문장 길이 일시 중지)으로 업그레이드됩니다.
- 태그가 마침표 옆에 있는 경우, 태그가 <break strength="x-strong"/>(단락 길이 일시 중지)으로 업그레이드됩니다.

단어 강조

<emphasis>

표준 TTS 형식만 이 태그를 지원합니다.

단어를 강조하려면 <emphasis> 태그를 사용합니다. 단어를 강조하면 말하기 속도와 볼륨이 변경됩니다. 더 강조하면 Amazon Polly가 텍스트를 더 크고 느리게 읽습니다. 덜 강조하면 더 작고 빠르게 읽습니다. 강조 수준을 지정하려면 `level` 속성을 사용합니다.

`level` 속성 값

- `Strong`: 스피치가 더 크고 느려지도록 볼륨을 높이고 말하기 속도를 늦춥니다.
- `Moderate`: 볼륨을 높이고 말하기 속도가 느려지지만 `strong`일 때보다는 덜합니다. 기본값은 `Moderate`입니다.
- `Reduced`: 볼륨을 낮추고 말하기 속도를 높입니다. 스피치가 부드럽고 빨라집니다.

> **Note**
>
> 음성의 일반적인 말하기 속도 및 볼륨은 `moderate` 수준과 `reduced` 수준 사이입니다.

예:

```
<speak>
     I already told you I <emphasis level="strong">really like</emphasis> that person.
</speak>
```

특정 단어에 대하여 다른 언어 지정

<lang>

이 태그는 제너레이티브, 롱폼, 뉴럴 및 표준 TTS 형식에서 지원됩니다.

<lang> 태그를 사용하여 특정 단어, 문구 또는 문장에 대해 다른 언어를 지정합니다. 일반적으로 외국어(단어 및 구)를 한 쌍의 `<lang>` 태그로 묶으면 더 잘 읽혀집니다. 언어를 지정하려면 `xml:lang` 속성을 사용합니다. 사용 가능한 언어에 대한 전체 목록은 Amazon Polly의 언어를 참조하세요.

`<lang>` 태그를 적용하지 않으면 입력 텍스트 안의 모든 단어를 `voice-id`에 지정한 음성의 언어로 읽습니다. `<lang>` 태그를 적용하면 단어를 해당 언어로 읽습니다.

예를 들어 `voice-id`이(가) (미국 영어 원어민인) Joanna인 경우 Amazon Polly는 다음 문장을 프랑스어 억양을 사용하지 않고 Joanna 음성으로 읽습니다.

```
<speak>
     Je ne parle pas français.
</speak>
```

`<lang>` 태그와 함께 Joanna 음성을 사용하면 Amazon Polly는 그 문장을 Joanna 음성을 통해 미국식 억양의 불어로 읽습니다.

```
<speak>
     <lang xml:lang="fr-FR">Je ne parle pas français.</lang>.
</speak>
```

Joanna의 음성은 원어민 프랑스어가 아니기 때문에 발음은 그녀의 모국인 미국 영어를 따릅니다. 예를 들면, 완벽한 프랑스어 발음에서는 단어 français에 목젖 울림 /R/이 있지만, Joanna의 미국식 영어 음성은 이 음소를 그에 해당하는 /r/ 소리로 발음합니다.

다음 텍스트에 이탈리아어 원어민인 Giorgio의 `voice-id`를 사용하면 Amazon Polly는 해당 문장을 Giorgio 음성을 통해 이탈리아어식으로 발음합니다.

```
<speak>
     Mi piace Bruce Springsteen.
</speak>
```

동일한 음성을 다음 `<lang>` 태그와 함께 사용하면 Amazon Polly는 Bruce Springsteen을 이탈리아식 억양의 영어로 발음합니다.

```
<speak>
```

```
     Mi piace <lang xml:lang="en-US">Bruce Springsteen.</lang>
</speak>
```

이 태그는 음성을 합성할 때 옵션 DefaultLangCode옵션 대신 사용할 수도 있습니다. 그러나, 이렇게 하려면 SSML을 사용하여 텍스트 서식을 지정해야 합니다.

텍스트에 사용자 지정 태그 배치

<mark>

이 태그는 제너레이티브, 롱폼, 뉴럴 및 표준 TTS 포맷에서 지원됩니다.

텍스트 내에 사용자 지정 태그를 넣으려면 <mark> 태그를 사용합니다. Amazon Polly는 태그에 대해 아무 조치도 취하지 않지만 SSML 메타데이터에서 태그의 위치를 반환합니다. 이 태그의 이름은 다음 형식에 따른 것이라면 어떤 것이든 가능합니다.

```
<mark name="tag_name"/>
```

예를 들어, 태그 이름이 "animal"인 경우 입력 텍스트는 다음과 같습니다.

```
<speak>
     Mary had a little <mark name="animal"/>lamb.
</speak>
```

Amazon Polly는 다음 SSML 메타데이터를 반환할 수 있습니다.

```
{"time":767,"type":"ssml","start":25,"end":46,"value":"animal"}
```

단락 사이에 일시 중지 추가

<p>

이 태그는 제너레이티브, 롱폼, 뉴럴 및 표준 TTS 형식에서 지원됩니다.

텍스트의 단락 사이에 일시 중지를 추가하려면 <p> 태그를 사용합니다. 이 태그를 사용하면 쉼표가 있는 자리나 문장 끝에 원어민이 대체로 사용하는 것보다 긴 일시 중지가 적용됩니다. <p> 태그를 사용해 단락을 묶습니다.

```
<speak>
```

```
    <p>This is the first paragraph. There should be a pause after this text is
 spoken.</p>
    <p>This is the second paragraph.</p>
</speak>
```

이는 <break strength="x-strong"/>를 사용해 일시 중지를 지정하는 것과 동일합니다.

철자대로의 발음 사용

<phoneme>

이 태그는 롱폼, 신경 및 표준 TTS 형식으로 지원됩니다.

Amazon Polly에서 특정 텍스트에 대해 철자대로의 발음을 사용하게 하려면 <phoneme> 태그를 사용합니다.

두 속성에는 <phoneme> 태그가 필요합니다. 이러한 태그는 Amazon Polly에서 사용되는 음성 기호와 정확한 발음을 위한 발음 기호를 나타냅니다.

- `alphabet`
  - `ipa` - IPA(International Phonetic Alphabet)가 사용되어야 함을 나타냅니다.
  - `x-sampa` - X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet)가 사용되어야 함을 나타냅니다.
- ph
  - 발음에 사용될 발음 기호를 지정합니다. 자세한 정보는 지원되는 언어의 음소 및 Viseme 표을(를) 참조하세요.

<phoneme> 태그를 사용하면, Amazon Polly는 선택된 음성이 사용하는 언어와 기본값으로써 연결되는 표준 발음 대신 ph 속성이 지정한 발음을 사용합니다.

예를 들어, 단어 "pecan"은 두 가지 방식으로 발음될 수 있습니다. 다음 예제에서는 “pecan”에 각 줄마다 다른 발음이 할당됩니다. Amazon Polly는 기본 발음을 사용하는 대신 ph 속성에 지정된 대로 pecan을 발음합니다.

IPA(International Phonetic Alphabet)

```
<speak>
    You say, <phoneme alphabet="ipa" ph="p##k##n">pecan</phoneme>.
    I say, <phoneme alphabet="ipa" ph="#pi.kæn">pecan</phoneme>.
```

```
</speak>
```

X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet)

```
<speak>
     You say, <phoneme alphabet='x-sampa' ph='pI"kA:n'>pecan</phoneme>.
     I say, <phoneme alphabet='x-sampa' ph='"pi.k{n'>pecan</phoneme>.
</speak>
```

또한 표준 중국어는 음운 발음에 병음을 사용합니다.

병음

```
<speak>
     ## <phoneme alphabet="x-amazon-pinyin" ph="bo2">#</phoneme>#
     ## <phoneme alphabet="x-amazon-pinyin" ph="bao2">#</phoneme>#
</speak>
```

일본어는 요미가나와 가나 발음을 사용합니다.

요미가나

```
<speak>
     ###<phoneme alphabet="x-amazon-yomigana" ph="####">##</phoneme>###
     ###<phoneme alphabet="x-amazon-yomigana" ph="####">##</phoneme>###
     ###<phoneme alphabet="x-amazon-yomigana" ph="Hirokazu">##</phoneme>###
</speak>
```

발음: 가나

```
<speak>
     ###<phoneme alphabet="x-amazon-pron-kana" ph="##'##">##</phoneme>###
</speak>
```

볼륨, 말하기 속도 및 피치 제어

<prosody>

Prosody 태그 속성은 표준 TTS 음성으로 완벽하게 지원됩니다. 신경 및 롱폼 음성은 `volume` 및 `rate` 속성을 지원하지만 `pitch` 속성은 지원하지 않습니다.

선택된 음성의 볼륨, 속도 또는 피치를 제어하려면 prosody 태그를 사용합니다.

볼륨, 스피치 속도 및 피치는 선택된 특정 음성에 따라 다릅니다. 서로 다른 언어를 사용하는 음성 간의 차이 외에도 동일한 언어를 사용하는 각 개인의 음성 간에도 차이가 존재합니다. 따라서 속성은 모든 언어에 걸쳐 유사하지만 각 언어들 사이에 분명한 차이가 존재하며 절대값이란 있을 수 없습니다.

prosody 태그에는 세 개의 속성이 있으며 각 속성에는 해당 속성을 설정하는 데 사용 가능한 몇 가지 값이 있습니다. 각 속성은 동일한 구문을 사용합니다.

```
<prosody attribute="value"></prosody>
```

- volume
  - default: 볼륨을 현재 음성의 기본 수준으로 재설정합니다.
  - silent, x-soft, soft, medium, loud, x-loud: 볼륨을 현재 음성에 사전 정의된 값으로 설정합니다.
  - +ndB, -ndB: 볼륨을 현재 볼륨 레벨으로부터 가감합니다. +0dB 값은 변경 없음, +6dB 값은 현재 볼륨의 약 2배, -6dB 값은 현재 볼륨의 약 절반을 의미합니다.

  예를 들어, 특정 구절의 볼륨을 다음과 같은 방식으로 설정할 수 있습니다.

  ```
  <speak>
       Sometimes it can be useful to <prosody volume="loud">increase the volume
       for a specific speech.</prosody>
  </speak>
  ```

  또는 다음과 같은 방식으로 설정할 수 있습니다.

  ```
  <speak>
       And sometimes a lower volume <prosody volume="-6dB">is a more effective way of
       interacting with your audience.</prosody>
  </speak>
  ```

- rate
  - x-slow, slow, medium, fast, x-fast. 선택된 음성에 대해 피치를 사전 정의된 값으로 설정합니다.
  - n%: 말하기 속도의 백분율 가감 예를 들어 100%의 값은 말하기 속도를 변경하지 않는 것이고, 200%의 값은 기본 속도의 2배에 해당하는 말하기 속도를 의미하며 50%의 값은 기본 속도의 절반에 해당하는 말하기 속도를 의미합니다. 이 값의 범위는 20%~200%입니다.

예를 들어, 특정 구절의 스피치 속도를 다음과 같은 방식으로 설정할 수 있습니다.

```
<speak>
     For dramatic purposes, you might wish to <prosody rate="slow">slow up the
 speaking
     rate of your text.</prosody>
</speak>
```

또는 다음과 같은 방식으로 설정할 수 있습니다.

```
<speak>
     Although in some cases, it might help your audience to <prosody rate="85%">slow
     the speaking rate slightly to aid in comprehension.</prosody>
</speak>
```

- `pitch`
  - `default`: 현재 음성의 음색을 기본 수준으로 재설정합니다.
  - `x-low`, `low`, `medium`, `high`, `x-high`: 현재 음성의 음색을 사전 정의된 값으로 설정합니다.
  - `+n%` 또는 `-n%`: 상대적 백분율로 음색을 조절합니다. 예를 들어 값 `+0%`는 기준 음색 변경이 없음을 의미하고, `+5%`는 약간 더 높은 기준 음색을 제공하며, `-5%`는 약간 더 낮은 기준 음색을 가져옵니다.

  예를 들어, 특정 구절의 피치를 다음과 같은 방식으로 설정할 수 있습니다.

```
<speak>
     Do you like sythesized speech <prosody pitch="high">with a pitch that is higher
     than normal?</prosody>
</speak>
```

  또는 다음과 같은 방식으로 설정할 수 있습니다.

```
<speak>
     Or do you prefer your speech <prosody pitch="-10%">with a somewhat lower pitch?
</prosody>
</speak>
```

<prosody> 태그는 최소 1개의 속성을 포함해야 하나, 동일한 태그 안에 여러 개를 포함할 수도 있습니다.

```
<speak>
     Each morning when I wake up, <prosody volume="loud" rate="x-slow">I speak
     quite slowly and deliberately until I have my coffee.</prosody>
</speak>
```

또한 다음과 같은 방식으로 중첩된 태그와 조합할 수 있습니다.

```
<speak>
     <prosody rate="85%">Sometimes combining attributes <prosody pitch="-10%">can
     change the impression your audience has of a voice</prosody> as well.</prosody>

</speak>
```

합성된 스피치의 최대 기간 설정

<prosody amazon:max-duration>

현재 TTS 형식만 이 태그를 지원합니다.

합성될 때 스피치에 걸리는 시간을 제어하려면 <prosody> 태그를 amazon:max-duration 속성과 함께 사용합니다.

합성된 스피치의 기간은 선택한 음성에 따라 조금 다릅니다. 이로 인해 합성된 스피치를 정확한 시간이 필요한 시각 자료 또는 기타 활동과 일치시킬 수 없을 수 있습니다. 이 문제는 특정 구가 다른 언어와 크게 다를 수 있다고 말하는 데 걸리는 시간 때문에 번역 애플리케이션의 경우 더 확대됩니다.

<prosody amazon:max-duration> 태그는 합성된 스피치를 원하는 시간(기간)과 일치시킵니다.

이 태그는 다음 구문을 사용합니다.

```
<prosody amazon:max-duration="time duration">
```

<prosody amazon:max-duration> 태그를 사용하면 기간을 초 또는 밀리초 단위로 지정할 수 있습니다.

- *n*s: 최대 기간(초)
- *n*ms: 최대 기간(밀리초)

예를 들어, 다음 말한 텍스트의 최대 기간은 2초입니다.

```
<speak>
     <prosody amazon:max-duration="2s">
          Human speech is a powerful way to communicate.
     </prosody>
</speak>
```

태그 안에 배치된 텍스트는 지정된 기간을 초과하지 않습니다. 선택한 음성 또는 언어가 일반적으로 기간보다 더 오래 걸리는 경우 Amazon Polly에서는 지정된 기간과 맞도록 스피치 속도를 높입니다.

지정된 기간이 일반 속도로 텍스트를 읽는 데 걸리는 시간보다 긴 경우 Amazon Polly에서는 스피치를 일반적으로 읽습니다. 스피치 속도를 줄이거나 무음을 추가하지 않으므로 결과 오디오가 요청된 사항보다 더 짧습니다.

> **Note**
>
> Amazon Polly에서는 일반 속도의 5배 이상으로 속도를 올리지 않습니다. 이 속도보다 빠르게 텍스트를 읽는 경우는 일반적으로 성립되지 않습니다. 스피치 속도를 최대로 올리는 경우에도 지정된 기간에 스피치를 맞출 수 없는 경우 오디오 속도가 높아지지만 지정된 기간보다 오래 지속됩니다.

`<prosody amazon:max-duration>` 태그에 한 문장 또는 여러 문장을 포함할 수 있으며, 텍스트 내에 여러 `<prosody amazon:max-duration>` 태그를 사용할 수 있습니다.

예:

```
<speak>
     <prosody amazon:max-duration="2400ms">
        Human speech is a powerful way to communicate.
     </prosody>
     <break strength="strong"/>
     <prosody amazon:max-duration="5100ms">
        Even a simple ‘Hello’ can convey a lot of information depending on the pitch,
 intonation, and tempo.
     </prosody>
     <break strength="strong"/>
     <prosody amazon:max-duration="8900ms">
        We naturally understand this information, which is why speech is ideal for
 creating applications where
        a screen isn’t practical or possible, or simply isn’t convenient.
     </prosody>
```

```
</speak>
```

<prosody amazon:max-duration> 태그를 사용하면 Amazon Polly에서 합성된 스피치가 반환될 때 지연 시간이 늘어날 수 있습니다. 지연의 정도는 특정 구절 및 구절의 길이에 따라 다릅니다. 비교적 짧은 텍스트 구절로 구성된 텍스트를 사용하는 것이 좋습니다.

제한 사항

<prosody amazon:max-duration> 태그 사용 방법과 이 태그가 다른 SSML 태그와 함께 작동하는 방법 모두에 제한이 있습니다.

- <prosody amazon:max-duration> 태그 내에 있는 텍스트는 1500자를 초과할 수 없습니다.
- <prosody amazon:max-duration> 태그를 중첩할 수 없습니다. 다른 태그 내에 <prosody amazon:max-duration> 태그 하나를 배치하는 경우 Amazon Polly에서는 내부 태그를 무시합니다.

  예를 들어, 다음의 경우 <prosody amazon:max-duration="5s"> 태그가 무시됩니다.

```
<speak>
     <prosody amazon:max-duration="16s">
          Human speech is a powerful way to communicate.

          <prosody amazon:max-duration="5s">
               Even a simple ‘Hello’ can convey a lot of information depending on the
 pitch, intonation, and tempo.
          </prosody>

          We naturally understand this information, which is why speech is ideal for
 creating applications where a screen isn’t practical or possible, or simply isn’t
 convenient.
     </prosody>
</speak>
```

- <prosody> 태그를 rate 태그 내의 <prosody amazon:max-duration> 속성과 함께 사용할 수 없습니다. 이는 텍스트를 말할 때 해당 태그와 속성이 모두 속도에 영향을 주기 때문입니다.

  다음 예에서 Amazon Polly는 <prosody rate="2"> 태그를 무시합니다.

```
<speak>
```

```
     <prosody amazon:max-duration="7500ms">
          Human speech is a powerful way to communicate.

          <prosody rate="2">
               Even a simple ‘Hello’ can convey a lot of information depending on the
 pitch, intonation, and tempo.
          </prosody>
     </prosody>
</speak>
```

일시 중지 및 **max-duration**

max-duration 태그를 사용하면 텍스트 내에 일시 중지를 계속 삽입할 수 있습니다. 그러나 Amazon Polly에서는 스피치의 최대 기간을 계산할 때 일시 중지의 길이를 포함합니다. 또한 Amazon Polly에서는 쉼표와 마침표가 구절 내에 배치되어 있는 경우 발생하는 짧은 일시 중지를 유지하며, 이를 최대 기간에 포함합니다.

예를 들어, 다음 블록에서 600밀리초의 브레이크와 쉼표 및 마침표로 인해 발생한 브레이크는 8초 스피치 내에서 발생합니다.

```
<speak>
     <prosody amazon:max-duration="8s">
          Human speech is a powerful way to communicate.
          <break time="600ms"/>
          Even a simple ‘Hello’ can convey a lot of information depending on the pitch,
 intonation, and tempo.
     </prosody>
</speak>
```

문장 사이에 일시 중지 추가

<s>

이 태그는 제너레이티브, 롱폼, 뉴럴 및 표준 TTS 형식에서 지원됩니다.

텍스트의 행 또는 문장 사이에 일시 중지를 추가하려면 <s>p> 태그를 사용합니다. 이 태그를 사용하면 다음과 동일한 효과가 나옵니다.

- 마침표(.)로 끝나는 문장
- <break strength="strong"/>으로 일시 중지 지정

<break> 태그와 달리, <s> 태그로 문장을 묶습니다. 이러한 방식은 시(詩)처럼 문장보다는 행으로 이루어진 스피치를 합성하는 데 유용합니다.

다음 예에서는 <s> 태그로 첫 번째 문장과 두 번째 문장 다음에 모두 일시 중지를 짧게 생성합니다. 최종 문장에는 <s> 태그가 없지만 문장 끝에 마침표가 있으므로 문장 뒤에 짧은 일시 중지가 생기게 됩니다.

```
<speak>
     <s>Mary had a little lamb</s>
     <s>Whose fleece was white as snow</s>
     And everywhere that Mary went, the lamb was sure to go.
</speak>
```

특별한 유형의 단어를 말하는 방식 제어

<say-as>

characters옵션을 제외한 <say-as> 태그는 제너레이티브, 롱폼, 뉴럴 및 표준 TTS 포맷에서 지원됩니다. Amazon Polly가 신경 음성을 사용 중이고 런타임에 characters 옵션이 포함된 <say-as> 태그가 발견되는 경우 영향을 받는 문장이 관련 표준 음성을 사용하여 합성된다는 점에 유의하세요. 하지만 영향을 받은 문장에는 여전히 신경망 음성을 사용하는 것처럼 요금이 청구됩니다.

<say-as> 태그에서 interpret-as 속성을 사용하여 Amazon Polly에게 특정 문자, 단어 또는 숫자를 읽는 방식을 알려줍니다. 이 태그를 사용하면 추가 컨텍스트를 제공하여 Amazon Polly가 텍스트를 렌더링하는 방법에 대한 모든 불확실성을 제거할 수 있습니다.

<say-as> 태그는 1개의 속성, interpret-as를 사용하는데 이 속성은 여러 개의 사용 가능한 값을 사용합니다. 각각은 동일한 구문을 사용합니다.

```
<say-as interpret-as="value">[text to be interpreted]</say-as>
```

interpret-as에는 다음과 같은 값을 사용할 수 있습니다.

- characters또는spell-out: 에서와 같이 텍스트의 각 글자를 입력합니다. a-b-c

> **Note**
>
> 신경 음성의 경우 현재 이 옵션이 지원되지 않습니다. 신경망 음성을 사용 중이고 런타임 시 Amazon Polly에서 이 SSML 코드를 발견하면 영향을 받는 문장이 관련 표준 음성을 사용하

여 합성됩니다. 하지만 신경 음성을 사용하는 것처럼 이 문장에 요금이 부과된다는 점에 유의하세요.

- `cardinal` 또는 `number`: 숫자로 된 텍스트를 1,234와 같이 값을 기수로 해석합니다.
- `ordinal`: 숫자로 된 텍스트를 1,234번째와 같이 서수로 해석합니다.
- `digits`: 1-2-3-4와 같이 각 자릿수를 개별적으로 발음합니다.
- `fraction`: 숫자로 된 텍스트를 분수로 해석합니다. 이 태그는 3/20과 같은 일반 분수와 2 ½과 같은 혼합 분수 모두에 적용됩니다. 자세한 내용은 아래를 참조하세요.
- `unit`: 숫자로 된 텍스트를 측정값으로 해석합니다. 값은 `1/2inch`와 같이 숫자 또는 분수 다음에, 사이의 공백이 없이, 단위 1가지가 뒤따르거나. `1meter`와 같이 오직 1단위만 있어야 합니다.
- `date`: 텍스트를 날짜로 해석합니다. 날짜의 형식은 형식 속성과 함께 지정되어야 합니다(아래 참조). 자세한 내용은 아래를 참조하세요.
- `time`: `1'21"`과 같이 숫자로 된 텍스트를 기간(분과 초)으로 해석합니다.
- `address`: 텍스트를 거리 주소의 일부분으로 해석합니다.
- `expletive`: 태그 내에 포함된 콘텐츠에 "삐" 소리를 냅니다.
- `telephone`: `2025551212`과 같이 숫자로 된 텍스트를 7자리 또는 10자리의 전화 번호로 해석합니다. 또한 `2025551212x345`와 같이 이 값을 전화 번호에 구내 번호를 추가하는 데 사용할 수도 있습니다. 자세한 내용은 아래를 참조하세요.

> **Note**
>
> 현재 모든 언어에서 `telephone` 옵션을 사용할 수 있는 것은 아닙니다. 그러나 영어 변종(en-AU, en-GB, en-IN, en-US 및 en-GB-WLS), 스페인어 변종(es-ES, es-MX 및 es-US), 프랑스어 변종(fr-FR 및 fr-CA), 포르투갈어 변종(pt-BR 및 pt-PT), 독일어(de-DE), 이탈리아어(it-IT), 일본어(ja-JP), 러시아어(ru-RU)에서는 사용할 수 있습니다. 또한 아랍어 (arb) 와 같은 언어에서는 설정된 번호를 전화 번호로 자동 처리하므로 실제로 SSML 태그를 구현하지 않는 경우도 있다는 점도 유의해야 합니다. `telephone`

분수

Amazon Polly는 `interpret-as="fraction"` 속성을 갖는 `say-as` 태그 안의 값을 일반적인 분수로 해석합니다. 다음은 분수에 대한 구문입니다.

- 분수

구문: ##/## 예: 2/9.

예를 들어 <say-as interpret-as="fraction">2/9</say-as>는 "9분의 2"로 발음됩니다.

- 음수가 아닌 혼수

  구문: ##+##/## 예: 3+1/2.

  예를 들어 <say-as interpret-as="fraction">3+1/2</say-as>은 "삼과 이분의 일"로 발음됩니다.

> **Note**
>
> “3"과 “1/2" 사이에는 반드시 +가 있어야 합니다. Amazon Polly는 “3 1/2"와 같이 +가 없는 혼합 숫자를 지원하지 않습니다.

날짜

interpret-as가 date로 설정된 경우 날짜 형식도 나타내야 합니다.

이 방식은 다음 구문을 사용합니다.

```
<say-as interpret-as="date" format="format">[date]</say-as>
```

예:

```
<speak>
     I was born on <say-as interpret-as="date" format="mdy">12-31-1900</say-as>.
</speak>
```

다음 형식은 date 속성과 함께 사용할 수 있습니다.

- mdy: M. onth-day-year
- dmy: Day-month-year.
- ymd: 와이ear-month-day.
- md: 월-일
- dm: 월-일

- ym: 년-월
- my: 년-월
- d: 일.
- m: Month.
- y: Year.
- yyyymmdd: 와이ear-month-day. 이 형식을 사용하는 경우 Amazon Polly가 물음표를 사용하여 날짜의 일부분을 건너뛰도록 할 수 있습니다.

  예를 들어, Amazon Polly는 다음 항목을 "9월 22일"로 렌더링합니다.

  ```
  <say-as interpret-as="date">????0922</say-as>
  ```

  Format은 필요하지 않기 때문입니다.

전화번호

Amazon Polly는 <say-as> 태그가 없어도 텍스트의 형식에 따라 올바르게 제공된 텍스트를 해석하려고 시도합니다. 예를 들어, 텍스트에 "202-555-1212"가 포함된 경우 Amazon Polly는 10자리 전화 번호로 해석한 다음 각 대시를 짧게 일시 중지하면서 각 개별 숫자를 따로따로 읽습니다. 이 경우 <say-as interpret-as="telephone">을 사용하지 않아도 됩니다. 그러나 텍스트 "2025551212"를 제공하고 Amazon Polly가 전화 번호로 말하기를 원하는 경우 <say-as interpret-as="telephone">을 지정해야 합니다.

각 요소의 해석에 사용되는 논리는 언어별로 다릅니다. 예를 들어, 미국 및 영국 영어는 전화 번호를 발음하는 방법이 다릅니다. 영국 영어는 동일한 숫자가 이어서 나오면 "double five" 또는 "triple four"처럼 그룹으로 묶어서 말합니다. 그 차이점을 확인하려면 다음의 예를 미국 음성과 영국 음성으로 테스트해 봅니다.

```
<speak>
     Richard's number is <say-as interpret-as="telephone">2122241555</say-as>
</speak>
```

두문자어 및 약어 발음

<sub>

이 태그는 제너레이티브, 롱폼, 뉴럴 및 표준 TTS 형식에서 지원됩니다.

<sub> 태그를 alias 속성과 함께 사용하여 두문자어나 약어 등 선택된 텍스트를 다른 단어(또는 발음)로 대체합니다.

이때 다음 구문을 사용합니다.

```
<sub alias="new word">abbreviation</sub>
```

다음 예에서는 오디오 콘텐츠의 의미를 더 선명히 하기 위해 해당 원소 기호를 "Mercury"란 통명으로 대체합니다.

```
<speak>
     My favorite chemical element is <sub alias="Mercury">Hg</sub>, because it looks so
 shiny.
</speak>
```

스피치의 일부를 구체적으로 지정함으로써 발음 개선

<w>

이 태그는 제너레이티브, 롱폼, 뉴럴 및 표준 TTS 형식에서 지원됩니다.

<w> 태그를 사용해 해당 단어가 스피치 안에 위치하는 부분 또는 대체 의미를 구체적으로 지정하여 단어의 발음을 사용자 지정할 수 있습니다. role 속성을 사용하여 이를 수행합니다.

이 태그는 다음 구문을 사용합니다.

```
<w role="attribute">text</w>
```

다음 값은 role 속성에 사용할 수 있습니다.

스피치의 일부를 지정하려면:

- amazon:VB: 단어를 동사(단순 현재)로 해석합니다.
- amazon:VBD: 단어를 과거형 동사로 해석합니다.
- amazon:DT: 단어를 한정사로 해석합니다.
- amazon:IN: 단어를 전치사로 해석합니다.

- amazon:JJ: 단어를 형용사로 해석합니다.
- amazon:NN: 단어를 명사로 해석합니다.

예를 들어, "read" 단어의 미국식 영어 발음의 경우 이 단어가 스피치 안에 위치하는 부분에 따라 태그를 사용하여 다른 결과를 낼 수 있습니다.

```
<speak>
     The word <say-as interpret-as="characters">read</say-as> may be interpreted
     as either the present simple form <w role="amazon:VB">read</w>, or the past
     participle form <w role="amazon:VBD">read</w>.
</speak>
```

특정 의미를 지정하려면

- amazon:DEFAULT: 단어의 기본 의미를 사용합니다.
- amazon:SENSE_1: 있는 경우 단어의 기본이 아닌 의미를 사용합니다. 예를 들어 단어 "bass"는 의미에 따라 다르게 발음됩니다. 기본 의미는 화음에서 가장 낮은 부분입니다. 대체 의미는 민물고기의 한 종으로, 역시 "bass"라고 하지만 다르게 발음됩니다. <w role="amazon:SENSE_1">bass</w>을(를) 사용하면 해당 오디오 텍스트를 기본이 아닌 발음(민물고기)으로 렌더링합니다.

다음을 합성하면 발음과 의미의 차이를 들을 수 있습니다.

```
<speak>
    Depending on your meaning, the word <say-as interpret-as="characters">bass</say-
as>
    may be interpreted as either a musical element: bass, or as its alternative
 meaning,
    a freshwater fish <w role="amazon:SENSE_1">bass</w>.
</speak>
```

> **Note**
>
> 일부 언어의 경우 지원되는 다른 스피치 부분을 선택할 수 있습니다.

숨소리 추가

<amazon:breath> 및 <amazon:auto-breaths>

표준 TTS 형식만 이 태그를 지원합니다.

자연스럽게 들리는 스피치에는 올바르게 말하는 단어와 숨소리가 모두 포함됩니다. 스피치 합성에 숨소리를 추가하면 더 자연스럽게 들리는 스피치를 만들 수 있습니다. <amazon:breath> 및 <amazon:auto-breaths> 태그는 숨소리를 제공합니다. 다음과 같은 옵션이 있습니다:

- 수동 모드: 사용자가 텍스트 내에 있는 숨소리의 위치, 길이, 볼륨을 설정합니다.
- 자동화 모드: Amazon Polly에서 숨소리를 스피치 출력에 자동으로 삽입합니다.
- 혼합 모드: 사용자와 Amazon Polly가 모두 숨소리를 추가합니다.

수동 모드

수동 모드에서는 숨소리를 배치할 입력 텍스트에 <amazon:breath/> 태그를 지정합니다. duration 및 volume 속성을 각각 사용하여 숨소리의 길이와 볼륨을 사용자 지정할 수 있습니다.

- duration: 숨소리의 길이를 제어합니다. 유효한 값은 default, x-short, short, medium, long, x-long입니다. 기본 값은 medium입니다.
- volume: 숨소리가 얼마나 크게 들리는지를 제어합니다. 유효한 값은 default, x-soft, soft, medium, loud, x-loud입니다. 기본 값은 medium입니다.

> **Note**
>
> 각 속성 값의 정확한 길이와 볼륨은 사용되는 특정 Amazon Polly 음성에 따라 다릅니다.

기본값을 사용하여 숨소리를 설정하려면 속성 없이 <amazon:breath/>을(를) 사용합니다.

예를 들어, 속성을 사용하여 숨소리의 기간과 볼륨을 중간으로 설정하려면 속성을 다음과 같이 설정합니다.

```
<speak>
     Sometimes you want to insert only <amazon:breath duration="medium" volume="x-
loud"/>a single breath.
</speak>
```

기본값을 사용하려면 태그만 사용합니다.

```
<speak>
     Sometimes you need <amazon:breath/>to insert one or more average breaths
 <amazon:breath/> so that the
     text sounds correct.
</speak>
```

다음과 같이 구절 내에 개별 숨소리를 추가할 수 있습니다.

```
<speak>
     <amazon:breath duration="long" volume="x-loud"/> <prosody rate="120%"> <prosody
 volume="loud">
     Wow! <amazon:breath duration="long" volume="loud"/> </prosody> That was quite
 fast. <amazon:breath
     duration="medium" volume="x-loud"/> I almost beat my personal best time on this
 track. </prosody>
</speak>
```

자동화 모드

자동화 모드에서는 <amazon:auto-breaths> 태그를 사용하여 Amazon Polly에서 적절한 간격으로 숨소리 잡음을 자동으로 생성하도록 지정합니다. 간격의 빈도, 볼륨 및 기간을 설정할 수 있습니다. 자동화된 숨소리를 적용할 텍스트의 시작 부분에 </amazon:auto-breaths> 태그를 지정하고 끝 부분에 닫는 태그를 지정합니다.

> **Note**
>
> 수동 모드 태그인 <amazon:breath/>와는 달리, <amazon:auto-breaths> 태그에는 닫는 태그(</amazon:auto-breaths>)가 필요합니다.

다음과 같은 옵션 속성을 <amazon:auto-breaths> 태그와 함께 사용할 수 있습니다.

- volume: 숨소리가 얼마나 크게 들리는지를 제어합니다. 유효한 값은 default, x-soft, soft, medium, loud, x-loud입니다. 기본 값은 medium입니다.
- frequency: 텍스트에서 숨소리가 얼마나 자주 들리는지를 제어합니다. 유효한 값은 default, x-low, low, medium, high, x-high입니다. 기본 값은 medium입니다.
- duration: 숨소리의 길이를 제어합니다. 유효한 값은 default, x-short, short, medium, long, x-long입니다. 기본 값은 medium입니다.

기본적으로 숨소리의 빈도는 입력 텍스트에 따라 다릅니다. 하지만 주로 쉼표와 마침표 뒤에 숨소리가 들립니다.

다음 예제에서는 <amazon:auto-breaths> 태그를 사용하는 방법을 보여줍니다. 콘텐츠에 어떤 옵션을 사용할지 결정하려면 적용 가능한 예제를 Amazon Polly 콘솔에 복사하고 차이를 들어 봅니다.

- 옵션 파라미터 없이 자동화 모드를 사용합니다.

```
<speak>
     <amazon:auto-breaths>Amazon Polly is a service that turns text into lifelike
 speech,
     allowing you to create applications that talk and build entirely new categories
 of speech-
     enabled products. Amazon Polly is a text-to-speech service that uses advanced
 deep learning
     technologies to synthesize speech that sounds like a human voice. With dozens of
 lifelike
     voices across a variety of languages, you can select the ideal voice and build
 speech-
     enabled applications that work in many different countries.</amazon:auto-
breaths>
</speak>
```

- 볼륨 제어와 함께 자동화 모드를 사용합니다. 비지정 파라미터(duration 및 frequency)는 기본값(medium)으로 설정됩니다.

```
<speak>
     <amazon:auto-breaths volume="x-soft">Amazon Polly is a service that turns text
 into lifelike
     speech, allowing you to create applications that talk and build entirely new
 categories of
     speech-enabled products. Amazon Polly is a text-to-speech service, that uses
 advanced deep
     learning technologies to synthesize speech that sounds like a human voice. With
 dozens of
     lifelike voices across a variety of languages, you can select the ideal voice
 and build speech-
     enabled applications that work in many different countries.</amazon:auto-
breaths>
</speak>
```

- 빈도 제어와 함께 자동화 모드를 사용합니다. 비지정 파라미터(duration 및 volume)는 기본값(medium)으로 설정됩니다.

```
<speak>
     <amazon:auto-breaths frequency="x-low">Amazon Polly is a service that turns text
 into lifelike
     speech, allowing you to create applications that talk and build entirely new
 categories of
     speech-enabled products. Amazon Polly is a text-to-speech service, that uses
 advanced deep
     learning technologies to synthesize speech that sounds like a human voice. With
 dozens of
     lifelike voices across a variety of languages, you can select the ideal voice
 and build speech-
     enabled applications that work in many different countries.</amazon:auto-
breaths>
</speak>
```

- 다중 파라미터와 함께 자동화 모드를 사용합니다. 비지정 `Duration` 파라미터의 경우 Amazon Polly 에서 기본값(`medium`)을 사용합니다.

```
<speak>
     <amazon:auto-breaths volume="x-loud" frequency="x-low">Amazon Polly is a service
 that turns
     text into lifelike speech, allowing you to create applications that talk and
 build entirely new
     categories of speech-enabled products. Amazon Polly is a text-to-speech service,
 that uses
     advanced deep learning technologies to synthesize speech that sounds like a
 human voice. With
     dozens of lifelike voices across a variety of languages, you can select the
 ideal voice and build
     speech-enabled applications that work in many different countries.</amazon:auto-
breaths>
</speak>
```

뉴스 진행자 말투

<amazon:domain name="news">

뉴스 진행자 말투는 미국 영어(en-US) Matthew 및 Joanna 음성과 미국 스페인어(es-US) Lupe 음성, 영국 영어(en-GB) Amy 음성으로만 제공됩니다. `Neural` 형식을 사용할 때만 지원됩니다.

뉴스 진행자 스타일을 사용하려면 SSML 태그와 다음 구문을 사용합니다.

```
<amazon:domain name="news">text</amazon:domain>
```

예를 들어 다음과 같이 Amy 음성으로 뉴스 진행자 말투를 사용할 수 있습니다.

```
<speak>
<amazon:domain name="news">
From the Tuesday, April 16th, 1912 edition of The Guardian newspaper:

The maiden voyage of the White Star liner Titanic, the largest ship ever launched, has
 ended in disaster.

The Titanic started her trip from Southampton for New York on Wednesday. Late on Sunday
 night she struck
an iceberg off the Grand Banks of Newfoundland. By wireless telegraphy she sent out
 signals of distress,
and several liners were near enough to catch and respond to the call.
</amazon:domain>
</speak>
```

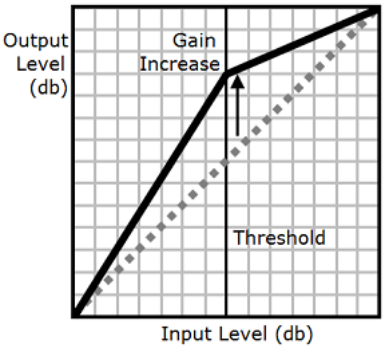
동적 범위 압축 추가

<amazon:effect name="drc">

이 태그는 롱폼, 신경 및 표준 TTS 형식으로 지원됩니다.

오디오 파일에 사용된 텍스트, 언어, 음성에 따라 소리 크기는 작은 소리부터 큰 소리까지 다를 수 있습니다. 자동차 소리 같은 주변 소리로 인해 작은 소리가 가려져서 오디오 트랙을 정확히 듣기 어려울 수 있습니다. 오디오 파일의 특정 사운드 볼륨을 높이려면 동적 범위 압축(drc) 태그를 사용하세요.

drc 태그는 오디오에 중간 "크기" 임계값을 설정하고, 해당 임계값 주변의 사운드 볼륨(게인)을 높입니다. 즉, 경계값에 가장 가까우면 최대 게인이 적용되고, 경계값에서 멀수록 게인이 감소합니다.



따라서 소음이 많은 환경에서 중간 크기 사운드를 쉽게 들을 수 있으며 오디오 파일을 전체적으로 듣기 쉽게 만듭니다.

drc 태그는 부울 파라미터입니다(참 또는 거짓). 이 태그는 <amazon:effect name="drc"> 구문을 사용하며 </amazon:effect>로 닫습니다.

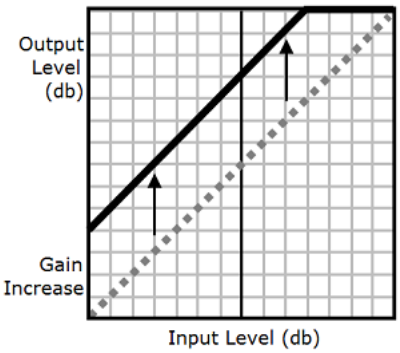
drc 태그는 Amazon Polly에서 지원하는 모든 음성이나 언어에 사용할 수 있습니다. 녹음 전체 섹션에 적용하거나 몇 단에만 적용할 수 있습니다. 예:

```
<speak>
     Some audio is difficult to hear in a moving vehicle, but <amazon:effect
 name="drc"> this audio
     is less difficult to hear in a moving vehicle.</amazon:effect>
</speak>
```

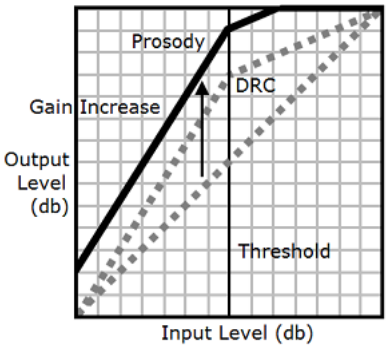
> **Note**
>
> 구문에 "drc"를 사용하면 대소문자를 구별합니다.amazon:effect

drc 태그에 prosody volume 사용

다음 그림에서 보는 것처럼, prosody volume 태그는 전체 오디오 파일의 볼륨을 원래 수준(점선)에서 조정된 수준(실선)으로 균일하게 올립니다. 파일의 특정 부분 볼륨을 더 높이려면 drc 태그와 prosody volume 태그를 함께 사용하세요. 조합 태그는 prosody volume 태그 설정에는 영향을 주지 않습니다.



drc 태그와 prosody volume 태그를 함께 사용하면 Amazon Polly는 drc 태그를 먼저 적용하여 중간 크기 사운드(경계값에 가까운 사운드)를 강화합니다. 그런 다음 prosody volume 태그를 적용하여 전체 오디오 트랙의 볼륨을 균일하게 더 높입니다.



태그를 함께 사용하려면 한 태그를 다른 태그 내에 넣으세요. 예:

```
<speak>
     <prosody volume="loud">This text needs to be understandable and loud.
 <amazon:effect name="drc">
     This text also needs to be more understandable in a moving car.</amazon:effect></
prosody>
</speak>
```

이 텍스트에서 `prosody volume` 태그는 전체 구절의 볼륨을 "크게" 높입니다. `drc` 태그는 둘째 문장에서 중간 크기 값의 볼륨을 강화합니다.

> **Note**
>
> `drc` 태그와 `prosody volume` 태그를 함께 사용할 경우 중첩 태그에 대한 표준 XML 규칙을 사용하세요.

부드러운 어투

<amazon:effect phonation="soft">

현재 TTS 형식만 이 태그를 지원합니다.

<amazon:effect phonation="soft">입력 텍스트를 softer-than-normal 음성으로 말하도록 지정하려면 태그를 사용하세요.

이때 다음 구문을 사용합니다.

```
<amazon:effect phonation="soft">text</amazon:effect>
```

예를 들어, 이 태그에서 Matthew 음성을 다음과 같은 방식으로 사용할 수 있습니다.

```
<speak>
     This is Matthew speaking in my normal voice. <amazon:effect phonation="soft">This
     is Matthew speaking in my softer voice.</amazon:effect>
</speak>
```

음색 제어

vocal-tract-length<아마존:이펙트 >

현재 TTS 형식만 이 태그를 지원합니다.

음색은 음의 고저가 같을 때도 음성 간의 차이를 알 수 있는 특징입니다. 음색을 내는 가장 중요한 생리적 특성 중 하나는 성대의 길이이며, 성대주름에서 입술 가장자리까지의 공기 통로입니다.

Amazon Polly에서 스피치의 음색을 조절하려면 `vocal-tract-length` 태그를 사용합니다. 이 태그는 말하는 사람의 성도 길이를 변경하여 다르게 들리게 합니다. `vocal-tract-length`을(를) 늘리면 말하는 사람의 음성이 더 커집니다. 반대로 줄이면 음성이 작아집니다. 이 태그는 Amazon Polly Text-to-Speech 포트폴리오에 있는 모든 음성에 사용할 수 있습니다.

음색을 변경하려면 다음 값을 사용합니다.

- +n% 또는 -n%: 현재 음성의 현재 음색 레벨을 상대적 백분율로 변경하여 성도 길이를 조정합니다. 예를 들어 +4% 또는 -2% 등으로 지정할 수 있습니다. 유효한 값은 +100%부터 -50%까지입니다. 이 범위를 벗어나는 값은 잘립니다. 예를 들어 +111%는 +100%와 동일하게 적용되고 -60%은 -50%와 동일하게 적용됩니다.
- n%: 현재 음성 성도 길이의 절대 백분율로 성도 길이를 변경합니다. 예를 들어 110% 또는 75% 등으로 지정합니다. 절대값 110%는 상대값 +10%와 동일합니다. 절대값 100%는 현재 음성의 기본값과 동일합니다.

다음 예는 성도 길이를 변경하여 음색을 변경하는 방법을 보여줍니다.

```
<speak>
     This is my original voice, without any modifications. <amazon:effect vocal-tract-
length="+15%">
     Now, imagine that I am much bigger. </amazon:effect> <amazon:effect vocal-tract-
length="-15%">
```

```
     Or, perhaps you prefer my voice when I'm very small. </amazon:effect> You can also
 control the
     timbre of my voice by making minor adjustments. <amazon:effect vocal-tract-
length="+10%">
     For example, by making me sound just a little bigger. </
amazon:effect><amazon:effect
     vocal-tract-length="-10%"> Or, making me sound only somewhat smaller. </
amazon:effect>
</speak>
```

여러 태그 결합

vocal-tract-length 태그와 Amazon Polly에서 지원하는 다른 SSML 태그를 결합할 수 있습니다. 음색(성도 길이)와 음의 고저는 서로 밀접하기 연결되기 때문에 vocal-tract-length 태그와 <prosody pitch> 태그를 함께 사용하여 최적의 결과를 얻을 수 있습니다. 가장 사실적인 음성을 만들어 내기 위해 두 태그의 변경 백분율을 다르게 사용하는 것이 좋습니다. 원하는 결과를 얻기 위해 다양한 조합을 사용해 보세요.

다음 예제는 태그를 결합하는 방법을 보여줍니다.

```
<speak>
     The pitch and timbre of a person's voice are connected in human speech.
     <amazon:effect vocal-tract-length="-15%"> If you are going to reduce the vocal
 tract length,
     </amazon:effect><amazon:effect vocal-tract-length="-15%"> <prosody pitch="+20%">
 you
     might consider increasing the pitch, too. </prosody></amazon:effect>
     <amazon:effect vocal-tract-length="+15%"> If you choose to lengthen the vocal
 tract,
     </amazon:effect> <amazon:effect vocal-tract-length="+15%"> <prosody pitch="-10%">
     you might also want to lower the pitch. </prosody></amazon:effect>
</speak>
```

속삭임 기능

<amazon:effect name="whispered">

현재 TTS 형식만 이 태그를 지원합니다.

이 태그는 입력 텍스트를 일반 스피치보다 속삭이는 음성으로 말해야 함을 나타냅니다. 이 태그는 Amazon Polly Text-to-Speech 포트폴리오에 있는 모든 음성과 함께 사용할 수 있습니다.

이 방식은 다음 구문을 사용합니다.

```
<amazon:effect name="whispered">text</amazon:effect>
```

예:

```
<speak>
     <amazon:effect name="whispered">If you make any noise, </amazon:effect>
     she said, <amazon:effect name="whispered">they will hear us.</amazon:effect>
</speak>
```

이 경우 캐릭터가 말하는 합성된 스피치는 속삭이는 음성이 되고, "she said" 구는 선택한 Amazon Polly 음성의 합성된 일반 스피치로 읽혀집니다.

원하는 효과에 따라 운율 속도를 최대 10%까지 느리게 하여 "속삭임" 효과를 향상시킬 수 있습니다.

예:

```
<speak>
     When any voice is made to whisper, <amazon:effect name="whispered">
     <prosody rate="-10%">the sound is slower and quieter than normal speech
     </prosody></amazon:effect>
</speak>
```

속삭이는 음성에 대한 스피치 마크를 생성하는 경우 오디오 스트림은 속삭이는 음성도 포함하여 스피치 마크가 오디오 스트림과 일치하도록 해야 합니다.

어휘집 관리

발음 어휘를 사용하면 단어의 발음을 사용자 지정할 수 있습니다. Amazon Polly는 지역에 어휘집을 저장하는 데 사용할 수 있는 API 작업을 제공합니다. AWS 이렇게 하면 어휘가 특정 리전으로 한정됩니다. `SynthesizeSpeech` 작업을 사용하여 텍스트를 합성할 때 해당 리전의 어휘를 하나 이상 사용할 수 있습니다. 이렇게 하면 합성이 시작되기 전에 지정된 어휘가 입력 텍스트에 적용됩니다. 자세한 내용은 SynthesizeSpeech을(를) 참조하세요.

> **Note**
>
> 이러한 어휘는 발음 사전 사양(PLS) W3C 권장 사항을 준수해야 합니다. 자세한 내용은 W3C 웹 사이트의 발음 어휘 사양(PLS) 버전 1.0을 참조하세요.

스피치 합성 엔진에서 어휘를 사용하는 방법의 예는 다음과 같습니다.

- 흔히 사용되는 단어는 “g3t sm4rt”(get smart)와 같이 문자 대신 숫자를 사용하여 멋을 부리는 경우가 있습니다. 사람은 이런 단어를 제대로 읽을 수 있습니다. 하지만 TTS(Text-to-Speech) 엔진은 텍스트를 문자 그대로 읽고 이름을 철자 그대로 발음합니다. 여기에서 Amazon Polly의 어휘 기능을 활용하여 합성되는 스피치를 사용자 지정할 수 있습니다. 이 예제에서는 어휘에 있는 “g3t sm4rt”라는 단어의 별칭(get smart)을 지정할 수 있습니다.
- 텍스트에는 W3C와 같은 두문자어가 포함될 수 있습니다. 어휘를 사용하면 W3C라는 단어의 별칭을 정의하여 전체 형식(월드 와이드 웹 컨소시엄)으로 읽을 수 있습니다.

어휘를 통해 Amazon Polly에서 선택한 언어에 자주 사용되지 않는 단어를 발음하는 방식을 추가로 제어할 수 있습니다. 예를 들어 발음기호를 사용하여 발음을 지정할 수 있습니다. 자세한 내용은 W3C 웹 사이트의 발음 어휘 사양(PLS) 버전 1.0을 참조하세요.

주제

- 여러 어휘집 적용
- Amazon Polly 콘솔에서의 어휘집 관리
- 의 어휘집 관리 AWS CLI

여러 어휘집 적용

텍스트에 최대 5개의 어휘를 적용할 수 있습니다. 텍스트에 적용하는 어휘가 두 개 이상의 어휘에 동일한 자소가 나타나는 경우, 적용되는 순서에 따라 결과 스피치가 달라질 수 있습니다. “안녕하세요, 제 이름은 Bob입니다.”라는 텍스트를 예로 들어보겠습니다. 그리고 서로 다른 두 어휘의 어휘소가 동일한 Bob 자소를 사용한다고 해봅시다.

LexA

```
<lexeme>
   <grapheme>Bob</grapheme>
   <alias>Robert</alias>
</lexeme>
```

LexB

```
<lexeme>
   <grapheme>Bob</grapheme>
   <alias>Bobby</alias>
</lexeme>
```

어휘가 LexA, LexB 순으로 나열되어 있는 경우 합성된 스피치는 “안녕하세요, 제 이름은 Robert입니다.”가 됩니다. 어휘가 LexB, LexA 순으로 나열되어 있는 경우 합성된 스피치는 “안녕하세요, 제 이름은 Bobby입니다.”가 됩니다.

Example - LexB보다 LexA를 먼저 적용

```
aws polly synthesize-speech \
--lexicon-names LexA LexB \
--output-format mp3 \
--text 'Hello, my name is Bob' \
--voice-id Justin \
bobAB.mp3
```

스피치 출력: “안녕하세요, 제 이름은 Robert입니다.”

Example - LexA보다 LexB를 먼저 적용

```
aws polly synthesize-speech \
--lexicon-names LexB LexA \
--output-format mp3 \
```

```
--text 'Hello, my name is Bob' \
--voice-id Justin \
bobBA.mp3
```

스피치 출력: “안녕하세요, 제 이름은 Bobby입니다.”

Amazon Polly 콘솔을 사용하여 어휘를 적용하는 방법에 대한 자세한 내용은 콘솔에서 어휘 적용 (음성 합성)을 참조하세요.

Amazon Polly 콘솔에서의 어휘집 관리

Amazon Polly 콘솔을 사용하여 어휘를 업로드, 다운로드, 적용, 필터링 및 삭제할 수 있습니다. 다음 절차는 이러한 각 프로세스를 보여줍니다.

콘솔에 어휘집 업로드

발음 어휘를 사용하려면 먼저 업로드해야 합니다. 콘솔에는 어휘를 업로드할 수 있는 두 개의 위치가 있는데 이들은 텍스트 투 스피치 탭과 어휘 탭입니다.

다음 프로세스는 선택한 언어에서 자주 사용되지 않는 단어와 구문의 발음 방식을 사용자 지정할 때 사용할 수 있는 어휘를 추가하는 방법을 설명합니다.

어휘 탭에서 어휘 추가

1. AWS Management Console 로그인하고 https://console.aws.amazon.com/polly/ 에서 Amazon Polly 콘솔을 엽니다.
2. 어휘 탭을 선택합니다.
3. 어휘 업로드를 선택합니다.
4. 어휘 이름을 입력한 후 어휘 파일 선택을 사용하여 업로드할 어휘를 찾습니다. 확장자가 .pls 또는 .xml인 PLS 파일만 업로드할 수 있습니다.
5. 어휘 업로드를 선택합니다. 같은 이름의 어휘(.pls 또는 .xml 파일)가 이미 있는 경우 어휘를 업로드하면 기존 어휘를 덮어씁니다.

텍스트 음성 변환 탭에서 어휘를 추가하려면

1. AWS Management Console 로그인하고 https://console.aws.amazon.com/polly/ 에서 Amazon Polly 콘솔을 엽니다.
2. 텍스트 투 스피치 탭을 선택합니다.

3. 추가 설정을 확장하고 발음 사용자 지정을 켠 후 어휘 업로드를 선택합니다.
4. 어휘 이름을 입력한 후 어휘 파일 선택을 사용하여 업로드할 어휘를 찾습니다. 확장자가 .pls 또는 .xml인 PLS 파일만 사용할 수 있습니다.
5. 어휘 업로드를 선택합니다. 이름이 같은 어휘(.pls 또는 .xml 파일)가 이미 있는 경우 어휘를 업로드하면 기존 어휘를 덮어씁니다.

콘솔에서 어휘 적용 (음성 합성)

다음 절차는 “W3C”를 “월드 와이드 웹 컨소시엄”으로 대체하는 `W3c.pls` 어휘를 적용하여 입력 텍스트에 어휘를 적용하는 방법을 보여줍니다. 텍스트에 여러 어휘를 적용하면 하향식 순서로 적용되며 첫 번째 일치 항목이 이후 일치 항목보다 우선합니다. 어휘에 지정된 언어가 선택한 언어와 동일한 경우에만 텍스트에 어휘가 적용됩니다.

일반 텍스트 또는 SSML 입력에 어휘를 적용할 수 있습니다.

Example - W3C.pls 어휘 적용

이 연습에 필요한 어휘를 생성하려면 오퍼레이션 사용 PutLexicon 을 참조하세요. 일반 텍스트 편집기를 사용하여 주제 상단에 표시된 W3C.pls 어휘를 생성합니다. 이 파일을 어디에 저장했는지 기억해 두세요.

W3C.pls 어휘를 입력에 적용

이 예제에서는 “W3C”를 “월드 와이드 웹 컨소시엄”으로 대체하는 어휘를 소개합니다. 이 연습의 결과를 미국 영어와 다른 언어의 연습의 콘솔에서 SSML 사용과 비교하세요.

1. AWS Management Console 로그인하고 https://console.aws.amazon.com/polly/ 에서 Amazon Polly 콘솔을 엽니다.
2. 다음 중 하나를 수행합니다.
   - SSML을 비활성화한 후 이 텍스트를 텍스트 입력 상자에 입력하거나 붙여 넣습니다.

     ```
     He was caught up in the game.
     In the middle of the 10/3/2014 W3C meeting
     he shouted, "Score!" quite loudly.
     ```

   - SSML을 활성화한 후 이 텍스트를 텍스트 입력 상자에 입력하거나 붙여 넣습니다.

     ```
     <speak>He wasn't paying attention.<break time="1s"/>
     ```

```
In the middle of the 10/3/2014 W3C meeting
he shouted, "Score!" quite loudly.</speak>
```

3. 언어 목록에서 영어, 미국을 선택한 후 이 텍스트에 사용할 음성을 선택합니다.
4. 추가 설정을 확장하고 발음 사용자 지정을 활성화합니다.
5. 어휘 목록에서 `W3C (English, US)`를 선택합니다.

   `W3C (English, US)` 어휘가 목록에 없는 경우 어휘 업로드를 선택하여 업로드한 후 목록에서 선택합니다. 이 어휘를 생성하려면 오퍼레이션 사용 PutLexicon 을 참조하세요.
6. 스피치를 즉시 들으려면 듣기를 선택합니다.
7. 스피치를 파일에 저장하려면

   a. 다운로드를 선택합니다.

   b. 다른 파일 형식으로 변경하려면 음성 파일 형식 설정을 활성화하고 원하는 파일 형식을 선택한 후 다운로드를 선택합니다.

이전 단계를 반복하되 다른 언어를 선택하고 출력 차이를 확인합니다.

콘솔에서 어휘 목록 필터링

다음 절차는 선택한 언어의 어휘만 표시되도록 어휘 목록을 필터링하는 방법을 설명합니다.

언어별로 나열된 어휘 필터링

1. AWS Management Console 로그인하고 https://console.aws.amazon.com/polly/ 에서 Amazon Polly 콘솔을 엽니다.
2. 어휘 탭을 선택합니다.
3. 원하는 언어를 선택합니다.
4. 언어 목록에서 필터링할 언어를 선택합니다.

   목록에는 선택한 언어의 어휘만 표시됩니다.

콘솔에서 어휘집 다운로드

다음 프로세스는 하나 이상의 어휘를 다운로드하는 방법을 설명합니다. 파일에서 어휘집 항목을 추가, 제거 또는 수정한 다음 다시 업로드하여 어휘를 유지할 수 있습니다. up-to-date

하나 이상의 어휘를 다운로드하려면

1. AWS Management Console 로그인하고 https://console.aws.amazon.com/polly/ 에서 Amazon Polly 콘솔을 엽니다.
2. 어휘 탭을 선택합니다.
3. 다운로드할 어휘를 선택합니다.
   a. 단일 어휘를 다운로드하려면 목록에서 이름을 선택합니다.
   b. 여러 어휘를 단일 압축 아카이브 파일로 다운로드하려면 다운로드할 목록의 각 항목 옆의 확인란을 선택합니다.
4. 다운로드를 선택합니다.
5. 어휘를 다운로드할 폴더를 엽니다.
6. 저장을 선택합니다.

콘솔에서 어휘집 삭제하기

어휘를 삭제하려면

다음 프로세스는 어휘를 삭제하는 방법을 설명합니다. 어휘를 삭제한 후 다시 추가해야 다시 사용할 수 있습니다. 개별 어휘 옆의 확인란을 선택하여 하나 이상의 어휘를 동시에 삭제할 수 있습니다.

1. AWS Management Console 로그인하고 https://console.aws.amazon.com/polly/ 에서 Amazon Polly 콘솔을 엽니다.
2. 어휘 탭을 선택합니다.
3. 목록에서 삭제할 어휘를 하나 이상 선택합니다.
4. 삭제를 선택합니다.
5. 확인 텍스트를 입력한 후 삭제를 선택하여 리전에서 어휘를 제거하거나 취소를 선택하여 해당 어휘를 보관합니다.

의 어휘집 관리 AWS CLI

다음 항목에서는 발음 어휘를 관리하는 데 필요한 AWS CLI 명령을 다룹니다.

주제

- 오퍼레이션 사용 PutLexicon

- GetLexicon 오퍼레이션 사용
- ListLexicons 오퍼레이션 사용
- DeleteLexicon 오퍼레이션 사용

오퍼레이션 사용 PutLexicon

Amazon Polly를 사용하면 계정의 특정 AWS 지역에 발음 PutLexicon 어휘를 저장하는 데 사용할 수 있습니다. 이렇게 하면 서비스에서 텍스트 합성을 시작하기 전에 적용하려는 저장된 어휘를 SynthesizeSpeech 요청에 하나 이상 지정할 수 있습니다. 자세한 정보는 어휘집 관리을 참조하세요.

이 섹션에서는 예제 어휘와 이를 저장하고 테스트하기 위한 step-by-step 지침을 제공합니다.

> **Note**
>
> 이러한 어휘는 발음 어휘 사양(PLS) W3C 권장 사항을 준수해야 합니다. 자세한 내용은 W3C 웹 사이트의 발음 어휘 사양(PLS) 버전 1.0을 참조하세요.

예 1: 어휘가 하나인 어휘집

다음과 같은 W3C PLS 준수 어휘를 고려해 보세요.

```
<?xml version="1.0" encoding="UTF-8"?>
<lexicon version="1.0"
      xmlns="http://www.w3.org/2005/01/pronunciation-lexicon"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.w3.org/2005/01/pronunciation-lexicon
        http://www.w3.org/TR/2007/CR-pronunciation-lexicon-20071212/pls.xsd"
      alphabet="ipa"
      xml:lang="en-US">
  <lexeme>
    <grapheme>W3C</grapheme>
    <alias>World Wide Web Consortium</alias>
  </lexeme>
</lexicon>
```

유의할 사항:

- `<lexicon>` 요소에 지정된 두 가지 속성:

- `xml:lang` 속성은 어휘가 적용되는 언어 코드와 `en-US`를 지정합니다. `SynthesizeSpeech` 호출에서 지정한 음성과 동일한 언어 코드(en-US)가 있는 경우 Amazon Polly에서 이 예제 어휘를 사용할 수 있습니다.

  > **Note**
  >
  > `DescribeVoices` 작업을 사용하여 음성과 관련된 언어 코드를 찾을 수 있습니다.

- `alphabet` 속성은 `IPA`를 지정합니다. 이는 발음에 국제 발음기호(IPA) 알파벳을 사용한다는 뜻입니다. IPA는 발음을 표기하기 위한 알파벳입니다. Amazon Polly는 확장 음성 평가 방법 발음기호(X-SAMPA)도 지원합니다.

- `<lexeme>` 요소는 `<grapheme>`(즉, 단어의 텍스트 표현)와 `<alias>` 사이의 매핑을 설명합니다.

이 어휘를 테스트하려면 다음을 수행합니다.

1. 어휘를 `example.pls`로 저장합니다.
2. `put-lexicon` AWS CLI 명령을 실행하여 us-east-2 지역에 어휘집 (이름 포함`w3c`) 을 저장합니다.

```
aws polly put-lexicon \
--name w3c \
--content file://example.pls
```

3. `synthesize-speech` 명령을 실행하여 샘플 텍스트를 오디오 스트림(`speech.mp3`)에 합성하고, 옵션으로 `lexicon-name` 파라미터를 지정합니다.

```
aws polly synthesize-speech \
--text 'W3C is a Consortium' \
--voice-id Joanna \
--output-format mp3 \
--lexicon-names="w3c" \
speech.mp3
```

4. 결과물 speech.mp3를 재생하면 텍스트의 W3C라는 단어가 월드 와이드 웹 컨소시엄으로 대체된 것을 확인할 수 있습니다.

위의 예제 어휘에서는 별칭을 사용합니다. 어휘에 언급된 IPA 알파벳은 사용되지 않습니다. 다음 어휘는 IPA 알파벳이 있는 <phoneme> 요소를 사용하여 표기 발음을 지정합니다.

```
<?xml version="1.0" encoding="UTF-8"?>
<lexicon version="1.0"
      xmlns="http://www.w3.org/2005/01/pronunciation-lexicon"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.w3.org/2005/01/pronunciation-lexicon
        http://www.w3.org/TR/2007/CR-pronunciation-lexicon-20071212/pls.xsd"
      alphabet="ipa"
      xml:lang="en-US">
  <lexeme>
    <grapheme>pecan</grapheme>
    <phoneme>p##k##n</phoneme>
  </lexeme>
</lexicon>
```

동일한 단계를 수행하여 이 어휘를 테스트합니다. “피칸”이라는 단어가 포함된 입력 텍스트(예: “피칸 파이는 맛있어요”)를 지정합니다.

예 2: 어휘가 여러 개 있는 어휘집

이 예제에서 어휘를 통해 지정하는 어휘소는 합성을 위한 입력 텍스트에만 적용됩니다. 다음 어휘소를 고려해 보세요.

```
<?xml version="1.0" encoding="UTF-8"?>
<lexicon version="1.0"
      xmlns="http://www.w3.org/2005/01/pronunciation-lexicon"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.w3.org/2005/01/pronunciation-lexicon
        http://www.w3.org/TR/2007/CR-pronunciation-lexicon-20071212/pls.xsd"
      alphabet="ipa" xml:lang="en-US">

  <lexeme>
    <grapheme>W3C</grapheme>
    <alias>World Wide Web Consortium</alias>
  </lexeme>
  <lexeme>
    <grapheme>W3C</grapheme>
```

```
    <alias>WWW Consortium</alias>
  </lexeme>
  <lexeme>
    <grapheme>Consortium</grapheme>
    <alias>Community</alias>
  </lexeme>
</lexicon>
```

어휘에는 세 개의 어휘소가 지정되며, 그 중 두 개의 어휘소는 다음과 같이 W3C 자소의 별칭을 정의합니다.

- 첫 번째 <lexeme> 요소는 별칭(월드 와이드 웹 컨소시엄)을 정의합니다.
- 두 번째 <lexeme>는 대체 별칭(WWW 컨소시엄)을 정의합니다.

Amazon Polly는 어휘의 특정 자소를 첫 번째 대체어로 대체합니다.

세 번째 <lexeme>는 컨소시엄이라는 단어의 대체어(커뮤니티)를 정의합니다.

먼저 이 어휘를 테스트해 보겠습니다. 다음 샘플 텍스트를 오디오 파일(speech.mp3)에 합성하고 SynthesizeSpeech를 호출할 때 어휘를 지정한다고 가정해 보겠습니다.

```
The W3C is a Consortium
```

SynthesizeSpeech는 먼저 다음과 같이 어휘를 적용합니다.

- 첫 번째 어휘소에 따라 W3C라는 단어를 월드 와이드 웹 컨소시엄으로 수정합니다. 수정된 텍스트는 다음과 같이 나타납니다.

  ```
  The World Wide Web Consortium is a Consortium
  ```

- 세 번째 어휘소에 정의된 별칭은 원본 텍스트의 일부인 컨소시엄이라는 단어에만 적용되며, 그 결과 다음과 같은 텍스트가 나타납니다.

  ```
  The World Wide Web Consortium is a Community.
  ```

다음과 같이 사용하여 이를 테스트할 수 있습니다. AWS CLI

1. 어휘를 example.pls로 저장합니다.
2. put-lexicon 명령을 실행하여 이름이 w3c인 어휘를 us-east-2 리전에 저장합니다.

```
aws polly put-lexicon \
--name w3c \
--content file://example.pls
```

3. list-lexicons 명령을 실행하여 w3c 어휘가 반환된 어휘 목록에 있는지 확인합니다.

```
aws polly list-lexicons
```

4. synthesize-speech 명령을 실행하여 샘플 텍스트를 오디오 파일(speech.mp3)에 합성하고 옵션으로 lexicon-name 파라미터를 지정합니다.

```
aws polly synthesize-speech \
--text 'W3C is a Consortium' \
--voice-id Joanna \
--output-format mp3 \
--lexicon-names="w3c" \
speech.mp3
```

5. 결과물인 speech.mp3 파일을 재생하여 합성된 음성에 변경된 텍스트가 적용되는지 확인합니다.

예 3: 여러 어휘집 지정

SynthesizeSpeech를 호출할 때 여러 어휘를 지정할 수 있습니다. 이 경우 왼쪽에서 오른쪽 순으로 지정된 어휘 중 첫 번째 어휘가 이전의 모든 어휘보다 우선합니다.

다음 두 어휘를 고려해 보세요. 참고로 어휘마다 동일한 W3C 자소에 대해 서로 다른 별칭이 설명되어 있습니다.

- 어휘 1: w3c.pls

```
<?xml version="1.0" encoding="UTF-8"?>
<lexicon version="1.0"
      xmlns="http://www.w3.org/2005/01/pronunciation-lexicon"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.w3.org/2005/01/pronunciation-lexicon
        http://www.w3.org/TR/2007/CR-pronunciation-lexicon-20071212/pls.xsd"
      alphabet="ipa" xml:lang="en-US">
  <lexeme>
    <grapheme>W3C</grapheme>
    <alias>World Wide Web Consortium</alias>
```

```
  </lexeme>
</lexicon>
```

- 어휘 2: w3cAlternate.pls

```
<?xml version="1.0" encoding="UTF-8"?>
<lexicon version="1.0"
      xmlns="http://www.w3.org/2005/01/pronunciation-lexicon"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.w3.org/2005/01/pronunciation-lexicon
        http://www.w3.org/TR/2007/CR-pronunciation-lexicon-20071212/pls.xsd"
      alphabet="ipa" xml:lang="en-US">

  <lexeme>
    <grapheme>W3C</grapheme>
    <alias>WWW Consortium</alias>
  </lexeme>
</lexicon>
```

이들 어휘를 각각 w3c 및 w3cAlternate로 저장한다고 가정해 보겠습니다. SynthesizeSpeech 호출에서 어휘를 순서대로(w3c 다음에 w3cAlternate) 지정하는 경우 첫 번째 어휘에 정의된 W3C의 별칭이 두 번째 어휘보다 우선합니다. 어휘를 테스트하려면 다음을 수행합니다.

1. w3c.pls 및 w3cAlternate.pls 파일에 어휘를 로컬로 저장합니다.
2. 명령을 사용하여 이러한 어휘를 업로드합니다. put-lexicon AWS CLI

   - w3c.pls 어휘를 업로드하고 w3c로 저장합니다.

     ```
     aws polly put-lexicon \
     --name w3c \
     --content file://w3c.pls
     ```

   - 서비스에 w3cAlternate.pls 어휘를 w3cAlternate로 업로드합니다.

     ```
     aws polly put-lexicon \
     --name w3cAlternate \
     --content file://w3cAlternate.pls
     ```

3. `synthesize-speech` 명령을 실행하여 샘플 텍스트를 오디오 스트림(`speech.mp3`)에 합성하고 `lexicon-name` 파라미터를 사용하여 두 어휘를 모두 지정합니다.

```
aws polly synthesize-speech \
--text 'PLS is a W3C recommendation' \
--voice-id Joanna \
--output-format mp3 \
--lexicon-names '["w3c","w3cAlternative"]' \
speech.mp3
```

4. 결과물인 `speech.mp3`를 테스트합니다. 다음과 같이 표시되어야 합니다.

```
PLS is a World Wide Web Consortium recommendation
```

API용 추가 코드 샘플 PutLexicon

- Java 샘플: PutLexicon
- Python(Boto3) 샘플: PutLexicon

GetLexicon 오퍼레이션 사용

Amazon Polly는 특정 리전의 계정에 저장한 발음 어휘의 내용을 검색하는 GetLexicon API 작업을 제공합니다.

다음 `get-lexicon` AWS CLI 명령은 `example` 어휘의 내용을 검색합니다.

```
aws polly get-lexicon \
--name example
```

계정에 어휘를 아직 저장하지 않은 경우 `PutLexicon` 작업을 사용하여 어휘를 저장할 수 있습니다. 자세한 내용은 오퍼레이션 사용 PutLexicon 을(를) 참조하세요.

다음은 응답 예입니다. 응답은 어휘 내용 외에도 어휘가 적용되는 언어 코드, 어휘에 정의된 어휘소의 수, 리소스의 Amazon 리소스 이름(ARN), 어휘의 크기(바이트) 등의 메타데이터를 반환합니다. `LastModified` 값은 Unix 타임스탬프입니다.

```
{
    "Lexicon": {
```

```
        "Content": "lexicon content in plain text PLS format",
        "Name": "example"
    },
    "LexiconAttributes": {
        "LanguageCode": "en-US",
        "LastModified": 1474222543.989,
        "Alphabet": "ipa",
        "LexemesCount": 1,
        "LexiconArn": "arn:aws:polly:us-east-2:account-id:lexicon/example",
        "Size": 495
    }
}
```

API용 추가 코드 샘플 GetLexicon

- Java 샘플: GetLexicon
- Python(Boto3) 샘플: GetLexicon

ListLexicons 오퍼레이션 사용

Amazon Polly는 특정 지역의 계정에 있는 발음 사전 목록을 가져오는 데 사용할 수 있는 ListLexicons API 작업을 제공합니다. AWS 다음 AWS CLI 호출은 us-east-2 지역의 사용자 계정에 있는 어휘를 나열합니다.

```
aws polly list-lexicons
```

다음은 예제 응답이며, 이름이 `w3c` 및 `tomato`인 두 개의 어휘를 보여줍니다. 각 어휘에 대해 응답은 어휘가 적용되는 언어 코드, 어휘에 정의된 어휘소의 수, 크기(바이트) 등과 같은 메타데이터를 반환합니다. 언어 코드는 어휘에 정의된 어휘소가 적용되는 언어와 로케일을 설명합니다.

```
{
    "Lexicons": [
        {
            "Attributes": {
                "LanguageCode": "en-US",
                "LastModified": 1474222543.989,
                "Alphabet": "ipa",
                "LexemesCount": 1,
                "LexiconArn": "arn:aws:polly:aws-region:account-id:lexicon/w3c",
```

```
                "Size": 495
            },
            "Name": "w3c"
        },
        {
            "Attributes": {
                "LanguageCode": "en-US",
                "LastModified": 1473099290.858,
                "Alphabet": "ipa",
                "LexemesCount": 1,
                "LexiconArn": "arn:aws:polly:aws-region:account-id:lexicon/tomato",
                "Size": 645
            },
            "Name": "tomato"
        }
    ]
}
```

API용 추가 코드 샘플 ListLexicon

- Java 샘플: ListLexicons
- Python(Boto3) 샘플: ListLexicon

DeleteLexicon 오퍼레이션 사용

Amazon Polly는 계정의 특정 AWS 지역에서 발음 어휘를 삭제하는 DeleteLexicon API 작업을 제공합니다. 다음은 지정된 어휘를 AWS CLI 삭제합니다.

다음 AWS CLI 예제는 유닉스, 리눅스, macOS용으로 포맷되었습니다. Windows의 경우 각 줄 끝에 있는 백슬래시(\) Unix 연속 문자를 캐럿(^)으로 바꿉니다. 입력 텍스트는 큰 따옴표(")로 감싸고 내부 태그에는 작은 따옴표(')를 사용합니다.

```
aws polly delete-lexicon \
--name example
```

API용 추가 코드 샘플 DeleteLexicon

- Java 샘플: DeleteLexicon
- Python(Boto3) 샘플: DeleteLexicon

긴 오디오 파일 생성

대량 텍스트 구절의 TTS 파일을 생성하려면 Amazon Polly의 비동기식 합성 기능을 사용합니다. 이 기능에서는 다음 세 가지 SpeechSynthesisTask API를 사용합니다.

- StartSpeechSynthesisTask: 새 합성 작업을 시작합니다.
- GetSpeechSynthesisTask: 이전에 제출된 합성 작업에 대한 세부 정보를 반환합니다.
- ListSpeechSynthesisTasks: 제출된 합성 작업을 모두 나열합니다.

SynthesizeSpeech 작업은 대부분의 경우 지연 시간이 상대적으로 짧은 실시간에 가깝게 오디오를 생성합니다. 이를 위해서 이 작업에서는 3000자만 합성할 수 있습니다.

Amazon Polly의 비동기식 합성 기능을 사용하면 문서를 합성하고 반환하는 방식을 변경함으로써 더 큰 텍스트 문서를 처리하는 어려움을 해결할 수 있습니다. StartSpeechSynthesisTask를 사용하여 입력 텍스트를 제출하고 합성 요청을 하면 Amazon Polly는 요청을 대기열에 넣은 다음 시스템 리소스를 사용할 수 있게 되면 바로 백그라운드에서 요청을 비동기적으로 처리합니다. 그러 다음 Amazon Polly는 결과 스피치 또는 스피치 마크 스트림(필수)을 Amazon Simple Storage Service(Amazon S3) 버킷에 직접 업로드하고(선택) SNS 주제를 통해 완성된 파일의 가용성에 대해 알려줍니다.

이러한 방식으로 실시간에 가까운 처리를 제외한 모든 기능을 최대 100,000개의 요금이 청구되는 문자(또는 총 200,000자) 길이의 텍스트에 사용할 수 있습니다.

이 방법을 사용하여 문서를 합성하려면 오디오 파일을 저장할 수 있는 쓰기 가능한 Amazon S3 버킷이 있어야 합니다. 옵션 SNS 주제 식별자를 제공하여 합성화된 오디오 준비가 완료될 때 알림을 받을 수 있습니다. 합성 작업이 완료되면 Amazon Polly에서 해당 주제에 대한 메시지를 게시합니다. 이 메시지에는 합성 작업이 실패한 경우 유용한 오류 정보가 포함될 수도 있습니다. 이렇게 하려면 합성 작업을 생성하는 사용자도 SNS 주제에 게시할 수 있어야 합니다. SNS 주제를 생성하고 구독하는 방법에 대한 자세한 내용은 Amazon SNS 문서를 참조하세요.

암호화(Encryption)

원하는 경우 출력 파일을 S3 버킷에 암호화된 형식으로 저장할 수 있습니다. 이렇게 하려면 제공되는 가장 강력한 블록 암호 중 하나인 256비트 Advanced Encryption Standard(AES-256)를 사용하여 Amazon S3 버킷 암호화를 활성화합니다.

주제

- 비동기 합성을 위한 IAM 정책 설정

- 콘솔에서 긴 오디오 파일 생성
- 에서 긴 오디오 파일 만들기 AWS CLI

비동기 합성을 위한 IAM 정책 설정

비동기식 합성 기능을 사용하려면 다음을 허용하는 IAM 정책이 필요합니다.

- 새로운 Amazon Polly 작업 사용
- 출력 S3 버킷에 쓰기
- 상태 SNS 주제에 게시[선택]

다음 정책은 비동기식 합성에 필요한 권한만 부여하며 IAM 사용자에 연결될 수 있습니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "polly:StartSpeechSynthesisTask",
        "polly:GetSpeechSynthesisTask",
        "polly:ListSpeechSynthesisTasks"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::bucket-name/*"
    },
    {
      "Effect": "Allow",
      "Action": "sns:Publish",
      "Resource": "arn:aws:sns:region:account:topic"
    }
  ]
}
```

콘솔에서 긴 오디오 파일 생성

Amazon Polly 콘솔을 사용하면 AWS CLI와 함께 사용할 수 있는 동일한 기능으로 비동기식 합성을 사용하여 긴 스피치를 생성할 수 있습니다. 이는 다른 합성과 같이 텍스트 투 스피치 탭을 사용하여 수행됩니다.

다른 비동기식 합성 기능도 콘솔을 통해 사용할 수 있습니다. S3 합성 작업 탭은 `ListSpeechSynthesisTasks` 기능을 반영하므로 S3 버킷에 저장된 모든 작업을 표시하고 사용자가 원하는 경우 해당 작업을 필터링할 수 있습니다. `GetSpeechSynthesisTask` 기능 반영으로 인해 특정 단일 작업을 클릭하면 해당 세부 정보가 표시됩니다.

Amazon Polly 콘솔을 사용하여 큰 텍스트를 합성하려면

1. AWS Management Console 로그인하고 https://console.aws.amazon.com/polly/ 에서 Amazon Polly 콘솔을 엽니다.
2. 텍스트 투 스피치 탭을 선택합니다. 필요한 경우 엔진으로 롱폼을 선택합니다.
3. SSML을 비활성화 또는 활성화한 상태에서 입력란에 텍스트를 입력하거나 붙여 넣습니다.
4. 해당 텍스트에 대한 언어, 리전 및 음성을 선택합니다.
5. S3에 저장을 선택합니다.

   > **Note**
   >
   > 텍스트 길이가 실시간 `SynthesizeSpeech` 작업에 대한 3,000자 제한을 초과하는 경우 다운로드 및 수신 옵션 모두 회색으로 표시됩니다.

6. 콘솔에 양식이 열리고 출력 파일을 저장할 위치를 선택할 수 있습니다.

   a. 대상 Amazon S3 버킷의 이름을 채웁니다.

   b. 경우에 따라 출력의 접두사 키를 채웁니다.

      > **Note**
      >
      > 출력 S3 버킷은 쓰기 가능해야 합니다.

   c. 합성 작업이 완료될 때 알림을 받으려면 옵션 SNS 주제 식별자를 제공합니다.

> **Note**
>
> 현재 콘솔 사용자가 게시할 때 이 옵션을 사용하려면 SNS가 열려 있어야 합니다. 자세한 내용은 Amazon Simple Notification Service(SNS)를 참조하세요.

d. S3에 저장을 선택합니다.

스피치 합성 작업에 대한 정보를 검색하려면

1. 콘솔에서 S3 합성 작업 탭을 선택합니다.
2. 작업은 날짜 순서대로 표시됩니다. 상태를 기준으로 작업을 필터링하려면 모든 상태를 선택한 다음 사용할 상태를 선택합니다.
3. 특정 작업의 세부 정보를 보려면 연결된 작업 ID를 선택합니다.

에서 긴 오디오 파일 만들기 AWS CLI

Amazon Polly 비동기식 합성 기능은 다음 세 가지 SpeechSynthesisTask API를 사용하여 대량의 텍스트 작업을 수행합니다.

- `StartSpeechSynthesisTask`: 새 합성 작업을 시작합니다.
- `GetSpeechSynthesisTask`: 이전에 제출된 합성 작업에 대한 세부 정보를 반환합니다.
- `ListSpeechSynthesisTasks`: 제출된 합성 작업을 모두 나열합니다.

대량의 텍스트 합성(**`StartSpeechSynthesisTask`**)

실시간 `SynthesizeSpeech`를 사용하여 생성할 수 있는 파일보다 큰 오디오 파일을 생성하려는 경우 `StartSpeechSynthesisTask` 작업을 사용합니다. `StartSpeechSynthesisTask`에는 `SynthesizeSpeech` 작업에 필요한 인수 이외에 Amazon S3 버킷 이름도 필요합니다. 작업에 대한 상태 알림을 받으려는 경우 다른 두 개의 옵션 인수인 출력 파일에 대한 키 접두사와 SNS 주제에 대한 ARN도 사용 가능합니다.

- `OutputS3BucketName`: 합성을 업로드해야 하는 Amazon S3 버킷의 이름입니다. 이 버킷은 Amazon Polly 서비스와 동일한 리전에 있어야 합니다. 또한 호출하는 데 사용되는 IAM 사용자는 해당 버킷에 액세스할 수 있어야 합니다. [필수]

- `OutputS3KeyPrefix`: 출력 파일의 키 접두사입니다. 버킷의 사용자 지정 디렉터리 종류의 키에 출력 스피치 파일을 저장하려는 경우 이 파라미터를 사용합니다. [선택]
- `SnsTopicArn`: 작업 상태에 대한 알림을 받으려면 경우 사용할 SNS 주제 ARN입니다. 이 SNS 주제는 Amazon Polly 서비스와 동일한 리전에 있어야 합니다. 또한 호출하는 데 사용되는 IAM 사용자는 해당 주제에 액세스할 수 있어야 합니다. [선택]

예를 들어, 다음 예제를 사용하여 미국 동부 (오하이오) 지역에서 `start-speech-synthesis-task` AWS CLI 명령을 실행할 수 있습니다.

다음 AWS CLI 예제는 유닉스, 리눅스, macOS용으로 포맷되었습니다. Windows의 경우 각 줄 끝에 있는 백슬래시(\) Unix 연속 문자를 캐럿(^)으로 바꿉니다. 입력 텍스트는 큰 따옴표(")로 감싸고 내부 태그에는 작은 따옴표(')를 사용합니다.

```
aws polly start-speech-synthesis-task \
  --region us-east-2 \
  --endpoint-url "https://polly.us-east-2.amazonaws.com/" \
  --output-format mp3 \
  --output-s3-bucket-name your-bucket-name \
  --output-s3-key-prefix optional/prefix/path/file \
  --voice-id Joanna \
  --text file://text_file.txt
```

이렇게 하면 다음과 비슷한 모양의 응답이 생성됩니다.

```
"SynthesisTask":
{
     "OutputFormat": "mp3",
     "OutputUri": "https://s3.us-east-2.amazonaws.com/your-bucket-name/optional/prefix/
path/file.<task_id>.mp3",
     "TextType": "text",
     "CreationTime": [..],
     "RequestCharacters": [..],
     "TaskStatus": "scheduled",
     "TaskId": [task_id],
     "VoiceId": "Joanna"
 }
```

`start-speech-synthesis-task` 작업은 다음과 같은 여러 새 필드를 반환합니다.

- `OutputUri`: 출력 스피치 파일의 위치입니다.
- `TaskId`: Amazon Polly에서 생성한 스피치 합성 작업의 고유 식별자입니다.
- `CreationTime`: 작업이 처음 제출된 시점의 타임스탬프입니다.
- `RequestCharacters`: 작업에서 요금이 청구되는 문자 수입니다.
- `TaskStatus`: 제출된 작업의 상태에 대한 정보를 제공합니다.

  작업이 제출될 때 초기 상태에서 `scheduled`를 표시합니다. Amazon Polly에서 작업 처리를 시작하면 상태가 `inProgress`로 변경되고 나중에 `completed` 또는 `failed`로 변경됩니다. 작업이 실패하면 or 작업을 호출할 때 오류 메시지가 반환됩니다. GetSpeechSynthesisTask ListSpeechSynthesisTasks

작업이 완료되면 `OutputUri`에서 지정된 위치에서 스피치 파일을 사용할 수 있습니다.

스피치 합성 작업에 대한 정보 검색

`GetSpeechSynthesisTask` 작업을 사용하여 오류, 상태 등의 작업 정보를 가져올 수 있습니다. 이를 위해서는 `task-id`에 의해 반환된 `StartSpeechSynthesisTask`가 필요합니다.

예를 들어, 다음 예제를 사용하여 `get-speech-synthesis-task` AWS CLI 명령을 실행할 수 있습니다.

```
aws polly get-speech-synthesis-task \
--region us-east-2 \
--endpoint-url "https:// polly.us-east-2.amazonaws.com/" \
--task-id task identifier
```

또한 `ListSpeechSynthesisTasks` 작업을 사용하여 현재 리전에서 실행한 스피치 합성 작업을 모두 나열할 수 있습니다.

예를 들어, 다음 예제를 사용하여 `list-speech-synthesis-tasks` AWS CLI 명령을 실행할 수 있습니다.

```
aws polly list-speech-synthesis-tasks \
--region us-east-2 \
--endpoint-url "https:// polly.us-east-2.amazonaws.com/"
```

코드 및 애플리케이션 예제

이 섹션에서는 Amazon Polly를 살펴보기 위해 사용할 수 있는 코드 샘플 및 예제 애플리케이션을 제공합니다.

주제

- 샘플 코드
- 애플리케이션의 예

샘플 코드 주제에는 프로그래밍 언어별로 정리된 코드 스니펫이 포함되어 있으며 다양한 Amazon Polly 기능에 대한 예제로 구분되어 있습니다. 예제 애플리케이션 주제에는 Amazon Polly를 탐색할 때 독립적으로 사용할 수 있는 프로그래밍 언어별로 구성된 애플리케이션이 포함되어 있습니다.

이러한 예를 사용하기 전에 Amazon Polly의 작동 방식에서 설명하는 단계를 먼저 읽고 아마존 Polly 시작하기에서 설명하는 단계를 수행하는 것이 좋습니다.

샘플 코드

이 주제에는 Amazon Polly를 탐색할 때 사용할 수 있는 다양한 기능에 대한 코드 샘플이 포함되어 있습니다.

프로그래밍 언어별 샘플 코드

- 자바 샘플
- Python 샘플

자바 샘플

다음 코드 샘플은 Java 기반 애플리케이션을 사용하여 Amazon Polly에서 다양한 작업을 수행하는 방법을 보여줍니다. 이 샘플은 완전한 예제는 아니지만 AWS SDK for Java를 사용하는 대규모 Java 애플리케이션에 포함될 수 있습니다.

코드 스니펫

- DeleteLexicon
- DescribeVoices
- GetLexicon

- ListLexicons
- PutLexicon
- StartSpeechSynthesisTask
- 스피치 마크
- SynthesizeSpeech

DeleteLexicon

다음 Java 코드 샘플은 Java 기반 애플리케이션을 사용하여 AWS 리전에 저장된 특정 어휘를 삭제하는 방법을 보여줍니다. 삭제된 어휘는 스피치 합성에 사용할 수 없으며 `GetLexicon` 또는 `ListLexicon` API를 사용하여 검색할 수도 없습니다.

이 작업에 대한 자세한 내용은 `DeleteLexicon` API 참조를 참조하세요.

```
package com.amazonaws.polly.samples;

import com.amazonaws.services.polly.AmazonPolly;
import com.amazonaws.services.polly.AmazonPollyClientBuilder;
import com.amazonaws.services.polly.model.DeleteLexiconRequest;

public class DeleteLexiconSample {
    private String LEXICON_NAME = "SampleLexicon";

    AmazonPolly client = AmazonPollyClientBuilder.defaultClient();

    public void deleteLexicon() {
        DeleteLexiconRequest deleteLexiconRequest = new
 DeleteLexiconRequest().withName(LEXICON_NAME);

        try {
            client.deleteLexicon(deleteLexiconRequest);
        } catch (Exception e) {
            System.err.println("Exception caught: " + e);
        }
    }
}
```

DescribeVoices

다음 Java 코드 예에서는 Java 기반 애플리케이션을 사용하여 스피치 합성을 요청할 때 사용할 수 있는 음성 목록을 생성하는 방법을 보여줍니다. 필요한 경우 언어 코드를 지정하여 사용 가능한 음성을 필터링할 수 있습니다. 예를 들어 en-US를 지정하면 사용 가능한 모든 미국 영어 음성 목록을 반환합니다.

이 작업에 대한 자세한 내용은 DescribeVoices API 참조를 참조하세요.

```
package com.amazonaws.polly.samples;

import com.amazonaws.services.polly.AmazonPolly;
import com.amazonaws.services.polly.AmazonPollyClientBuilder;
import com.amazonaws.services.polly.model.DescribeVoicesRequest;
import com.amazonaws.services.polly.model.DescribeVoicesResult;

public class DescribeVoicesSample {
    AmazonPolly client = AmazonPollyClientBuilder.defaultClient();

    public void describeVoices() {
        DescribeVoicesRequest allVoicesRequest = new DescribeVoicesRequest();
        DescribeVoicesRequest enUsVoicesRequest = new
 DescribeVoicesRequest().withLanguageCode("en-US");

        try {
            String nextToken;
            do {
                DescribeVoicesResult allVoicesResult =
 client.describeVoices(allVoicesRequest);
                nextToken = allVoicesResult.getNextToken();
                allVoicesRequest.setNextToken(nextToken);

                System.out.println("All voices: " + allVoicesResult.getVoices());
            } while (nextToken != null);

            do {
                DescribeVoicesResult enUsVoicesResult =
 client.describeVoices(enUsVoicesRequest);
                nextToken = enUsVoicesResult.getNextToken();
                enUsVoicesRequest.setNextToken(nextToken);

                System.out.println("en-US voices: " + enUsVoicesResult.getVoices());
            } while (nextToken != null);
```

```
        } catch (Exception e) {
            System.err.println("Exception caught: " + e);
        }
    }
}
```

GetLexicon

다음 Java 코드 샘플은 Java 기반 애플리케이션을 사용하여 지역에 저장된 특정 발음 어휘의 콘텐츠를 생성하는 방법을 보여줍니다. AWS

이 작업에 대한 자세한 내용은 GetLexicon API 참조를 참조하세요.

```
package com.amazonaws.polly.samples;

import com.amazonaws.services.polly.AmazonPolly;
import com.amazonaws.services.polly.AmazonPollyClientBuilder;
import com.amazonaws.services.polly.model.GetLexiconRequest;
import com.amazonaws.services.polly.model.GetLexiconResult;

public class GetLexiconSample {
    private String LEXICON_NAME = "SampleLexicon";

    AmazonPolly client = AmazonPollyClientBuilder.defaultClient();

    public void getLexicon() {
        GetLexiconRequest getLexiconRequest = new
 GetLexiconRequest().withName(LEXICON_NAME);

        try {
            GetLexiconResult getLexiconResult = client.getLexicon(getLexiconRequest);
            System.out.println("Lexicon: " + getLexiconResult.getLexicon());
        } catch (Exception e) {
            System.err.println("Exception caught: " + e);
        }
    }
}
```

ListLexicons

다음 Java 코드 샘플은 Java 기반 애플리케이션을 사용하여 지역에 저장된 발음 어휘의 목록을 생성하는 방법을 보여줍니다. AWS

이 작업에 대한 자세한 내용은 ListLexicons API 참조를 참조하세요.

```
package com.amazonaws.polly.samples;

import com.amazonaws.services.polly.AmazonPolly;
import com.amazonaws.services.polly.AmazonPollyClientBuilder;
import com.amazonaws.services.polly.model.LexiconAttributes;
import com.amazonaws.services.polly.model.LexiconDescription;
import com.amazonaws.services.polly.model.ListLexiconsRequest;
import com.amazonaws.services.polly.model.ListLexiconsResult;

public class ListLexiconsSample {
    AmazonPolly client = AmazonPollyClientBuilder.defaultClient();

    public void listLexicons() {
        ListLexiconsRequest listLexiconsRequest = new ListLexiconsRequest();

        try {
            String nextToken;
            do {
                ListLexiconsResult listLexiconsResult =
 client.listLexicons(listLexiconsRequest);
                nextToken = listLexiconsResult.getNextToken();
                listLexiconsRequest.setNextToken(nextToken);

                for (LexiconDescription lexiconDescription :
 listLexiconsResult.getLexicons()) {
                    LexiconAttributes attributes = lexiconDescription.getAttributes();
                    System.out.println("Name: " + lexiconDescription.getName()
                            + ", Alphabet: " + attributes.getAlphabet()
                            + ", LanguageCode: " + attributes.getLanguageCode()
                            + ", LastModified: " + attributes.getLastModified()
                            + ", LexemesCount: " + attributes.getLexemesCount()
                            + ", LexiconArn: " + attributes.getLexiconArn()
                            + ", Size: " + attributes.getSize());
                }
            } while (nextToken != null);
        } catch (Exception e) {
            System.err.println("Exception caught: " + e);
        }
    }
}
```

PutLexicon

다음 Java 코드 샘플은 Java 기반 애플리케이션을 사용하여 AWS 리전에 발음 어휘를 저장하는 방법을 보여줍니다.

이 작업에 대한 자세한 내용은 PutLexicon API 참조를 참조하세요.

```
package com.amazonaws.polly.samples;

import com.amazonaws.services.polly.AmazonPolly;
import com.amazonaws.services.polly.AmazonPollyClientBuilder;
import com.amazonaws.services.polly.model.PutLexiconRequest;

public class PutLexiconSample {
    AmazonPolly client = AmazonPollyClientBuilder.defaultClient();

    private String LEXICON_CONTENT = "<?xml version=\"1.0\" encoding=\"UTF-8\"?>" +
            "<lexicon version=\"1.0\" xmlns=\"http://www.w3.org/2005/01/pronunciation-
lexicon\" xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\" " +
            "xsi:schemaLocation=\"http://www.w3.org/2005/01/pronunciation-lexicon
 http://www.w3.org/TR/2007/CR-pronunciation-lexicon-20071212/pls.xsd\" " +
            "alphabet=\"ipa\" xml:lang=\"en-US\">" +
            "<lexeme><grapheme>test1</grapheme><alias>test2</alias></lexeme>" +
            "</lexicon>";
    private String LEXICON_NAME = "SampleLexicon";

    public void putLexicon() {
        PutLexiconRequest putLexiconRequest = new PutLexiconRequest()
                .withContent(LEXICON_CONTENT)
                .withName(LEXICON_NAME);

        try {
            client.putLexicon(putLexiconRequest);
        } catch (Exception e) {
            System.err.println("Exception caught: " + e);
        }
    }
}
```

StartSpeechSynthesisTask

다음 Java 코드 샘플은 Java 기반 애플리케이션을 사용하여 긴 스피치(최대 100,000 유료 문자)을 합성하고 Amazon S3 버킷에 직접 저장하는 방법을 보여줍니다.

자세한 내용은 StartSpeechSynthesisTask API 참조를 참조하세요.

```
package com.amazonaws.parrot.service.tests.speech.task;

import com.amazonaws.parrot.service.tests.AbstractParrotServiceTest;
import com.amazonaws.services.polly.AmazonPolly;
import com.amazonaws.services.polly.model.*;
import org.awaitility.Duration;

import java.util.concurrent.TimeUnit;

import static org.awaitility.Awaitility.await;

public class StartSpeechSynthesisTaskSample {

    private static final int SYNTHESIS_TASK_TIMEOUT_SECONDS = 300;
    private static final AmazonPolly AMAZON_POLLY_CLIENT =
 AmazonPollyClientBuilder.defaultClient();
    private static final String PLAIN_TEXT = "This is a sample text to be
 synthesized.";
    private static final String OUTPUT_FORMAT_MP3 = OutputFormat.Mp3.toString();
    private static final String OUTPUT_BUCKET = "synth-books-buckets";
    private static final String SNS_TOPIC_ARN = "arn:aws:sns:eu-
west-2:123456789012:synthesize-finish-topic";
    private static final Duration SYNTHESIS_TASK_POLL_INTERVAL = Duration.FIVE_SECONDS;
    private static final Duration SYNTHESIS_TASK_POLL_DELAY = Duration.TEN_SECONDS;

    public static void main(String... args) {
        StartSpeechSynthesisTaskRequest request = new StartSpeechSynthesisTaskRequest()
                .withOutputFormat(OUTPUT_FORMAT_MP3)
                .withText(PLAIN_TEXT)
                .withTextType(TextType.Text)
                .withVoiceId(VoiceId.Amy)
                .withOutputS3BucketName(OUTPUT_BUCKET)
                .withSnsTopicArn(SNS_TOPIC_ARN)
                .withEngine("neural");

        StartSpeechSynthesisTaskResult result =
 AMAZON_POLLY_CLIENT.startSpeechSynthesisTask(request);
        String taskId = result.getSynthesisTask().getTaskId();

        await().with()
                .pollInterval(SYNTHESIS_TASK_POLL_INTERVAL)
                .pollDelay(SYNTHESIS_TASK_POLL_DELAY)
```

```
                .atMost(SYNTHESIS_TASK_TIMEOUT_SECONDS, TimeUnit.SECONDS)
                .until(
                        () ->
 getSynthesisTaskStatus(taskId).equals(TaskStatus.Completed.toString())
                );
    }

    private static SynthesisTask getSynthesisTask(String taskId) {
        GetSpeechSynthesisTaskRequest getSpeechSynthesisTaskRequest = new
 GetSpeechSynthesisTaskRequest()
                .withTaskId(taskId);
        GetSpeechSynthesisTaskResult result
 =AMAZON_POLLY_CLIENT.getSpeechSynthesisTask(getSpeechSynthesisTaskRequest);
        return result.getSynthesisTask();
    }

    private static String getSynthesisTaskStatus(String taskId) {
        GetSpeechSynthesisTaskRequest getSpeechSynthesisTaskRequest = new
 GetSpeechSynthesisTaskRequest()
                .withTaskId(taskId);
        GetSpeechSynthesisTaskResult result
 =AMAZON_POLLY_CLIENT.getSpeechSynthesisTask(getSpeechSynthesisTaskRequest);
        return result.getSynthesisTask().getTaskStatus();
    }

}
```

스피치 마크

다음 코드 샘플은 Java 기반 애플리케이션을 사용하여 입력된 텍스트의 스피치 마크를 합성하는 방법을 보여줍니다. 이 기능은 API를 사용합니다. SynthesizeSpeech

이 기능에 대한 자세한 내용은 스피치 마크을(를) 참조하세요.

해당 API에 대한 자세한 내용은 SynthesizeSpeech API 참조를 참조하세요.

```
package com.amazonaws.polly.samples;

import com.amazonaws.services.polly.AmazonPolly;
import com.amazonaws.services.polly.AmazonPollyClientBuilder;
```

```
import com.amazonaws.services.polly.model.OutputFormat;
import com.amazonaws.services.polly.model.SpeechMarkType;
import com.amazonaws.services.polly.model.SynthesizeSpeechRequest;
import com.amazonaws.services.polly.model.SynthesizeSpeechResult;
import com.amazonaws.services.polly.model.VoiceId;

import java.io.File;
import java.io.FileOutputStream;
import java.io.InputStream;

public class SynthesizeSpeechMarksSample {
    AmazonPolly client = AmazonPollyClientBuilder.defaultClient();

    public void synthesizeSpeechMarks() {
        String outputFileName = "/tmp/speechMarks.json";

        SynthesizeSpeechRequest synthesizeSpeechRequest = new SynthesizeSpeechRequest()
                .withOutputFormat(OutputFormat.Json)
                .withSpeechMarkTypes(SpeechMarkType.Viseme, SpeechMarkType.Word)
                .withVoiceId(VoiceId.Joanna)
                .withText("This is a sample text to be synthesized.");

        try (FileOutputStream outputStream = new FileOutputStream(new
 File(outputFileName))) {
            SynthesizeSpeechResult synthesizeSpeechResult =
 client.synthesizeSpeech(synthesizeSpeechRequest);
            byte[] buffer = new byte[2 * 1024];
            int readBytes;

            try (InputStream in = synthesizeSpeechResult.getAudioStream()){
                while ((readBytes = in.read(buffer)) > 0) {
                    outputStream.write(buffer, 0, readBytes);
                }
            }
        } catch (Exception e) {
            System.err.println("Exception caught: " + e);
        }
    }
}
```

SynthesizeSpeech

다음 Java 코드 샘플은 Java 기반 애플리케이션을 사용하여 스피치를 짧은 텍스트로 합성하여 거의 실시간으로 처리하는 방법을 보여줍니다.

자세한 내용은 SynthesizeSpeech API 참조를 참조하세요.

```
package com.amazonaws.polly.samples;

import com.amazonaws.services.polly.AmazonPolly;
import com.amazonaws.services.polly.AmazonPollyClientBuilder;
import com.amazonaws.services.polly.model.OutputFormat;
import com.amazonaws.services.polly.model.SynthesizeSpeechRequest;
import com.amazonaws.services.polly.model.SynthesizeSpeechResult;
import com.amazonaws.services.polly.model.VoiceId;

import java.io.File;
import java.io.FileOutputStream;
import java.io.InputStream;

public class SynthesizeSpeechSample {
    AmazonPolly client = AmazonPollyClientBuilder.defaultClient();

    public void synthesizeSpeech() {
        String outputFileName = "/tmp/speech.mp3";

        SynthesizeSpeechRequest synthesizeSpeechRequest = new SynthesizeSpeechRequest()
                .withOutputFormat(OutputFormat.Mp3)
                .withVoiceId(VoiceId.Joanna)
                .withText("This is a sample text to be synthesized.")
                .withEngine("neural");

        try (FileOutputStream outputStream = new FileOutputStream(new
 File(outputFileName))) {
            SynthesizeSpeechResult synthesizeSpeechResult =
 client.synthesizeSpeech(synthesizeSpeechRequest);
            byte[] buffer = new byte[2 * 1024];
            int readBytes;

            try (InputStream in = synthesizeSpeechResult.getAudioStream()){
                while ((readBytes = in.read(buffer)) > 0) {
                    outputStream.write(buffer, 0, readBytes);
                }
```

```
            }
        } catch (Exception e) {
            System.err.println("Exception caught: " + e);
        }
    }
}
```

Python 샘플

다음 코드 샘플은 Python(boto3) 기반 애플리케이션을 사용하여 Amazon Polly에서 다양한 작업을 수행하는 방법을 보여줍니다. 이들 샘플은 전체 예제를 위한 것은 아니지만 AWS SDK for Python (Boto)를 사용하는 대규모 Python 애플리케이션에 포함될 수 있습니다.

코드 스니펫

- DeleteLexicon
- GetLexicon
- ListLexicon
- PutLexicon
- StartSpeechSynthesisTask
- SynthesizeSpeech

DeleteLexicon

다음 Python 코드 예제는 AWS SDK for Python (Boto) 를 사용하여 로컬 AWS 구성에 지정된 지역의 어휘를 삭제합니다. 이 예에서는 지정한 어휘만 삭제합니다. 실제로 어휘를 삭제하기 전에 계속 진행할 것인지 묻는 메시지가 표시됩니다.

다음 코드 예제는 AWS SDK 구성 파일에 저장된 기본 자격 증명을 사용합니다. 구성 파일 생성에 대한 자세한 내용은 2.1단계: 설정 AWS CLI을 참조하세요.

이 작업에 대한 자세한 내용은 `DeleteLexicon` API 참조를 참조하세요.

```
from argparse import ArgumentParser
from sys import version_info

from boto3 import Session
from botocore.exceptions import BotoCoreError, ClientError
```

```
# Define and parse the command line arguments
cli = ArgumentParser(description="DeleteLexicon example")
cli.add_argument("name", type=str, metavar="LEXICON_NAME")
arguments = cli.parse_args()

# Create a client using the credentials and region defined in the adminuser
# section of the AWS credentials and configuration files
session = Session(profile_name="adminuser")
polly = session.client("polly")

# Request confirmation
prompt = input if version_info >= (3, 0) else raw_input
proceed = prompt((u"This will delete the \"{0}\" lexicon,"
                  " do you want to proceed? [y,n]: ").format(arguments.name))

if proceed in ("y", "Y"):
    print(u"Deleting {0}...".format(arguments.name))

    try:
        # Request deletion of a lexicon by name
        response = polly.delete_lexicon(Name=arguments.name)
    except (BotoCoreError, ClientError) as error:
        # The service returned an error, exit gracefully
        cli.error(error)

    print("Done.")
else:
    print("Cancelled.")
```

GetLexicon

다음 Python 코드는 AWS SDK for Python (Boto) 를 사용하여 AWS 지역에 저장된 모든 어휘를 검색합니다. 이 예제는 어휘 이름을 명령줄 파라미터로 받아들이고 해당 어휘만 가져와서 로컬에 저장된 tmp 경로를 출력합니다.

다음 코드 예제는 AWS SDK 구성 파일에 저장된 기본 자격 증명을 사용합니다. 구성 파일 생성에 대한 자세한 내용은 2.1단계: 설정 AWS CLI을 참조하세요.

이 작업에 대한 자세한 내용은 `GetLexicon` API 참조를 참조하세요.

```
from argparse import ArgumentParser
from os import path
from tempfile import gettempdir
```

```
from boto3 import Session
from botocore.exceptions import BotoCoreError, ClientError

# Define and parse the command line arguments
cli = ArgumentParser(description="GetLexicon example")
cli.add_argument("name", type=str, metavar="LEXICON_NAME")
arguments = cli.parse_args()

# Create a client using the credentials and region defined in the adminuser
# section of the AWS credentials and configuration files
session = Session(profile_name="adminuser")
polly = session.client("polly")

print(u"Fetching {0}...".format(arguments.name))

try:
    # Fetch lexicon by name
    response = polly.get_lexicon(Name=arguments.name)
except (BotoCoreError, ClientError) as error:
    # The service returned an error, exit gracefully
    cli.error(error)

# Get the lexicon data from the response
lexicon = response.get("Lexicon", {})

# Access the lexicon's content
if "Content" in lexicon:
    output = path.join(gettempdir(), u"%s.pls" % arguments.name)
    print(u"Saving to %s..." % output)

    try:
        # Save the lexicon contents to a local file
        with open(output, "w") as pls_file:
            pls_file.write(lexicon["Content"])
    except IOError as error:
        # Could not write to file, exit gracefully
        cli.error(error)
else:
    # The response didn't contain lexicon data, exit gracefully
    cli.error("Could not fetch lexicons contents")

print("Done.")
```

ListLexicon

다음 Python 코드 예제는 AWS SDK for Python (Boto) 를 사용하여 로컬 AWS 구성에 지정된 지역의 계정에 있는 어휘를 나열합니다. 구성 파일 생성에 대한 자세한 내용은 2.1단계: 설정 AWS CLI을 참조하세요.

이 작업에 대한 자세한 내용은 `ListLexicons` API 참조를 참조하세요.

```
import sys

from boto3 import Session
from botocore.exceptions import BotoCoreError, ClientError

# Create a client using the credentials and region defined in the adminuser
# section of the AWS credentials and configuration files
session = Session(profile_name="adminuser")
polly = session.client("polly")

try:
    # Request the list of available lexicons
    response = polly.list_lexicons()
except (BotoCoreError, ClientError) as error:
    # The service returned an error, exit gracefully
    print(error)
    sys.exit(-1)

# Get the list of lexicons in the response
lexicons = response.get("Lexicons", [])
print("{0} lexicon(s) found".format(len(lexicons)))

# Output a formatted list of lexicons with some of the attributes
for lexicon in lexicons:
    print((u" - {Name} ({Attributes[LanguageCode]}), "
           "{Attributes[LexemesCount]} lexeme(s)").format(**lexicon))
```

PutLexicon

다음 코드 샘플은 Python (boto3) 기반 응용 프로그램을 사용하여 지역에 발음 어휘를 저장하는 방법을 보여줍니다. AWS

이 작업에 대한 자세한 내용은 `PutLexicon` API 참조를 참조하세요.

유의할 사항:

- 로컬 어휘 파일 이름과 저장된 어휘 이름을 제공하여 코드를 업데이트해야 합니다.
- 이 예제에서는 `pls`라는 하위 디렉토리에 사전 파일이 생성되었다고 가정합니다. 경로를 적절히 업데이트해야 합니다.

다음 코드 예제는 SDK 구성 파일에 저장된 기본 자격 증명을 사용합니다. AWS 구성 파일 생성에 대한 자세한 내용은 2.1단계: 설정 AWS CLI을 참조하세요.

이 작업에 대한 자세한 내용은 `PutLexicon` API 참조를 참조하세요.

```
from argparse import ArgumentParser

from boto3 import Session
from botocore.exceptions import BotoCoreError, ClientError

# Define and parse the command line arguments
cli = ArgumentParser(description="PutLexicon example")
cli.add_argument("path", type=str, metavar="FILE_PATH")
cli.add_argument("-n", "--name", type=str, required=True,
                 metavar="LEXICON_NAME", dest="name")
arguments = cli.parse_args()

# Create a client using the credentials and region defined in the adminuser
# section of the AWS credentials and configuration files
session = Session(profile_name="adminuser")
polly = session.client("polly")

# Open the PLS lexicon file for reading
try:
    with open(arguments.path, "r") as lexicon_file:
        # Read the pls file contents
        lexicon_data = lexicon_file.read()

        # Store the PLS lexicon on the service.
        # If a lexicon with that name already exists,
        # its contents will be updated
        response = polly.put_lexicon(Name=arguments.name,
                                     Content=lexicon_data)
except (IOError, BotoCoreError, ClientError) as error:
    # Could not open/read the file or the service returned an error,
    # exit gracefully
    cli.error(error)
```

```
print(u"The \"{0}\" lexicon is now available for use.".format(arguments.name))
```

StartSpeechSynthesisTask

다음 Python 코드 예제는 AWS SDK for Python (Boto) 를 사용하여 로컬 AWS 구성에 지정된 지역의 계정에 있는 어휘를 나열합니다. 구성 파일 생성에 대한 자세한 내용은 2.1단계: 설정 AWS CLI을 참조하세요.

자세한 내용은 StartSpeechSynthesisTask API 참조를 참조하세요.

```
import boto3
import time

polly_client = boto3.Session(
                aws_access_key_id='',
    aws_secret_access_key='',
    region_name='eu-west-2').client('polly')

response = polly_client.start_speech_synthesis_task(VoiceId='Joanna',
                OutputS3BucketName='synth-books-buckets',
                OutputS3KeyPrefix='key',
                OutputFormat='mp3',
                Text='This is a sample text to be synthesized.',
                Engine='neural')

taskId = response['SynthesisTask']['TaskId']

print( "Task id is {} ".format(taskId))

task_status = polly_client.get_speech_synthesis_task(TaskId = taskId)

print(task_status)
```

SynthesizeSpeech

다음 Python 코드 예제는 AWS SDK for Python (Boto) 를 사용하여 스피치를 짧은 텍스트로 합성하여 거의 실시간으로 처리하는 방법을 보여줍니다. 자세한 내용은 SynthesizeSpeech 작업에 대한 참조를 참조하세요.

이 예제에서는 짧은 일반 텍스트 문자열을 사용합니다. SSML 텍스트를 사용하여 출력을 더 세밀하게 제어할 수 있습니다. 자세한 내용은 SSML 문서에서 음성 생성을(를) 참조하세요.

```
import boto3

polly_client = boto3.Session(
                aws_access_key_id=,
    aws_secret_access_key=,
    region_name='us-west-2').client('polly')

response = polly_client.synthesize_speech(VoiceId='Joanna',
                OutputFormat='mp3',
                Text = 'This is a sample text to be synthesized.',
                Engine = 'neural')

file = open('speech.mp3', 'wb')
file.write(response['AudioStream'].read())
file.close()
```

애플리케이션의 예

이 섹션에는 Amazon Polly를 탐색할 때 사용할 수 있는 예제 애플리케이션 형태의 추가 예제가 포함되어 있습니다.

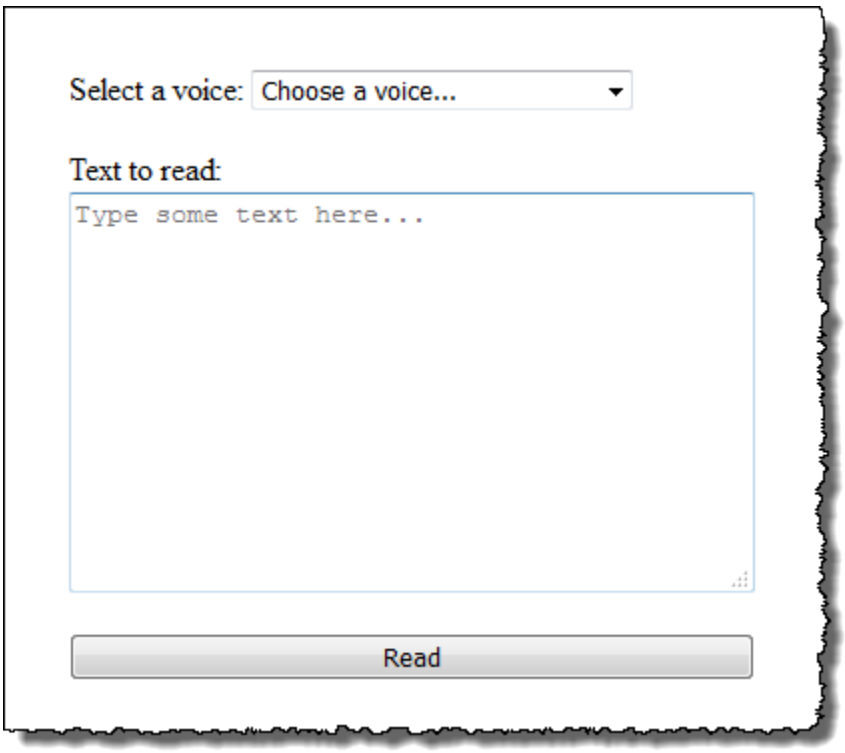
프로그래밍 언어별 예제 애플리케이션

- 파이썬 예제 (HTML5 클라이언트와 파이썬 서버)
- Java 예제
- iOS 예제
- 안드로이드 예제

파이썬 예제 (HTML5 클라이언트와 파이썬 서버)

이 예제 애플리케이션은 다음과 같은 요소로 구성됩니다.

- HTTP 청크 전송 코딩을 사용하는 HTTP 1.1 서버(청크 전송 코딩 참조)
- HTTP 1.1 서버와 상호 작용하는 간단한 HTML5 사용자 인터페이스(아래 그림 참조)



이 예제의 목표는 Amazon Polly를 사용하여 브라우저 기반 HTML5 애플리케이션에서 스피치를 스트리밍하는 방법을 보여주는 것입니다. 응답성이 중요한 요소인 사용 사례(예: 대화 시스템, 스크린 리더 등)에서는 텍스트가 합성될 때 Amazon Polly에서 생성된 오디오 스트림을 사용하는 것이 좋습니다.

이 예제 애플리케이션을 실행하려면 다음이 필요합니다.

- HTML5 및 EcmaScript 5 표준을 준수하는 웹 브라우저 (예: 크롬 23.0 이상, 파이어폭스 21.0 이상, 인터넷 익스플로러 9.0 이상)
- Python 버전 3.0 이상

애플리케이션을 테스트하려면

1. 서버 코드를 `server.py`로 저장합니다. 코드는 파이썬 예제: 파이썬 서버 코드 (server.py)을(를) 참조하세요.
2. HTML5 클라이언트 코드를 `index.html`로 저장합니다. 코드는 파이썬 예제: HTML5 사용자 인터페이스 (index.html)을(를) 참조하세요.
3. server.py를 저장한 경로에서 다음 명령을 실행하여 애플리케이션을 시작합니다(일부 시스템에서는 명령을 실행할 때 `python` 대신 `python3`를 사용해야 할 수도 있음).

```
$ python  server.py
```

애플리케이션이 시작되면 터미널에 URL이 표시됩니다.

4. 터미널에 표시된 URL을 웹 브라우저에서 엽니다.

   파라미터로 사용할 애플리케이션 서버의 주소와 포트를 `server.py`로 전달할 수 있습니다. 자세한 내용을 보려면 `python server.py -h`을(를) 실행하세요.

5. 스피치를 들으려면 목록에서 음성을 선택하고 텍스트를 입력한 다음 읽기를 선택합니다. Amazon Polly가 사용 가능한 첫 번째 오디오 데이터 청크를 전송하는 즉시 스피치가 재생되기 시작합니다.
6. 애플리케이션 테스트를 마치고 Python 서버를 중지하려면 서버를 실행하는 터미널에서 Ctrl+C를 누릅니다.

> **Note**
>
> 서버에서 AWS SDK for Python (Boto)를 사용하여 Boto3 클라이언트를 만듭니다. 클라이언트는 컴퓨터의 AWS 구성 파일에 저장된 자격 증명을 사용하여 Amazon Polly에 대한 요청에 서명하고 이를 인증합니다. AWS 구성 파일을 생성하고 자격 증명을 저장하는 방법에 대한 자세한 내용은AWS Command Line Interface 사용 설명서의 AWS Command Line Interface구성을 참조하세요.

파이썬 예제: HTML5 사용자 인터페이스 (index.html)

이 섹션에서는 파이썬 예제 (HTML5 클라이언트와 파이썬 서버)에 설명된 HTML5 클라이언트를 위한 코드를 제공합니다.

```
<html>

<head>
    <title>Text-to-Speech Example Application</title>
    <script>
        /*
         * This sample code requires a web browser with support for both the
         * HTML5 and ECMAScript 5 standards; the following is a non-comprehensive
         * list of compliant browsers and their minimum version:
         *
         * - Chrome 23.0+
```

```
         * - Firefox 21.0+
         * - Internet Explorer 9.0+
         * - Edge 12.0+
         * - Opera 15.0+
         * - Safari 6.1+
         * - Android (stock web browser) 4.4+
         * - Chrome for Android 51.0+
         * - Firefox for Android 48.0+
         * - Opera Mobile 37.0+
         * - iOS (Safari Mobile and Chrome) 3.2+
         * - Internet Explorer Mobile 10.0+
         * - Blackberry Browser 10.0+
         */

        // Mapping of the OutputFormat parameter of the SynthesizeSpeech API
        // and the audio format strings understood by the browser
        var AUDIO_FORMATS = {
            'ogg_vorbis': 'audio/ogg',
            'mp3': 'audio/mpeg',
            'pcm': 'audio/wave; codecs=1'
        };

        /**
         * Handles fetching JSON over HTTP
         */
        function fetchJSON(method, url, onSuccess, onError) {
            var request = new XMLHttpRequest();
            request.open(method, url, true);
            request.onload = function () {
                // If loading is complete
                if (request.readyState === 4) {
                    // if the request was successful
                    if (request.status === 200) {
                        var data;

                        // Parse the JSON in the response
                        try {
                            data = JSON.parse(request.responseText);
                        } catch (error) {
                            onError(request.status, error.toString());
                        }

                        onSuccess(data);
                    } else {
```

```
                    onError(request.status, request.responseText)
                }
            }
        };

        request.send();
    }

    /**
     * Returns a list of audio formats supported by the browser
     */
    function getSupportedAudioFormats(player) {
        return Object.keys(AUDIO_FORMATS)
            .filter(function (format) {
                var supported = player.canPlayType(AUDIO_FORMATS[format]);
                return supported === 'probably' || supported === 'maybe';
            });
    }

    // Initialize the application when the DOM is loaded and ready to be
    // manipulated
    document.addEventListener("DOMContentLoaded", function () {
        var input = document.getElementById('input'),
            voiceMenu = document.getElementById('voice'),
            text = document.getElementById('text'),
            player = document.getElementById('player'),
            submit = document.getElementById('submit'),
            supportedFormats = getSupportedAudioFormats(player);

        // Display a message and don't allow submitting the form if the
        // browser doesn't support any of the available audio formats
        if (supportedFormats.length === 0) {
            submit.disabled = true;
            alert('The web browser in use does not support any of the' +
                  ' available audio formats. Please try with a different' +
                  ' one.');
        }

        // Play the audio stream when the form is submitted successfully
        input.addEventListener('submit', function (event) {
            // Validate the fields in the form, display a message if
            // unexpected values are encountered
            if (voiceMenu.selectedIndex <= 0 || text.value.length === 0) {
                alert('Please fill in all the fields.');
```

```
            } else {
                var selectedVoice = voiceMenu
                                        .options[voiceMenu.selectedIndex]
                                        .value;

                // Point the player to the streaming server
                player.src = '/read?voiceId=' +
                    encodeURIComponent(selectedVoice) +
                    '&text=' + encodeURIComponent(text.value) +
                    '&outputFormat=' + supportedFormats[0];
                player.play();
            }

            // Stop the form from submitting,
            // Submitting the form is allowed only if the browser doesn't
            // support Javascript to ensure functionality in such a case
            event.preventDefault();
        });

        // Load the list of available voices and display them in a menu
        fetchJSON('GET', '/voices',
            // If the request succeeds
            function (voices) {
                var container = document.createDocumentFragment();

                // Build the list of options for the menu
                voices.forEach(function (voice) {
                    var option = document.createElement('option');
                    option.value = voice['Id'];
                    option.innerHTML = voice['Name'] + ' (' +
                        voice['Gender'] + ', ' +
                        voice['LanguageName'] + ')';
                    container.appendChild(option);
                });

                // Add the options to the menu and enable the form field
                voiceMenu.appendChild(container);
                voiceMenu.disabled = false;
            },
            // If the request fails
            function (status, response) {
                // Display a message in case loading data from the server
                // fails
                alert(status + ' - ' + response);
```

```
                });
        });

    </script>
    <style>
        #input {
            min-width: 100px;
            max-width: 600px;
            margin: 0 auto;
            padding: 50px;
        }

        #input div {
            margin-bottom: 20px;
        }

        #text {
            width: 100%;
            height: 200px;
            display: block;
        }

        #submit {
            width: 100%;
        }
    </style>
</head>

<body>
    <form id="input" method="GET" action="/read">
        <div>
            <label for="voice">Select a voice:</label>
            <select id="voice" name="voiceId" disabled>
                <option value="">Choose a voice...</option>
            </select>
        </div>
        <div>
            <label for="text">Text to read:</label>
            <textarea id="text" maxlength="1000" minlength="1" name="text"
                    placeholder="Type some text here..."></textarea>
        </div>
        <input type="submit" value="Read" id="submit" />
    </form>
    <audio id="player"></audio>
```

```
</body>

</html>
```

파이썬 예제: 파이썬 서버 코드 (server.py)

이 섹션에서는 파이썬 예제 (HTML5 클라이언트와 파이썬 서버)에 설명된 Python 서버의 코드를 제공합니다.

```
"""
Example Python 2.7+/3.3+ Application

This application consists of a HTTP 1.1 server using the HTTP chunked transfer
coding (https://tools.ietf.org/html/rfc2616#section-3.6.1) and a minimal HTML5
user interface that interacts with it.

The goal of this example is to start streaming the speech to the client (the
HTML5 web UI) as soon as the first consumable chunk of speech is returned in
order to start playing the audio as soon as possible.
For use cases where low latency and responsiveness are strong requirements,
this is the recommended approach.

The service documentation contains examples for non-streaming use cases where
waiting for the speech synthesis to complete and fetching the whole audio stream
at once are an option.

To test the application, run 'python server.py' and then open the URL
displayed in the terminal in a web browser (see index.html for a list of
supported browsers). The address and port for the server can be passed as
parameters to server.py. For more information, run: 'python server.py -h'
"""
from argparse import ArgumentParser
from collections import namedtuple
from contextlib import closing
from io import BytesIO
from json import dumps as json_encode
import os
import sys

if sys.version_info >= (3, 0):
    from http.server import BaseHTTPRequestHandler, HTTPServer
    from socketserver import ThreadingMixIn
    from urllib.parse import parse_qs
```

```
else:
    from BaseHTTPServer import BaseHTTPRequestHandler, HTTPServer
    from SocketServer import ThreadingMixIn
    from urlparse import parse_qs

from boto3 import Session
from botocore.exceptions import BotoCoreError, ClientError

ResponseStatus = namedtuple("HTTPStatus",
                            ["code", "message"])

ResponseData = namedtuple("ResponseData",
                          ["status", "content_type", "data_stream"])

# Mapping the output format used in the client to the content type for the
# response
AUDIO_FORMATS = {"ogg_vorbis": "audio/ogg",
                 "mp3": "audio/mpeg",
                 "pcm": "audio/wave; codecs=1"}
CHUNK_SIZE = 1024
HTTP_STATUS = {"OK": ResponseStatus(code=200, message="OK"),
               "BAD_REQUEST": ResponseStatus(code=400, message="Bad request"),
               "NOT_FOUND": ResponseStatus(code=404, message="Not found"),
               "INTERNAL_SERVER_ERROR": ResponseStatus(code=500, message="Internal
 server error")}
PROTOCOL = "http"
ROUTE_INDEX = "/index.html"
ROUTE_VOICES = "/voices"
ROUTE_READ = "/read"


# Create a client using the credentials and region defined in the adminuser
# section of the AWS credentials and configuration files
session = Session(profile_name="adminuser")
polly = session.client("polly")


class HTTPStatusError(Exception):
    """Exception wrapping a value from http.server.HTTPStatus"""

    def __init__(self, status, description=None):
        """
        Constructs an error instance from a tuple of
        (code, message, description), see http.server.HTTPStatus
```

```
        """
        super(HTTPStatusError, self).__init__()
        self.code = status.code
        self.message = status.message
        self.explain = description


class ThreadedHTTPServer(ThreadingMixIn, HTTPServer):
    """An HTTP Server that handle each request in a new thread"""
    daemon_threads = True


class ChunkedHTTPRequestHandler(BaseHTTPRequestHandler):
    """HTTP 1.1 Chunked encoding request handler"""
    # Use HTTP 1.1 as 1.0 doesn't support chunked encoding
    protocol_version = "HTTP/1.1"

    def query_get(self, queryData, key, default=""):
        """Helper for getting values from a pre-parsed query string"""
        return queryData.get(key, [default])[0]

    def do_GET(self):
        """Handles GET requests"""

        # Extract values from the query string
        path, _, query_string = self.path.partition('?')
        query = parse_qs(query_string)

        response = None

        print(u"[START]: Received GET for %s with query: %s" % (path, query))

        try:
            # Handle the possible request paths
            if path == ROUTE_INDEX:
                response = self.route_index(path, query)
            elif path == ROUTE_VOICES:
                response = self.route_voices(path, query)
            elif path == ROUTE_READ:
                response = self.route_read(path, query)
            else:
                response = self.route_not_found(path, query)

            self.send_headers(response.status, response.content_type)
```

```
            self.stream_data(response.data_stream)

        except HTTPStatusError as err:
            # Respond with an error and log debug
            # information
            if sys.version_info >= (3, 0):
                self.send_error(err.code, err.message, err.explain)
            else:
                self.send_error(err.code, err.message)

            self.log_error(u"%s %s %s - [%d] %s", self.client_address[0],
                           self.command, self.path, err.code, err.explain)

        print("[END]")

    def route_not_found(self, path, query):
        """Handles routing for unexpected paths"""
        raise HTTPStatusError(HTTP_STATUS["NOT_FOUND"], "Page not found")

    def route_index(self, path, query):
        """Handles routing for the application's entry point'"""
        try:
            return ResponseData(status=HTTP_STATUS["OK"], content_type="text_html",
                                # Open a binary stream for reading the index
                                # HTML file
                                data_stream=open(os.path.join(sys.path[0],
                                                              path[1:]), "rb"))
        except IOError as err:
            # Couldn't open the stream
            raise HTTPStatusError(HTTP_STATUS["INTERNAL_SERVER_ERROR"],
                                  str(err))

    def route_voices(self, path, query):
        """Handles routing for listing available voices"""
        params = {}
        voices = []

        while True:
            try:
                # Request list of available voices, if a continuation token
                # was returned by the previous call then use it to continue
                # listing
                response = polly.describe_voices(**params)
            except (BotoCoreError, ClientError) as err:
```

```
                # The service returned an error
                raise HTTPStatusError(HTTP_STATUS["INTERNAL_SERVER_ERROR"],
                                      str(err))

            # Collect all the voices
            voices.extend(response.get("Voices", []))

            # If a continuation token was returned continue, stop iterating
            # otherwise
            if "NextToken" in response:
                params = {"NextToken": response["NextToken"]}
            else:
                break

        json_data = json_encode(voices)
        bytes_data = bytes(json_data, "utf-8") if sys.version_info >= (3, 0) \
            else bytes(json_data)

        return ResponseData(status=HTTP_STATUS["OK"],
                            content_type="application/json",
                            # Create a binary stream for the JSON data
                            data_stream=BytesIO(bytes_data))

    def route_read(self, path, query):
        """Handles routing for reading text (speech synthesis)"""
        # Get the parameters from the query string
        text = self.query_get(query, "text")
        voiceId = self.query_get(query, "voiceId")
        outputFormat = self.query_get(query, "outputFormat")

        # Validate the parameters, set error flag in case of unexpected
        # values
        if len(text) == 0 or len(voiceId) == 0 or \
                outputFormat not in AUDIO_FORMATS:
            raise HTTPStatusError(HTTP_STATUS["BAD_REQUEST"],
                                  "Wrong parameters")
        else:
            try:
                # Request speech synthesis
                response = polly.synthesize_speech(Text=text,
                                                   VoiceId=voiceId,
                                                   OutputFormat=outputFormat,
                                                   Engine="neural")
            except (BotoCoreError, ClientError) as err:
```

```
            # The service returned an error
            raise HTTPStatusError(HTTP_STATUS["INTERNAL_SERVER_ERROR"],
                                  str(err))

        return ResponseData(status=HTTP_STATUS["OK"],
                            content_type=AUDIO_FORMATS[outputFormat],
                            # Access the audio stream in the response
                            data_stream=response.get("AudioStream"))

def send_headers(self, status, content_type):
    """Send out the group of headers for a successful request"""
    # Send HTTP headers
    self.send_response(status.code, status.message)
    self.send_header('Content-type', content_type)
    self.send_header('Transfer-Encoding', 'chunked')
    self.send_header('Connection', 'close')
    self.end_headers()

def stream_data(self, stream):
    """Consumes a stream in chunks to produce the response's output'"""
    print("Streaming started...")

    if stream:
        # Note: Closing the stream is important as the service throttles on
        # the number of parallel connections. Here we are using
        # contextlib.closing to ensure the close method of the stream object
        # will be called automatically at the end of the with statement's
        # scope.
        with closing(stream) as managed_stream:
            # Push out the stream's content in chunks
            while True:
                data = managed_stream.read(CHUNK_SIZE)
                self.wfile.write(b"%X\r\n%s\r\n" % (len(data), data))

                # If there's no more data to read, stop streaming
                if not data:
                    break

            # Ensure any buffered output has been transmitted and close the
            # stream
            self.wfile.flush()

        print("Streaming completed.")
    else:
```

```
            # The stream passed in is empty
            self.wfile.write(b"0\r\n\r\n")
            print("Nothing to stream.")

# Define and parse the command line arguments
cli = ArgumentParser(description='Example Python Application')
cli.add_argument(
    "-p", "--port", type=int, metavar="PORT", dest="port", default=8000)
cli.add_argument(
    "--host", type=str, metavar="HOST", dest="host", default="localhost")
arguments = cli.parse_args()

# If the module is invoked directly, initialize the application
if __name__ == '__main__':
    # Create and configure the HTTP server instance
    server = ThreadedHTTPServer((arguments.host, arguments.port),
                                ChunkedHTTPRequestHandler)
    print("Starting server, use <Ctrl-C> to stop...")
    print(u"Open {0}://{1}:{2}{3} in a web browser.".format(PROTOCOL,
                                                           arguments.host,
                                                           arguments.port,
                                                           ROUTE_INDEX))

    try:
        # Listen for requests indefinitely
        server.serve_forever()
    except KeyboardInterrupt:
        # A request to terminate has been received, stop the server
        print("\nShutting down...")
        server.socket.close()
```

Java 예제

이 예제에서는 Amazon Polly를 사용하여 Java 기반 애플리케이션에서 스피치를 스트리밍하는 방법을 보여 줍니다. 이 예제에서는 AWS SDK for Java를 사용하여 목록에서 선택한 음성으로 지정된 텍스트를 읽습니다.

표시된 코드는 주요 작업을 다루지만 최소한의 오류 검사만 수행합니다. Amazon Polly에서 오류가 발생하면 애플리케이션이 종료됩니다.

이 예제 애플리케이션을 실행하려면 다음이 필요합니다.

- Java 8 Development Kit(JDK)
- AWS SDK for Java
- Apache Maven

애플리케이션을 테스트하려면

1. JDK에 JAVA_HOME 환경 변수가 설정되어 있어야 합니다.

   예를 들어, Windows의 `C:\Program Files\Java\jdk1.8.0_121`에 JDK 1.8.0_121을 설치한 경우 명령 프롬프트에 다음을 입력합니다.

   ```
   set JAVA_HOME=""C:\Program Files\Java\jdk1.8.0_121""
   ```

   Linux의 `/usr/lib/jvm/java8-openjdk-amd64`에 JDK 1.8.0_121을 설치한 경우 명령 프롬프트에 다음을 입력합니다.

   ```
   export JAVA_HOME=/usr/lib/jvm/java8-openjdk-amd64
   ```

2. 명령줄에서 Maven을 실행하도록 Maven 환경 변수를 설정합니다.

   예를 들어 Windows의 `C:\Program Files\apache-maven-3.3.9`에 Maven 3.3.9를 설치한 경우 다음을 입력합니다.

   ```
   set M2_HOME=""C:\Program Files\apache-maven-3.3.9""
   set M2=%M2_HOME%\bin
   set PATH=%M2%;%PATH%
   ```

Linux의 /home/ec2-user/opt/apache-maven-3.3.9에 Maven 3.3.9를 설치한 경우 다음을 입력합니다.

```
export M2_HOME=/home/ec2-user/opt/apache-maven-3.3.9
export M2=$M2_HOME/bin
export PATH=$M2:$PATH
```

3. 이름이 polly-java-demo인 디렉터리를 새로 생성합니다.
4. polly-java-demo 디렉터리에 이름이 pom.xml인 파일을 생성하고 그 안에 다음 코드를 붙여 넣습니다.

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
                     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/
maven-4.0.0.xsd">
 <modelVersion>4.0.0</modelVersion>
 <groupId>com.amazonaws.polly</groupId>
 <artifactId>java-demo</artifactId>
 <version>0.0.1-SNAPSHOT</version>

 <dependencies>
  <!-- https://mvnrepository.com/artifact/com.amazonaws/aws-java-sdk-polly -->
  <dependency>
   <groupId>com.amazonaws</groupId>
   <artifactId>aws-java-sdk-polly</artifactId>
   <version>1.11.77</version>
  </dependency>
  <!-- https://mvnrepository.com/artifact/com.googlecode.soundlibs/jlayer -->
  <dependency>
   <groupId>com.googlecode.soundlibs</groupId>
   <artifactId>jlayer</artifactId>
   <version>1.0.1-1</version>
  </dependency>

 </dependencies>
 <build>
  <plugins>
   <plugin>
    <groupId>org.codehaus.mojo</groupId>
    <artifactId>exec-maven-plugin</artifactId>
    <version>1.2.1</version>
    <executions>
```

```
     <execution>
      <goals>
       <goal>java</goal>
      </goals>
     </execution>
    </executions>
    <configuration>
     <mainClass>com.amazonaws.demos.polly.PollyDemo</mainClass>
    </configuration>
   </plugin>
  </plugins>
 </build>
</project>
```

5. src/main/java/com/amazonaws/demos에 이름이 polly인 디렉터리를 새로 생성합니다.
6. polly 디렉터리에서 이름이 PollyDemo.java인 새 Java 소스 파일을 만들고 그 안에 다음 코드를 붙여 넣습니다.

```
package com.amazonaws.demos.polly;

import java.io.IOException;
import java.io.InputStream;

import com.amazonaws.ClientConfiguration;
import com.amazonaws.auth.DefaultAWSCredentialsProviderChain;
import com.amazonaws.regions.Region;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.polly.AmazonPollyClient;
import com.amazonaws.services.polly.model.DescribeVoicesRequest;
import com.amazonaws.services.polly.model.DescribeVoicesResult;
import com.amazonaws.services.polly.model.OutputFormat;
import com.amazonaws.services.polly.model.SynthesizeSpeechRequest;
import com.amazonaws.services.polly.model.SynthesizeSpeechResult;
import com.amazonaws.services.polly.model.Voice;

import javazoom.jl.player.advanced.AdvancedPlayer;
import javazoom.jl.player.advanced.PlaybackEvent;
import javazoom.jl.player.advanced.PlaybackListener;

public class PollyDemo {

 private final AmazonPollyClient polly;
```

```
private final Voice voice;
private static final String SAMPLE = "Congratulations. You have successfully built
this working demo
of Amazon Polly in Java. Have fun building voice enabled apps with Amazon Polly
(that's me!), and always
look at the AWS website for tips and tricks on using Amazon Polly and other great
services from AWS";

public PollyDemo(Region region) {
 // create an Amazon Polly client in a specific region
 polly = new AmazonPollyClient(new DefaultAWSCredentialsProviderChain(),
 new ClientConfiguration());
 polly.setRegion(region);
 // Create describe voices request.
 DescribeVoicesRequest describeVoicesRequest = new DescribeVoicesRequest();

 // Synchronously ask Amazon Polly to describe available TTS voices.
 DescribeVoicesResult describeVoicesResult =
polly.describeVoices(describeVoicesRequest);
 voice = describeVoicesResult.getVoices().get(0);
}

public InputStream synthesize(String text, OutputFormat format) throws IOException
{
 SynthesizeSpeechRequest synthReq =
 new SynthesizeSpeechRequest().withText(text).withVoiceId(voice.getId())
   .withOutputFormat(format).withEngine("neural");
 SynthesizeSpeechResult synthRes = polly.synthesizeSpeech(synthReq);

 return synthRes.getAudioStream();
}

public static void main(String args[]) throws Exception {
 //create the test class
 PollyDemo helloWorld = new PollyDemo(Region.getRegion(Regions.US_EAST_1));
 //get the audio stream
 InputStream speechStream = helloWorld.synthesize(SAMPLE, OutputFormat.Mp3);

 //create an MP3 player
 AdvancedPlayer player = new AdvancedPlayer(speechStream,
   javazoom.jl.player.FactoryRegistry.systemRegistry().createAudioDevice());

 player.setPlayBackListener(new PlaybackListener() {
  @Override
```

```
    public void playbackStarted(PlaybackEvent evt) {
     System.out.println("Playback started");
     System.out.println(SAMPLE);
    }

    @Override
    public void playbackFinished(PlaybackEvent evt) {
     System.out.println("Playback finished");
    }
   });


   // play it!
   player.play();

  }
}
```

7. polly-java-demo 디렉터리로 돌아가서 데모를 정리하고, 컴파일하고, 실행합니다.

```
mvn clean compile exec:java
```

iOS 예제

다음 예제에서는 Amazon Polly용 iOS SDK를 사용하여 음성 목록에서 선택한 음성으로 지정된 텍스트를 읽습니다.

여기에 표시된 코드는 주요 작업을 다루지만 오류는 처리하지 않습니다. 전체 코드는 AWS Mobile SDK for iOS Amazon Polly 데모를 참조하세요.

초기화

```
// Region of Amazon Polly.
let AwsRegion = AWSRegionType.usEast1

// Cognito pool ID. Pool needs to be unauthenticated pool with
// Amazon Polly permissions.
let CognitoIdentityPoolId = "YourCognitoIdentityPoolId"

// Initialize the Amazon Cognito credentials provider.
let credentialProvider = AWSCognitoCredentialsProvider(regionType: AwsRegion,
 identityPoolId: CognitoIdentityPoolId)

// Create an audio player
var audioPlayer = AVPlayer()
```

사용 가능한 음성 목록 가져오기

```
// Use the configuration as default
AWSServiceManager.default().defaultServiceConfiguration = configuration

// Get all the voices (no parameters specified in input) from Amazon Polly
// This creates an async task.
let task = AWSPolly.default().describeVoices(AWSPollyDescribeVoicesInput())

// When the request is done, asynchronously do the following block
// (we ignore all the errors, but in a real-world scenario they need
// to be handled)
task.continue(successBlock: { (awsTask: AWSTask) -> Any? in
      // awsTask.result is an instance of AWSPollyDescribeVoicesOutput in
      // case of the "describeVoices" method
      let voices = (awsTask.result! as AWSPollyDescribeVoicesOutput).voices
```

```
      return nil
})
```

스피치 합성

```
// First, Amazon Polly requires an input, which we need to prepare.
// Again, we ignore the errors, however this should be handled in
// real applications. Here we are using the URL Builder Request,
// since in order to make the synthesis quicker we will pass the
// presigned URL to the system audio player.
let input = AWSPollySynthesizeSpeechURLBuilderRequest()

// Text to synthesize
input.text = "Sample text"

// We expect the output in MP3 format
input.outputFormat = AWSPollyOutputFormat.mp3

// Choose the voice ID
input.voiceId = AWSPollyVoiceId.joanna

// Create an task to synthesize speech using the given synthesis input
let builder = AWSPollySynthesizeSpeechURLBuilder.default().getPreSignedURL(input)

// Request the URL for synthesis result
builder.continueOnSuccessWith(block: { (awsTask: AWSTask<NSURL>) -> Any? in
 // The result of getPresignedURL task is NSURL.
 // Again, we ignore the errors in the example.
 let url = awsTask.result!

 // Try playing the data using the system AVAudioPlayer
 self.audioPlayer.replaceCurrentItem(with: AVPlayerItem(url: url as URL))
 self.audioPlayer.play()

 return nil
})
```

안드로이드 예제

다음 예제에서는 Amazon Polly용 Android SDK를 사용하여 음성 목록에서 선택한 음성으로 지정된 텍스트를 읽습니다.

여기에 표시된 코드는 주요 작업을 다루지만 오류는 처리하지 않습니다. 전체 코드는 AWS Mobile SDK for Android Amazon Polly 데모를 참조하세요.

초기화

```
// Cognito pool ID. Pool needs to be unauthenticated pool with
// Amazon Polly permissions.
String COGNITO_POOL_ID = "YourCognitoIdentityPoolId";

// Region of Amazon Polly.
Regions MY_REGION = Regions.US_EAST_1;

// Initialize the Amazon Cognito credentials provider.
CognitoCachingCredentialsProvider credentialsProvider = new
 CognitoCachingCredentialsProvider(
        getApplicationContext(),
        COGNITO_POOL_ID,
        MY_REGION
);

// Create a client that supports generation of presigned URLs.
AmazonPollyPresigningClient client = new
 AmazonPollyPresigningClient(credentialsProvider);
```

사용 가능한 음성 목록 가져오기

```
// Create describe voices request.
DescribeVoicesRequest describeVoicesRequest = new DescribeVoicesRequest();

// Synchronously ask Amazon Polly to describe available TTS voices.
DescribeVoicesResult describeVoicesResult =
 client.describeVoices(describeVoicesRequest);
List<Voice> voices = describeVoicesResult.getVoices();
```

오디오 스트림용 URL 가져오기

```
// Create speech synthesis request.
SynthesizeSpeechPresignRequest synthesizeSpeechPresignRequest =
        new SynthesizeSpeechPresignRequest()
        // Set the text to synthesize.
        .withText("Hello world!")
        // Select voice for synthesis.
        .withVoiceId(voices.get(0).getId()) // "Joanna"
        // Set format to MP3.
        .withOutputFormat(OutputFormat.Mp3);

// Get the presigned URL for synthesized speech audio stream.
URL presignedSynthesizeSpeechUrl =
        client.getPresignedSynthesizeSpeechUrl(synthesizeSpeechPresignRequest);
```

합성 스피치 재생

```
// Use MediaPlayer: https://developer.android.com/guide/topics/media/mediaplayer.html

// Create a media player to play the synthesized audio stream.
MediaPlayer mediaPlayer = new MediaPlayer();
mediaPlayer.setAudioStreamType(AudioManager.STREAM_MUSIC);

try {
    // Set media player's data source to previously obtained URL.
    mediaPlayer.setDataSource(presignedSynthesizeSpeechUrl.toString());
} catch (IOException e) {
    Log.e(TAG, "Unable to set data source for the media player! " + e.getMessage());
}

// Prepare the MediaPlayer asynchronously (since the data source is a network stream).
mediaPlayer.prepareAsync();

// Set the callback to start the MediaPlayer when it's prepared.
mediaPlayer.setOnPreparedListener(new MediaPlayer.OnPreparedListener() {
    @Override
    public void onPrepared(MediaPlayer mp) {
        mp.start();
    }
});

// Set the callback to release the MediaPlayer after playback is completed.
mediaPlayer.setOnCompletionListener(new MediaPlayer.OnCompletionListener() {
```

```
    @Override
    public void onCompletion(MediaPlayer mp) {
 mp.release();
    }
});
```

Amazon Polly의 할당량

Amazon Polly는 과도한 요청을 거부하는 방식으로 고객 트래픽에 할당량을 적용합니다. 표준 음성이 포함된 SynthesizeSpeech 요청의 기본 할당량은 단일 리전 단일 AWS 계정에서 초당 80건의 트랜잭션(tps) 입니다. 제한이 증가하지 않은 상태에서 표준 음성을 사용하여 초당 100개의 SynthesizeSpeech 요청을 생성한 경우 Amazon Polly에서 초당 80개의 요청이 성공하고 초당 20개의 요청이 제한됩니다. 이러한 요청은 HTTP 상태 400과 ThrottlingException을 나타내는 응답 헤더를 반환합니다. Amazon Polly는 또한 요청 속도를 기준으로 모든 작업에 대한 트래픽을 조절합니다.

스피치 합성 제한 예제

- 영어 알파벳의 처음 24자를 한 번에 한 글자씩 합성합니다. 각 문자의 합성이 50밀리초 미만이고 작업 제한이 8tps인 경우 24개의 문자를 합성하는 데 최소 3초가 걸립니다. 이 시간 동안에는 초당 최대 8개의 문자를 합성할 수 있습니다. 그러나 추가 요청은 제한됩니다. 요청은 짧은 시간 동안 지속되므로 중복되지 않고 순차적으로 합성됩니다.
- 16단락 텍스트를 합성합니다. 각 단락이 합성되고 클라이언트 측에서 2초 이내에 완전히 수신되는 상황에서 작업이 8개로 제한되어 있는 경우 16개 단락을 모두 합성하는 데 최소 4초가 걸립니다. 처음 1초 동안에는 요청을 8개까지 시작할 수 있습니다. 동시 요청 중에는 동시성 제한으로 인해 새 합성을 시작하려는 모든 시도가 제한됩니다. 처음 요청이 처리되는 첫 2초 후 나머지 8개 단락을 합성할 수 있습니다.

Amazon Polly를 사용할 때는 다음 제한 사항을 염두에 두어야 합니다.

주제

- 지원되는 리전
- 할당량 및 제한 속도
- 발음 어휘
- SynthesizeSpeech API 오퍼레이션
- SpeechSynthesisTask API 오퍼레이션
- SSML(Speech Synthesis Markup Language)

지원되는 리전

Amazon Polly를 사용할 수 있는 AWS 지역 목록은 의 Amazon Polly 엔드포인트 및 할당량을 참조하십시오. Amazon Web Services 일반 참조

- 제너레이티브 보이스를 지원하는 지역의 경우 제너레이티브 보이스를 참조하십시오.
- 장편 음색을 지원하는 지역의 경우 장편 음성을 참조하세요.
- 신경 음성을 지원하는 리전은 신경 TTS용 the section called “기능 및 지역 호환성”을 참조하세요.

할당량 및 제한 속도

다음 표에는 Amazon Polly 작업당 제한 속도가 정의되어 있습니다. 필요한 경우 를 사용하여 조정 가능한 할당량에 대한 할당량 증가를 요청할 수 있습니다. AWS Management Console

Operation	Limit
어휘	

`DeleteLexicon`
`PutLexicon`
`GetLexicon`
`ListLexicons`
	이 작업이 완료된 후 2건의 초당 트랜잭션(tps)입니다.
	허용되는 최대 버스트는 4tps입니다.

스피치	
`DescribeVoices`	80tps 및 버스트 한도 100tps
`SynthesizeSpeech`	제너레이티브 보이스: 8tps 롱폼 음성: 8tps 및 버스트 한도 10tps 신경 음성: 8tps 및 버스트 한도 10tps 표준 음성: 80tps 및 버스트 한도 100tps
`StartSpeechSynthesisTask`	제너레이티브 보이스: 10 팁 롱폼 음성: 1tps 신경 음성: 1tps 표준 음성: 10tps 및 버스트 한도 12tps

Operation	Limit
`GetSynthesizeSpeechTask` 및 `ListSynthesizeSpeechTask`	최대 10tps 결합 허용

동시 요청

제너레이티브 음성의 경우 Amazon Polly는 최대 26개의 동시 요청을 지원합니다. 롱폼 음성의 경우 Amazon Polly는 최대 26개의 동시 요청을 지원합니다. 신경 음성의 경우 Amazon Polly는 최대 18개의 동시 요청에 대해 8tps의 버스트 한도와 10tps의 버스트 제한을 지원합니다. Amazon Polly는 동시 요청에 대한 제한도 지원합니다. 표준 음성의 경우 Amazon Polly는 최대 80개의 동시 요청에 대해 80tps를 지원합니다.

제한 완화를 위한 모범 사례

- 백오프 및 지터로 제한을 재시도하여 부하를 단기간에 분산하고 가용성에 영향을 주지 않으면서 예상치 못한 사용량 피크를 처리할 수 있습니다. 많은 프로그래밍 언어에서 기본적으로 AWS 코드 샘플 카탈로그 가 이 작업을 수행하도록 구성되어 있습니다. 자세한 내용은 기능 재시도 동작을 참조하세요.
- Amazon Polly 지표를 사용하세요. Amazon Polly는 현재 사용량을 분석하고 사용량 증가를 CloudWatch 예측하기 위해 자동으로 게시합니다.

> **Note**
>
> 할당량 증가(해당하는 경우)를 요청하기 전에 이 페이지의 가이드라인에 따라 tps 요구 사항을 계산합니다. Amazon Polly는 비용 절감을 위해 고객의 요구에 따라 필요한 연산 리소스만 확보합니다.

발음 어휘

- 계정당 어휘를 최대 100개까지 저장할 수 있습니다.
- 어휘 이름은 최대 20자의 영숫자 문자열로 지정할 수 있습니다.

- 각 어휘의 최대 크기는 40,000자입니다. (어휘의 크기는 작업 지연 시간에 영향을 미친다는 점에 유의하십시오.) SynthesizeSpeech
- 어휘의 <phoneme> 또는 <alias> 대체어마다 100자까지 지정할 수 있습니다.

어휘를 사용하는 방법에 대한 자세한 내용은 어휘집 관리를 참조하세요.

SynthesizeSpeech API 오퍼레이션

SynthesizeSpeech의 사용량을 추정할 때는 Amazon Polly에서 생성된 오디오, 특히 대화형 애플리케이션의 오디오가 재생되는 데 보통 몇 초 이상 걸린다는 점을 염두에 두어야 합니다. 이렇게 하면 동시 사용자 수가 많아도 SynthesizeSpeech 요청 속도가 줄어듭니다. 또한 Amazon Polly는 합성한 동시 SynthesizeSpeech 요청 수를 기준으로 요청을 조절합니다. 동시 요청에 대한 별도의 설정은 없습니다. 동시 요청 한도는 허용되는 tps 수와 항상 같은 값을 가지며 이에 따라 조정됩니다.

간단한 예제 애플리케이션. Amazon Polly를 사용하여 일련의 단편 소설을 재생하는 애플리케이션을 구축할 수 있습니다. 이런 종류의 앱을 사용하면 첫 번째 스토리가 재생되기 시작하고 사용자가 애플리케이션을 종료할 때까지 다음 스토리가 계속 재생됩니다. 각 스토리를 합성하는 데 약 0.5초, 재생하는 데 약 10초가 소요됩니다. 이 시나리오에서는 고객이 애플리케이션을 사용하는 동안 10초마다 한 건의 SynthesizeSpeech 호출이 발생할 것으로 예상할 수 있습니다. 이는 애플리케이션을 동시에 사용하는 고객 10명이 초당 호출 1건을 발생시킨다는 것으로 환산됩니다. 1,000명의 고객이 동시에 애플리케이션을 사용하는 경우 초당 평균 트랜잭션 속도인 100건에서 SynthesizeSpeech에 불과할 것으로 예상할 수 있습니다.

SynthesizeSpeech API 작업 사용과 관련하여 다음 제한 사항에 유의해야 합니다.

- 청구되는 입력 텍스트의 최대 크기는 3,000자(총 6,000자)입니다. SSML 태그는 청구 문자 수로 계산되지 않습니다.
- 입력 텍스트에 적용할 최대 5개의 어휘를 지정할 수 있습니다.
- 출력 오디오 스트림(합성)은 10분으로 제한되며, 이 시간을 초과할 경우 나머지 스피치는 잘립니다.

자세한 내용은 SynthesizeSpeech을(를) 참조하세요.

> **Note**
>
> SynthesizeSpeech API 작업의 몇 가지 제한을 StartSythensizeSpeechTask API 작업을 사용하여 우회할 수 있습니다. 자세한 정보는 긴 오디오 파일 생성을 참조하세요.

SpeechSynthesisTask API 오퍼레이션

`StartSpeechSynthesisTask`, `GetSpeechSynthesisTask` 및 `ListSpeechSynthesisTasks` API 작업 사용과 관련하여 다음 제한 사항에 유의해야 합니다.

- 청구되는 입력 텍스트의 최대 크기는 100,000자(총 200,000자)입니다. SSML 태그는 청구 문자 수로 계산되지 않습니다.
- 입력 텍스트에 적용할 최대 5개의 어휘를 지정할 수 있습니다.

SSML(Speech Synthesis Markup Language)

SSML 사용과 관련하여 다음 제한 사항에 유의해야 합니다.

- `<audio>`, `<lexicon>`, `<lookup>` 및 `<voice>` 태그는 지원되지 않습니다.
- `<break>` 요소는 각각 10초의 최대 기간을 지정할 수 있습니다.
- `<prosody>` 태그는 -80% 미만의 비율 속성 값을 지원하지 않습니다.

자세한 내용은 SSML 문서에서 음성 생성을(를) 참조하세요.

Amazon Polly의 보안

클라우드 AWS 보안이 최우선 과제입니다. AWS 고객은 가장 보안에 민감한 조직의 요구 사항을 충족하도록 구축된 데이터 센터 및 네트워크 아키텍처의 혜택을 누릴 수 있습니다.

보안은 기업과 기업 간의 AWS 공동 책임입니다. 공동 책임 모델은 이 사항을 클라우드의 보안 및 클라우드 내 보안으로 설명합니다.

- 클라우드 보안 - AWS 클라우드에서 AWS 서비스를 실행하는 인프라를 보호하는 역할을 합니다. AWS AWS 또한 안전하게 사용할 수 있는 서비스를 제공합니다. Amazon Polly에 적용되는 규정 준수 프로그램에 대해 자세히 알아보려면 규정 준수 프로그램별 범위 내AWS 서비스 규정 준수 참조하십시오.
- 클라우드에서의 보안 — 귀하의 책임은 사용하는 AWS 서비스에 따라 결정됩니다. 또한 귀하는 데이터의 민감도, 회사 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

이 설명서는 Amazon Polly를 사용할 때 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다. 다음 주제에서는 보안 및 규정 준수 목적에 맞게 Amazon Polly를 구성하는 방법을 보여줍니다. 또한 Amazon Polly 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스를 사용하는 방법도 알아봅니다.

주제

- Amazon Polly에서의 데이터 보호
- Amazon Polly의 Identity and Access Management
- Amazon Polly의 로깅 및 모니터링
- Amazon Polly에 대한 규정 준수 확인
- Amazon Polly의 복원성
- Amazon Polly의 인프라 보안
- Amazon Polly의 보안 모범 사례
- 인터페이스 VPC 엔드포인트에서 Amazon Polly 사용

Amazon Polly에서의 데이터 보호

Amazon Polly는 데이터 보호 규정 및 지침이 포함된 AWS 공동 책임 모델을 준수합니다. AWS 모든 서비스를 실행하는 글로벌 인프라를 보호할 책임이 있습니다. AWS AWS 고객 콘텐츠 및 개인 데이터 처

리를 위한 보안 구성 제어를 포함하여 이 인프라에서 호스팅되는 데이터에 대한 제어를 유지합니다. AWS 데이터 컨트롤러 또는 데이터 처리자 역할을 하는 고객 및 APN 파트너는 AWS 클라우드에 저장하는 모든 개인 데이터에 대한 책임을 집니다.

데이터 보호를 위해 AWS 계정 자격 증명을 보호하고 개별 사용자 AWS Identity and Access Management (IAM) 를 설정하여 각 사용자에게 직무를 수행하는 데 필요한 권한만 부여하도록 하는 것이 좋습니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정에 멀티 팩터 인증 설정(MFA)을 사용하세요.
- SSL/TLS를 사용하여 리소스와 통신하세요. AWS
- 를 사용하여 API 및 사용자 활동 로깅을 설정합니다. AWS CloudTrail
- AWS 서비스 내의 모든 기본 보안 제어와 함께 AWS 암호화 솔루션을 사용하십시오.

이름 필드와 같은 자유 형식 필드에 고객 계정 번호와 같은 중요 식별 정보를 절대 입력하지 마십시오. 여기에는 콘솔 AWS CLI, API 또는 SDK를 사용하여 Amazon Polly 또는 기타 AWS 서비스를 사용하는 경우가 포함됩니다. AWS Amazon Polly 또는 기타 서비스에 입력하는 모든 데이터를 진단 로그에 포함할 수 있습니다. 외부 서버에 URL을 제공할 때 해당 서버에 대한 요청을 검증하기 위해 자격 증명 정보를 URL에 포함시키지 마십시오.

데이터 보호에 대한 자세한 내용은 AWS 보안 블로그의 AWS 공동 책임 모델 및 GDPR 블로그 게시물을 참조하십시오.

유휴 데이터 암호화

Amazon Polly 음성 합성 출력은 자체 시스템에 저장할 수 있습니다. 또한 Amazon Polly를 호출한 다음 원하는 암호화 키를 사용하여 파일을 암호화하고 Amazon Simple Storage Service(Amazon S3) 또는 다른 보안 스토리지에 저장할 수도 있습니다. Amazon Polly the section called “SynthesizeSpeech” 호출은 상태를 저장하지 않으며 고객 자격 증명과 연결되지 않습니다. 나중에 Amazon Polly에서 가져올 수 없습니다.

전송 중 데이터 암호화

모든 텍스트 제출은 전송 중에 SSL(Secure Sockets Layer)에 의해 보호됩니다. Amazon Polly는 텍스트 제출의 콘텐츠를 보관하지 않습니다.

인터네트워크 트래픽 개인 정보

AWS 콘솔, CLI 또는 SDK를 통해 Amazon Polly에 액세스할 수 있습니다. 통신은 기밀성을 위해 TLS(전송 계층 보안) 세션 암호화를 활용하고 인증 및 무결성을 위해 디지털 서명을 사용합니다.

Amazon Polly의 Identity and Access Management

AWS Identity and Access Management (IAM) 은 관리자가 리소스에 대한 액세스를 안전하게 제어할 수 AWS 서비스 있도록 도와줍니다. AWS IAM 관리자는 어떤 사용자가 Amazon Polly 리소스를 사용할 수 있도록 인증(로그인)되고 권한이 부여(권한 있음)될 수 있는지 제어합니다. IAM은 추가 AWS 서비스 비용 없이 사용할 수 있습니다.

주제

- 고객
- 자격 증명을 통한 인증
- 정책을 사용한 액세스 관리
- Amazon Polly에서 IAM을 사용하는 방법
- Amazon Polly의 자격 증명 기반 정책 예
- Amazon Polly API 권한: 작업, 권한 및 리소스 참조
- Amazon Polly 자격 증명 및 액세스 문제 해결

고객

Amazon Polly에서 수행하는 작업에 따라 사용 방법 AWS Identity and Access Management (IAM) 이 다릅니다.

서비스 사용자 – Amazon Polly 서비스를 사용하여 작업을 수행하는 경우 필요한 자격 증명과 권한을 관리자가 제공합니다. 다른 Amazon Polly 기능을 사용하여 작업을 수행한다면 추가 권한이 필요할 수 있습니다. 액세스 권한 관리 방식을 이해하면 적절한 권한을 관리자에게 요청할 수 있습니다. Amazon Polly의 기능에 액세스할 수 없다면 Amazon Polly 자격 증명 및 액세스 문제 해결 단원을 참조하세요.

서비스 관리자 – 회사에서 Amazon Polly 리소스를 책임지고 있다면 Amazon Polly에 대한 모든 액세스 권한이 있을 것입니다. 서비스 관리자는 서비스 사용자가 액세스해야 하는 Amazon Polly 기능과 리소스를 결정합니다. 그런 다음, IAM 관리자에게 요청을 제출하여 서비스 사용자의 권한을 변경해야 합니다. 이 페이지의 정보를 검토하여 IAM의 기본 개념을 이해합니다. 회사가 Amazon Polly에서 IAM을 사용하는 방법에 대해 자세히 알아보려면 Amazon Polly에서 IAM을 사용하는 방법 단원을 참조하세요.

IAM 관리자 - IAM 관리자라면 Amazon Polly에 대한 액세스 권한을 관리하는 정책을 작성하는 방법을 자세히 알고 싶을 것입니다. IAM에서 사용할 수 있는 Amazon Polly 자격 증명 기반 정책(identity-based policies)의 예제를 확인하시려면 자격 증명 기반 정책 예제 Amazon Polly의 자격 증명 기반 정책 예 섹션을 참조하세요.

자격 증명을 통한 인증

인증은 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 AWS IAM Identity Center (IAM ID 센터) 사용자, 회사의 싱글 사인온 인증, Google 또는 Facebook 자격 증명이 페더레이션 ID의 예입니다. 페더레이션형 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 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 사용자 설명서의 AWS에서 다중 인증(MFA) 사용을 참조합니다.

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 작업에 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 태스크의 전체 목록은 IAM 사용자 안내서의 루트 사용자 보안 인증이 필요한 태스크를 참조하세요.

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 Service페더레이션 ID에 AWS 계정액세스하면 이들이 역할을 맡고 역할은 임시 자격 증명을 제공합니다.

중앙 집중식 액세스 관리를 위해 AWS IAM Identity Center을 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 내용은AWS IAM Identity Center 사용 설명서에서 IAM Identity Center란 무엇입니까?를 참조하세요.

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 자격 증명이 있는 IAM 사용자를 생성하는 대신 임시 자격 증명을 사용하는 것이 좋습니다. 하지만 IAM 사용자의 장기 자격 증명이 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 교체를 참조하세요.

IAM 그룹은 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 수임할 수 있습니다. 사용자는 영구적인 장기 자격 증명을 가지고 있지만, 역할은 임시 보안 인증만 제공합니다. 자세한 정보는 IAM 사용 설명서의 IAM 사용자를 만들어야 하는 경우(역할이 아님)를 참조하세요.

IAM 역할

IAM 역할은 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 수임할 수 있습니다. 역할 사용 방법에 대한 자세한 정보는 IAM 사용 설명서의 IAM 역할 사용을 참조하세요.

임시 보안 인증 정보가 있는 IAM 역할은 다음과 같은 상황에서 유용합니다.

- 페더레이션 사용자 액세스 - 페더레이션 아이덴티티에 권한을 부여하려면 역할을 생성하고 해당 역할의 권한을 정의합니다. 연동 자격 증명이 인증되면 역할이 연결되고 역할에 정의된 권한이 부여됩니다. 페더레이션 역할에 대한 자세한 내용은 IAM 사용 설명서의 타사 자격 증명 공급자의 역할 만들기를 참조하세요. IAM Identity Center를 사용하는 경우 권한 세트를 구성합니다. 인증 후 자격 증

명이 액세스할 수 있는 항목을 제어하기 위해 IAM Identity Center는 권한 세트를 IAM의 역할과 연관짓습니다. 권한 세트에 대한 자세한 내용은AWS IAM Identity Center 사용 설명서의 권한 세트를 참조하세요.

- 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 서비스경우에는 역할을 프록시로 사용하는 대신 정책을 리소스에 직접 연결할 수 있습니다. 크로스 계정 액세스를 위한 역할과 리소스 기반 정책의 차이점을 알아보려면 IAM 사용 설명서의 IAM 역할과 리소스 기반 정책의 차이를 참조하세요.
- 서비스 간 액세스 — 일부는 다른 AWS 서비스서비스의 기능을 AWS 서비스 사용합니다. 예컨대, 어떤 서비스에서 호출을 수행하면 일반적으로 해당 서비스는 Amazon EC2에서 애플리케이션을 실행하거나 Amazon S3에 객체를 저장합니다. 서비스는 호출하는 보안 주체의 권한을 사용하거나, 서비스 역할을 사용하거나, 또는 서비스 연결 역할을 사용하여 이 작업을 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 서비스서비스에 AWS 서비스 요청하기 위한 요청과 결합하여 사용합니다. FAS 요청은 다른 서비스 AWS 서비스 또는 리소스와의 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 세션을 참조하세요.
  - 서비스 역할 - 서비스 역할은 서비스가 사용자를 대신하여 태스크를 수행하기 위해 맡는 IAM 역할입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 정보는 IAM 사용자 설명서의 AWS 서비스에 대한 권한을 위임할 역할 생성을 참조합니다.
  - 서비스 연결 역할 — 서비스 연결 역할은 에 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 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 프로필을 생성합니다. 인스턴스 프로파일에는 역할이 포함되어 있으며 EC2 인스턴스에서 실행되는 프로그램이 임시 보안 인증을 얻을 수 있습니다. 자세한 정보는 IAM 사용 설명서의 IAM 역할을 사용하여 Amazon EC2 인스턴스에서 실행되는 애플리케이션에 권한 부여를 참조하세요.

IAM 역할을 사용할지 또는 IAM 사용자를 사용할지를 알아보려면 IAM 사용자 설명서의 IAM 역할(사용자 대신)을 생성하는 경우를 참조합니다.

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 정보는 IAM 사용 설명서의 JSON 정책 개요를 참조하세요.

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`태스크를 허용하는 정책이 있다고 가정합니다. 해당 정책을 사용하는 사용자는 AWS Management Console, AWS CLI, 또는 AWS API에서 역할 정보를 가져올 수 있습니다.

ID 기반 정책

ID 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 작업을 수행할 수 있는 지를 제어합니다. 자격 증명 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 IAM 정책 생성을 참조하세요.

자격 증명 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 선택을 참조하세요.

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 작업을 수행할 수 있는 지를 정의합니다. 리소스 기반 정책에서 보안 주체를 지정해야 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(ACLs)

액세스 제어 목록(ACLs)은 어떤 보안 주체(계정 멤버, 사용자 또는 역할)가 리소스에 액세스할 수 있는 권한을 가지고 있는 지를 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 안내서의 액세스 제어 목록(ACL) 개요를 참조하세요.

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 보안 인증 기반 정책에 따라 IAM 엔터티(IAM 사용자 또는 역할)에 부여할 수 있는 최대 권한을 설정하는 고급 기능입니다. 개체에 대한 권한 경계를 설정할 수 있습니다. 그 결과로 얻는 권한은 엔터티의 자격 증명 기반 정책과 그 권한 경계의 교집합입니다. `Principal` 필드에서 사용자나 역할을 보안 주체로 지정하는 리소스 기반 정책은 권한 경계를 통해 제한되지 않습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 권한 경계에 대한 자세한 정보는 IAM 사용자 설명서의 IAM 엔터티에 대한 권한 경계를 참조합니다.
- 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 정보는AWS Organizations 사용 설명서의 SCP 작동 방식을 참조하세요.
- 세션 정책 – 세션 정책은 역할 또는 연합된 사용자에 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 결과적으로 얻는 세션의 권한은 사용자 또는 역할 자격 증명 기반 정책의 교차 및 세션 정책입니다. 또한 권한을 리소스 기반 정책에서 가져올 수도 있습니다.

이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 자세한 정보는 IAM 사용자 설명서의 세션 정책을 참조합니다.

여러 정책 타입

여러 정책 타입이 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 로직을 참조하십시오.

Amazon Polly에서 IAM을 사용하는 방법

IAM을 사용하여 Amazon Polly에 대한 액세스를 관리하기 전에 Amazon Polly에서 사용할 수 있는 IAM 기능에 대해 알아봅니다.

Amazon Polly에서 사용할 수 있는 IAM 기능

IAM 특성	Amazon Polly 지원
ID 기반 정책	예
리소스 기반 정책	아니요
정책 작업	예
정책 리소스	예
정책 조건 키(서비스별)	아니요
ACL	아니요
ABAC(정책 내 태그)	아니요
임시 보안 인증	예
아마존 Polly용 포워드 액세스 세션 (FAS)	예
서비스 역할	아니요
서비스 연결 역할	아니요

Amazon Polly 및 AWS 기타 서비스가 대부분의 IAM 기능과 어떻게 작동하는지 자세히 알아보려면 IAM 사용 설명서의 IAM과 함께 작동하는 서비스를AWS 참조하십시오.

Amazon Polly의 자격 증명 기반 정책

ID 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 작업을 수행할 수 있는 지를 제어합니다. 자격 증명 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 IAM 정책 생성을 참조하세요.

IAM 자격 증명 기반 정책을 사용하면 허용되거나 거부되는 작업와 리소스뿐 아니라 작업이 허용되거나 거부되는 조건을 지정할 수 있습니다. 자격 증명 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 참조를 참조하세요.

Amazon Polly의 자격 증명 기반 정책 예

Amazon Polly 자격 증명 기반 정책 예제를 보려면 Amazon Polly의 자격 증명 기반 정책 예 단원을 참조하세요.

Amazon Polly 내의 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 작업을 수행할 수 있는 지를 정의합니다. 리소스 기반 정책에서 보안 주체를 지정해야 합니다. 보안 주체에는 계정, 사용자, 역할, 연동 사용자 등이 포함될 수 있습니다. AWS 서비스

계정 간 액세스를 활성화하려는 경우 전체 계정이나 다른 계정의 IAM 엔터티를 리소스 기반 정책의 보안 주체로 지정할 수 있습니다. 리소스 기반 정책에 크로스 계정 보안 주체를 추가하는 것은 트러스트

관계 설정의 절반밖에 되지 않는다는 것을 유념하세요. 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. 개체에 자격 증명 기반 정책을 연결하여 권한을 부여합니다. 하지만 리소스 기반 정책이 동일 계정의 보안 주체에 액세스를 부여하는 경우 추가 자격 증명 기반 정책이 필요하지 않습니다. 자세한 정보는 IAM 사용 설명서의 IAM 역할과 리소스 기반 정책의 차이를 참조하세요.

Amazon Polly의 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 `Action`요소는 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 Polly 작업 목록을 보려면 서비스 승인 참조의 Amazon Polly에서 정의한 작업을 참조하세요.

Amazon Polly의 정책 작업은 작업 앞에 다음 접두사를 사용합니다.

```
polly
```

단일 문에서 여러 작업을 지정하려면 다음과 같이 쉼표로 구분합니다.

```
"Action": [
      "polly:action1",
      "polly:action2"
         ]
```

Amazon Polly 자격 증명 기반 정책 예제를 보려면 Amazon Polly의 자격 증명 기반 정책 예 단원을 참조하세요.

Amazon Polly의 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 있는지 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 타입을 지원하는 작업에 대해 이 작업을 수행할 수 있습니다.

작업 나열과 같이 리소스 수준 권한을 지원하지 않는 작업의 경우, 와일드카드(*)를 사용하여 해당 문이 모든 리소스에 적용됨을 나타냅니다.

```
"Resource": "*"
```

Amazon Polly 리소스 유형 및 해당 ARN의 목록을 보려면 서비스 권한 부여 참조의 Amazon Polly에서 정의한 리소스를 참조하세요. 각 리소스의 ARN을 지정할 수 있는 작업을 알아보려면 Amazon Polly에서 정의한 작업을 참조하세요.

Amazon Polly 자격 증명 기반 정책 예제를 보려면 Amazon Polly의 자격 증명 기반 정책 예 단원을 참조하세요.

Amazon Polly의 정책 조건 키

서비스별 정책 조건 키 지원	아니요

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지 지정할 수 있습니다.

`Condition` 요소(또는 `Condition` 블록)를 사용하면 정책이 발효되는 조건을 지정할 수 있습니다. `Condition` 요소는 옵션입니다. 같거나 작음과 같은 조건 연산자를 사용하여 정책의 조건을 요청의 값과 일치시키는 조건식을 생성할 수 있습니다.

한 문에서 여러 Condition요소를 지정하거나 단일 Condition요소에서 여러 키를 지정하는 경우 AWS 는 논리적 AND태스크를 사용하여 평가합니다. 단일 조건 키에 여러 값을 지정하는 경우 는 논리적 OR 연산을 사용하여 조건을 AWS 평가합니다. 명문의 권한을 부여하기 전에 모든 조건을 충족해야 합니다.

조건을 지정할 때 자리 표시자 변수를 사용할 수도 있습니다. 예를 들어, IAM 사용자에게 IAM 사용자 이름으로 태그가 지정된 경우에만 리소스에 액세스할 수 있는 권한을 부여할 수 있습니다. 자세한 정보는 IAM 사용 설명서의 IAM 정책 요소: 변수 및 태그를 참조하세요.

AWS 글로벌 조건 키 및 서비스별 조건 키를 지원합니다. 모든 AWS 글로벌 조건 키를 보려면 IAM 사용 AWS 설명서의 글로벌 조건 컨텍스트 키를 참조하십시오.

Amazon Polly 조건 키 목록을 보려면 서비스 승인 참조의 Amazon Polly에 사용되는 조건 키를 참조하세요. 조건 키를 사용할 수 있는 작업과 리소스를 알아보려면 Amazon Polly에서 정의한 작업을 참조하세요.

Amazon Polly 자격 증명 기반 정책 예제를 보려면 Amazon Polly의 자격 증명 기반 정책 예 단원을 참조하세요.

Amazon Polly의 ACL

ACL 지원	아니요

액세스 제어 목록(ACLs)은 어떤 보안 주체(계정 멤버, 사용자 또는 역할)가 리소스에 액세스할 수 있는 권한을 가지고 있는 지를 제어합니다. ACLs는 JSON 정책 문서 형식을 사용하지 않지만 리소스 기반 정책과 유사합니다.

Amazon Polly와 ABAC

ABAC 지원(정책의 태그)	아니요

ABAC(속성 기반 액세스 제어)는 속성을 기반으로 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.

태그를 기반으로 액세스를 제어하려면 `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` 또는 `aws:TagKeys` 조건 키를 사용하여 정책의 조건 요소에 태그 정보를 제공합니다.

서비스가 모든 리소스 타입에 대해 세 가지 조건 키를 모두 지원하는 경우, 값은 서비스에 대해 예입니다. 서비스가 일부 리소스 타입에 대해서만 세 가지 조건 키를 모두 지원하는 경우, 값은 부분적입니다.

ABAC에 대한 자세한 정보는 IAM 사용 설명서의 ABAC란 무엇인가요?를 참조하세요. ABAC 설정 단계가 포함된 자습서를 보려면 IAM 사용 설명서의 속성 기반 액세스 제어(ABAC) 사용을 참조하세요.

Amazon Polly에서 임시 자격 증명 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 액세스하는 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 정보는 IAM 사용 설명서의 역할로 전환(콘솔)을 참조하세요.

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 인증 섹션을 참조하세요.

Amazon Polly용 크로스 서비스 포워드 액세스 세션 (FAS)

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 수신한 경우에만 이루어집니다. 이 경우 두 작업을 모두 수행할 수 있는 권한이 있어야 합니다. FAS 요청 시 정책 세부 정보는 전달 액세스 세션을 참조하세요.

Amazon Polly의 서비스 역할

서비스 역할 지원	아니요

서비스 역할은 서비스가 사용자를 대신하여 작업을 수행하기 위해 수임하는 IAM role(IAM 역할)입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 정보는 IAM 사용 설명서의 AWS 서비스에 대한 권한을 위임할 역할 생성을 참조하세요.

> **Warning**
>
> 서비스 역할에 대한 권한을 변경하면 Amazon Polly 기능이 중단될 수 있습니다. Amazon Polly에서 관련 지침을 제공하는 경우에만 서비스 역할을 편집합니다.

Amazon Polly의 서비스 연결 역할

서비스 연결 역할 지원	아니요

서비스 연결 역할은 에 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.

서비스 연결 역할 생성 또는 관리에 대한 자세한 내용은 IAM으로 작업하는AWS 서비스 단원을 참조하세요. 서비스 연결 역할 열에서 Yes가 포함된 서비스를 테이블에서 찾습니다. 해당 서비스에 대한 서비스 연결 역할 설명서를 보려면 Yes(네) 링크를 선택합니다.

Amazon Polly IAM 역할

자격 증명 기반 권한 정책을 IAM 역할에 연결하여 교차 계정 권한을 부여할 수 있습니다. 예를 들어 계정 A의 관리자는 다음과 같이 다른 계정 (예: 계정 B) 또는 AWS 서비스에 교차 AWS 계정 권한을 부여하는 역할을 생성할 수 있습니다.

1. 계정 A 관리자는 IAM 역할을 생성하고 계정 A의 리소스에 대한 권한을 부여하는 권한 정책을 역할에 연결합니다.
2. 계정 A 관리자는 계정 B를 역할을 수임할 보안 주체로 식별하는 역할에 신뢰 정책을 연결합니다.

3. 그런 다음 계정 B 관리자는 계정 B의 모든 사용자에게 역할을 수임할 권한을 위임할 수 있습니다. 이렇게 하면 계정 B의 사용자가 계정 A의 리소스를 만들거나 액세스할 수 있습니다. AWS 서비스에 역할을 수임할 권한을 부여하려는 경우 신뢰 정책의 보안 주체가 될 수도 있습니다. AWS

IAM을 사용하여 권한을 위임하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 액세스 관리를 참조하세요.

다음은 어휘를 입력하거나 가져오고 현재 사용 가능한 어휘를 나열할 수 있는 권한을 부여하는 정책 예제입니다.

Amazon Polly는 리소스 수준에서 작업에 대한 자격 증명 기반 정책을 지원합니다. 경우에 따라 ARN에 의해 리소스가 제한될 수 있습니다. 이는 SynthesizeSpeech, StartSpeechSynthesisTask, PutLexicon, GetLexicon 및 DeleteLexicon 작업에 해당됩니다. 이 경우 Resource 값은 ARN으로 표시됩니다. 예제: Resource 값인 arn:aws:polly:*us-east-2*:*account-id*:lexicon/*은 us-east-2 리전 내에서 소유한 모든 어휘에 대한 권한을 지정합니다.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Sid": "AllowPut-Get-ListActions",
      "Effect": "Allow",
      "Action": [
         "polly:PutLexicon",
         "polly:GetLexicon",
         "polly:ListLexicons"],
      "Resource": "arn:aws:polly:us-east-2:account-id:lexicon/*"
      }
   ]
}
```

그러나 일부 작업에서는 ARN을 사용하지 않습니다. DescribeVoices, ListLexicons, GetSpeechSynthesisTasks 및 ListSpeechSynthesisTasks 작업이 여기에 해당합니다.

사용자, 그룹, 역할 및 권한에 대한 자세한 내용은 IAM 사용 설명서의 자격 증명(사용자, 그룹 및 역할)을 참조하세요.

Amazon Polly의 자격 증명 기반 정책 예

기본적으로 사용자 및 역할은 Amazon Polly 리소스를 생성하거나 수정할 수 있는 권한이 없습니다. 또한 AWS Management Console, AWS Command Line Interface (AWS CLI) 또는 AWS API를 사용하여

작업을 수행할 수도 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 생성을 참조하세요.

각 리소스 유형에 대한 ARN 형식을 포함하여 Amazon Polly에서 정의한 작업 및 리소스 유형에 대한 자세한 내용은 서비스 인증 참조에서 Amazon Polly에 대한 작업, 리소스 및 조건 키를 참조하세요.

주제

- 정책 모범 사례
- Amazon Polly 콘솔 사용
- 사용자가 자신의 고유한 권한을 볼 수 있도록 허용
- AWS Amazon Polly의 관리형 (사전 정의된) 정책
- 고객 관리형 정책 예제

정책 모범 사례

자격 증명 기반 정책에 따라 계정에서 사용자가 Amazon Polly 리소스를 생성, 액세스 또는 삭제할 수 있는지 여부가 결정됩니다. 이 작업으로 인해 AWS 계정에 비용이 발생할 수 있습니다. ID 기반 정책을 생성하거나 편집할 때는 다음 지침과 권장 사항을 따르세요.

- AWS 관리형 정책으로 시작하고 최소 권한 권한으로 이동 — 사용자와 워크로드에 권한을 부여하려면 여러 일반적인 사용 사례에 권한을 부여하는 AWS 관리형 정책을 사용하세요. 해당 내용은 에서 사용할 수 있습니다. AWS 계정사용 사례에 맞는 AWS 고객 관리형 정책을 정의하여 권한을 더 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 태스크를 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. 예를 AWS 서비스들어 특정 작업을 통해 서비스 작업을 사용하는 경우 조건을 사용하여 서비스 작업에 대한 액세스 권한을 부여할 수도 AWS CloudFormation있습니다. 자세한 정보는 IAM 사용 설명서의 IAM JSON 정책 요소: 조건을 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 추천을 제공하여 안전하고 기능적인 정책을 작성하도록 돕습니다. 자세한 정보는 IAM 사용 설명서의 IAM Access Analyzer 정책 검증을 참조하tpdy.
- 멀티 팩터 인증 (MFA) 필요 - IAM 사용자 또는 루트 사용자가 필요한 시나리오가 있는 경우 추가 보안을 위해 AWS 계정 MFA를 활성화하십시오. API 작업을 직접 호출할 때 MFA가 필요하면 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 사례를 참조하세요.

Amazon Polly 콘솔 사용

Amazon Polly 콘솔에 액세스하려면 최소한의 권한 집합이 있어야 합니다. 이러한 권한을 통해 내 Amazon Polly 리소스를 나열하고 해당 리소스에 대한 세부 정보를 볼 수 있어야 합니다. AWS 계정최소 필수 권한보다 더 제한적인 자격 증명 기반 정책을 만들면 콘솔이 해당 정책에 연결된 엔터티(사용자 또는 역할)에 대해 의도대로 작동하지 않습니다.

AWS CLI 또는 AWS API만 호출하는 사용자에게 최소 콘솔 권한을 허용할 필요는 없습니다. 그 대신, 수행하려는 API 작업과 일치하는 작업에만 액세스할 수 있도록 합니다.

사용자와 역할이 Amazon Polly 콘솔을 계속 사용할 수 있도록 하려면 Amazon *ConsoleAccess* Polly *ReadOnly* AWS 또는 관리형 정책도 엔티티에 연결하십시오. 자세한 내용은 IAM 사용 설명서의 사용자에게 권한 추가를 참조하세요.

Amazon Polly 콘솔을 사용하려면 모든 Amazon Polly API에 권한을 부여해야 합니다. 추가 권한은 필요하지 않습니다. 전체 콘솔 기능을 얻으려면 다음 정책을 사용하면 됩니다.

```
{
  "Version": "2012-10-17",
     "Statement": [{
        "Sid": "Console-AllowAllPollyActions",
        "Effect": "Allow",
        "Action": [
           "polly:*"],
        "Resource": "*"
      }
   ]
```

```
}
```

사용자가 자신의 고유한 권한을 볼 수 있도록 허용

이 예시는 IAM 사용자가 자신의 사용자 자격 증명에 연결된 인라인 및 관리형 정책을 볼 수 있도록 허용하는 정책을 생성하는 방법을 보여줍니다. 이 정책에는 콘솔에서 또는 API를 사용하여 프로그래밍 방식으로 이 작업을 완료할 수 있는 AWS CLI 권한이 포함됩니다. AWS

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS Amazon Polly의 관리형 (사전 정의된) 정책

AWS 에서 생성하고 관리하는 독립형 IAM 정책을 제공하여 많은 일반적인 사용 사례를 해결합니다. AWS이러한 AWS 관리형 정책은 일반적인 사용 사례에 필요한 권한을 부여하므로 필요한 권한을 조사하지 않아도 됩니다. 자세한 내용은 IAM 사용 설명서의 AWS 관리형 정책을 참조하세요.

계정의 사용자에게 연결할 수 있는 다음과 같은 AWS 관리형 정책은 Amazon Polly에만 해당됩니다.

- AmazonPollyReadOnlyAccess— 리소스에 대한 읽기 전용 액세스 권한을 부여하고, 어휘를 나열하고, 어휘를 가져오고, 사용 가능한 음성을 나열하고, 음성을 합성할 수 있습니다 (합성된 음성에 사전 적용 포함).
- AmazonPollyFullAccess— 리소스 및 지원되는 모든 작업에 대한 전체 액세스 권한을 부여합니다.

> **Note**
>
> IAM 콘솔에 로그인하고 이 콘솔에서 특정 정책을 검색하여 이러한 권한 정책을 검토할 수 있습니다.

Amazon Polly 작업 및 리소스에 대한 권한을 허용하는 고유의 사용자 지정 IAM 정책을 생성할 수도 있습니다. 해당 권한이 필요한 IAM 사용자 또는 그룹에 이러한 사용자 지정 정책을 연결할 수 있습니다.

고객 관리형 정책 예제

이 섹션에서는 다양한 Amazon Polly 작업에 대한 권한을 부여하는 사용자 정책의 예를 제공합니다. 이러한 정책은 AWS SDK 또는 를 AWS CLI사용할 때 작동합니다. 콘솔을 사용할 때는 모든 Amazon Polly API에 권한을 부여하십시오.

> **Note**
>
> 모든 예제는 us-east-2 리전을 사용하며 가상의 계정 ID를 포함합니다.

예제

- 예제 1: 모든 Amazon Polly 작업 허용
- 예 2: 다음을 제외한 모든 Amazon Polly 작업 허용 DeleteLexicon
- 예 3: 허용 DeleteLexicon

- 예제 4: 지정된 리전에서 어휘 삭제 허용
- 예 5: 지정된 어휘집에 DeleteLexicon 대해 허용

예제 1: 모든 Amazon Polly 작업 허용

가입한 후(Amazon Polly 설정 참조) 사용자 생성, 권한 관리를 비롯한 계정 관리 작업을 수행할 수 있는 관리자 사용자를 생성합니다.

모든 Amazon Polly 작업에 대한 권한이 있는 사용자를 생성할 수 있습니다. 이 사용자가 Amazon Polly로 작업하는 서비스별 관리자라고 생각하세요. 이 사용자에게 다음 권한 정책을 연결할 수 있습니다.

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

예 2: 다음을 제외한 모든 Amazon Polly 작업 허용 DeleteLexicon

다음 권한 정책은 모든 리전에서 명시적으로 거부되는 삭제 권한을 포함하여 모든 작업(DeleteLexicon 제외)을 수행할 수 있는 권한을 사용자에게 부여합니다.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Sid": "AllowAllActions-DenyDelete",
      "Effect": "Allow",
      "Action": [
         "polly:DescribeVoices",
         "polly:GetLexicon",
         "polly:PutLexicon",
         "polly:SynthesizeSpeech",
         "polly:ListLexicons"],
      "Resource": "*"
      }
      {
```

```
      "Sid": "DenyDeleteLexicon",
      "Effect": "Deny",
      "Action": [
         "polly:DeleteLexicon"],
      "Resource": "*"
      }
   ]
}
```

예 3: 허용 DeleteLexicon

다음 권한 정책은 위치한 리전 또는 프로젝트에 상관없이 소유한 모든 어휘를 삭제할 수 있는 권한을 부여합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [{
      "Sid": "AllowDeleteLexicon",
      "Effect": "Allow",
      "Action": [
         "polly:DeleteLexicon"],
      "Resource": "*"
      }
   ]
}
```

예제 4: 지정된 리전에서 어휘 삭제 허용

다음 권한 정책은 단일 리전(이 경우 us-east-2)에서 소유한 모든 프로젝트에서 어휘를 삭제할 수 있는 권한을 부여합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [{
      "Sid": "AllowDeleteSpecifiedRegion",
      "Effect": "Allow",
      "Action": [
         "polly:DeleteLexicon"],
      "Resource": "arn:aws:polly:us-east-2:123456789012:lexicon/*"
      }
   ]
}
```

예 5: 지정된 어휘집에 DeleteLexicon 대해 허용

다음 권한 정책은 특정 리전(이 경우 us-east-2)에서 소유한 특정 어휘(이 경우 myLexicon)를 삭제할 수 있는 권한을 부여합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [{
      "Sid": "AllowDeleteForSpecifiedLexicon",
      "Effect": "Allow",
      "Action": [
         "polly:DeleteLexicon"],
      "Resource": "arn:aws:polly:us-east-2:123456789012:lexicon/myLexicon"
      }
   ]
}
```

Amazon Polly API 권한: 작업, 권한 및 리소스 참조

IAM ID에 연결할 수 있는 권한 정책 (ID 기반 정책) 을 설정할 때 다음 목록을 참조로 사용할 수 있습니다. 목록에는 각 Amazon Polly API 작업, 작업을 수행할 권한을 부여할 수 있는 해당 작업, 권한을 부여할 수 있는 AWS 리소스가 포함되어 있습니다. 정책의 `Action`필드에서 작업을 지정하고, 정책의 `Resource`필드에서 리소스 값을 지정합니다.

Amazon Polly 정책에서 AWS-wide 조건 키를 사용하여 조건을 표현할 수 있습니다. AWS-wide 키의 전체 목록은 IAM 사용 설명서에서 사용 가능한 키를 참조하십시오.

> **Note**
>
> 작업을 지정하려면 `polly` 접두사 다음에 API 작업 명칭을 사용합니다(예: `polly:GetLexicon`).

Amazon Polly는 리소스 수준에서 작업에 대한 자격 증명 기반 정책을 지원합니다. 따라서 `Resource` 값은 ARN으로 표시됩니다. 예제: `Resource` 값인 `arn:aws:polly:`*`us-east-2`*`:`*`account-id`*`:lexicon/*`은 `us-east-2` 리전 내에서 소유한 모든 어휘에 대한 권한을 지정합니다.

Amazon Polly는 리소스 수준의 작업에 대한 권한을 지원하지 않으므로, 대부분의 정책에서는 와일드카드 문자(*)를 `Resource` 값으로 지정합니다. 하지만 권한을 특정 리전으로 제한해야 하는 경우 이 와

일드카드 문자를 적절한 ARN(`arn:aws:polly:region:account-id:lexicon/*` .)으로 대체합니다.

Amazon Polly API 및 작업에 대한 필수 권한

API 작업: DeleteLexicon

필요한 권한(API 작업): `polly:DeleteLexicon`

리소스: `arn:aws:polly:region:account-id:lexicon/LexiconName`

API 작업: DescribeVoices

필요한 권한(API 작업): `polly:DescribeVoices`

리소스: `arn:aws:polly:region:account-id:lexicon/voice-name`

API 작업: GetLexicon

필요한 권한(API 작업): `polly:GetLexicon`

리소스: `arn:aws:polly:region:account-id:lexicon/voice-name`

API 작업: ListLexicons

필요한 권한(API 작업): `polly:ListLexicons`

리소스: `arn:aws:polly:region:account-id:lexicon/*`

API 작업: PutLexicon

필요한 권한(API 작업): `polly:ListLexicons`

리소스: *

API 작업: SynthesizeSpeech

필요한 권한(API 작업): `polly:SynthesizeSpeech`

리소스: *

Amazon Polly 자격 증명 및 액세스 문제 해결

다음 정보를 사용하여 Amazon Polly 및 IAM으로 작업할 때 발생할 수 있는 일반적인 문제를 진단하고 수정할 수 있습니다.

주제

- Amazon Polly에서 작업을 수행할 권한이 없음
- 저는 IAM을 수행할 권한이 없습니다. PassRole
- 외부 사용자가 내 Amazon Polly AWS 계정 리소스에 액세스할 수 있도록 허용하고 싶습니다.

Amazon Polly에서 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 polly:*GetWidget*권한이 없을 때 발생합니다.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 polly:GetWidget on resource: my-example-widget
```

이 경우 polly:*GetWidget*작업을 사용하여 *my-example-widget*리소스에 액세스할 수 있도록 mateojackson사용자 정책을 업데이트해야 합니다.

도움이 필요하면 관리자에게 문의하세요. AWS 관리자는 로그인 자격 증명을 제공한 사람입니다.

저는 IAM을 수행할 권한이 없습니다. PassRole

iam:PassRole 작업을 수행할 수 있는 권한이 없다는 오류가 수신되면 Amazon Polly에 역할을 전달할 수 있도록 정책을 업데이트해야 합니다.

새 서비스 역할 또는 서비스 연결 역할을 만드는 대신 기존 역할을 해당 서비스에 전달할 AWS 서비스 수 있는 기능도 있습니다. 이렇게 하려면 사용자가 서비스에 역할을 전달할 수 있는 권한을 가지고 있어야 합니다.

다음에 예로 보인 오류는 marymajor라는 IAM 사용자가 콘솔을 사용하여 Amazon Polly에서 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:PassRole작업을 수행할 수 있도록 Mary의 정책을 업데이트해야 합니다.

도움이 필요하면 관리자에게 문의하세요. AWS 관리자는 로그인 자격 증명을 제공한 사람입니다.

외부 사용자가 내 Amazon Polly AWS 계정 리소스에 액세스할 수 있도록 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 참조하세요.

- Amazon Polly에서 이러한 기능을 지원하는지 여부를 알아보려면 Amazon Polly에서 IAM을 사용하는 방법 단원을 참조하세요.
- 소유하고 AWS 계정 있는 모든 리소스에 대한 액세스 권한을 AWS 계정 부여하는 방법을 알아보려면 IAM 사용 설명서의 다른 IAM 사용자에게 액세스 권한 제공을 참조하십시오.
- 제3자에게 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 타사 AWS 계정AWS 계정 소유에 대한 액세스 제공을 참조하십시오.
- ID 페더레이션을 통해 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 외부에서 인증된 사용자에게 액세스 권한 제공(자격 증명 연동)을 참조하세요.
- 크로스 계정 액세스를 위한 역할과 리소스 기반 정책 사용의 차이점을 알아보려면 IAM 사용 설명서의 IAM 역할과 리소스 기반 정책의 차이를 참조하세요.

Amazon Polly의 로깅 및 모니터링

모니터링은 Amazon Polly 애플리케이션의 안정성, 가용성 및 성능을 유지하는 중요한 역할을 합니다. Amazon Polly API 호출을 모니터링하려면 를 사용할 수 있습니다. AWS CloudTrail작업 상태를 모니터링하려면 Amazon CloudWatch Logs를 사용하십시오.

- Amazon CloudWatch Alarms — CloudWatch 경보를 사용하면 지정한 기간 동안 단일 지표를 관찰할 수 있습니다. 지표가 지정된 임계값을 초과하는 경우 Amazon Simple Notification Service 주제 또는 AWS Auto Scaling 정책으로 알림이 전송됩니다. CloudWatch지표가 특정 상태에 있을 때는 경보가 작업을 호출하지 않습니다. 대신, 상태가 변경되어 지정된 기간 동안 유지되어야 합니다. 자세한 정보는 아마존 CloudWatch Polly와 통합하기을 참조하세요.
- CloudTrail logs — Amazon Polly에서 사용자, 역할 또는 AWS 서비스가 수행한 작업의 기록을 CloudTrail 제공합니다. 에서 수집한 정보를 사용하여 Amazon Polly에 요청한 내용을 확인할 수 있습니다. CloudTrail 또한 어떤 IP 주소에서 요청했는지, 누가 언제 요청했는지 등의 추가 세부 정보

도 확인할 수 있습니다. 자세한 내용은 를 사용하여 Amazon Polly API 호출을 로깅합니다. AWS CloudTrail을(를) 참조하세요.

Amazon Polly에 대한 규정 준수 확인

타사 감사자는 AWS 여러 규정 준수 프로그램의 일환으로 Amazon Polly의 보안 및 규정 준수를 평가합니다. 여기에는 SOC, PCI, FedRAMP, HIPAA 등이 포함됩니다.

특정 규정 준수 프로그램 범위 내 AWS 서비스 목록은 규정 준수 프로그램별 범위 내AWS 서비스 규정 준수 참조하십시오. 일반 정보는 AWS 규정 준수 프로그램AWS 보증 프로그램 규정AWS 참조하십시오.

를 사용하여 AWS Artifact타사 감사 보고서를 다운로드할 수 있습니다. 자세한 내용은 AWS Artifact에서 보고서 다운로드를 참조하십시오.

Amazon Polly를 사용할 때의 규정 준수 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 빠른 시작 안내서: 이 배포 안내서에서는 아키텍처 고려 사항에 관해 설명하고 AWS에서 보안 및 규정 준수에 중점을 둔 기본 환경을 배포하기 위한 단계를 제공합니다.
- HIPAA 보안 및 규정 준수를 위한 설계 백서 — 이 백서는 기업이 HIPAA 준수 애플리케이션을 개발하는 데 사용할 AWS 수 있는 방법을 설명합니다.
- AWS 규정 준수 리소스 규정 — 이 통합 문서 및 가이드 모음은 해당 산업 및 지역에 적용될 수 있습니다.
- AWS Config 개발자 안내서의 규칙을 통한 리소스 평가 — 이 AWS Config 서비스는 리소스 구성이 내부 관행, 업계 지침 및 규정을 얼마나 잘 준수하는지 평가합니다.
- AWS Security Hub— 이 AWS 서비스는 보안 업계 표준 및 모범 사례를 준수하는지 확인하는 데 도움이 AWS 되는 내부 보안 상태를 종합적으로 보여줍니다.

Amazon Polly의 복원성

AWS 글로벌 인프라는 지역 및 가용 AWS 영역을 중심으로 구축됩니다. AWS 지역은 물리적으로 분리되고 격리된 여러 가용 영역을 제공하며, 이러한 가용 영역은 지연 시간이 짧고 처리량이 높으며 중복성이 높은 네트워킹으로 연결됩니다. 가용 영역을 사용하면 중단 없이 영역 간에 자동으로 장애 극복 조치가 이루어지는 애플리케이션 및 데이터베이스를 설계하고 운영할 수 있습니다. 가용 영역은 기존의 단일 또는 다중 데이터 센터 인프라보다 가용성, 내결함성, 확장성이 뛰어납니다.

AWS 지역 및 가용 영역에 대한 자세한 내용은 글로벌 인프라를 참조하십시오AWS .

Amazon Polly의 인프라 보안

관리형 서비스인 Amazon Polly는 Amazon Web Services: 보안 프로세스 개요 백서에 설명된 AWS 글로벌 네트워크 보안 절차에 따라 보호됩니다.

AWS 게시된 API 호출을 사용하여 네트워크를 통해 Amazon Polly에 액세스할 수 있습니다. 클라이언트가 전송 계층 보안(TLS) 1.0 이상을 지원해야 합니다. TLS 1.2 이상을 권장합니다. 클라이언트는 Ephemeral Diffie-Hellman(DHE) 또는 Elliptic Curve Ephemeral Diffie-Hellman(ECDHE)과 같은 PFS(전달 완전 보안, Perfect Forward Secrecy)가 포함된 암호 제품군도 지원해야 합니다. Java 7 이상의 최신 시스템은 대부분 이러한 모드를 지원합니다.

또한 요청은 액세스 키 ID 및 IAM 주체와 관련된 비밀 액세스 키를 사용하여 서명해야 합니다. 또는 AWS Security Token Service(AWS STS)를 사용하여 임시 보안 인증을 생성하여 요청에 서명할 수 있습니다.

Amazon Polly의 보안 모범 사례

고객의 신뢰, 개인 정보 보호 및 콘텐츠 보안은 당사의 최우선 과제입니다. 당사는 사용자 콘텐츠에 대한 무단 액세스 또는 공개를 방지하고 콘텐츠 사용이 사용자에 대한 당사의 약속을 준수하도록 보장하기 위해 책임감 있고 정교한 기술적 및 물리적 제어를 구현합니다. 자세한 내용은 AWS 데이터 프라이버시 FAQ를 참조하세요.

Amazon Polly는 텍스트 제출의 콘텐츠를 보관하지 않습니다.

규정 준수, 침투 테스트, 게시판, 리소스 등 보안에 대한 폭넓은 관점을 보려면 Cloud AWS Security 웹 사이트를 방문하십시오.AWS

인터페이스 VPC 엔드포인트에서 Amazon Polly 사용

Amazon VPC (가상 사설 클라우드) 를 사용하여 AWS 리소스를 호스팅하는 경우, VPC와 Amazon Polly 사이에 사설 연결을 설정할 수 있습니다. 이 연결을 사용하여 공용 인터넷을 통과하지 않고 Amazon Polly에서 스피치를 합성할 수 있습니다.

Amazon VPC는 사용자가 AWS 정의한 가상 네트워크에서 AWS 리소스를 시작하는 데 사용할 수 있는 서비스입니다. VPC가 있으면 IP 주소 범위, 서브넷, 라우팅 테이블, 네트워크 게이트웨이 등 네트워

크 설정을 제어할 수 있습니다. VPC를 Amazon Polly에 연결하려면 Amazon Polly용 인터페이스 VPC 엔드포인트를 정의하세요. 이 유형의 엔드포인트를 사용하여 VPC를 AWS 서비스에 연결할 수 있습니다. 이 엔드포인트를 이용하면 인터넷 게이트웨이나 NAT(네트워크 주소 변환) 인스턴스 또는 VPN 연결 없이도 Amazon Polly에 안정적이고 확장 가능하게 연결됩니다. 자세한 내용은 Amazon VPC 사용 설명서의 Amazon VPC란 무엇입니까?를 참조하세요.

인터페이스 VPC 엔드포인트는 사설 IP 주소가 있는 Elastic Network Interface AWS 서비스 사용 간에 사설 통신을 가능하게 하는 AWS 기술인 에 의해 AWS PrivateLink구동됩니다. 자세한 내용은 New - AWS PrivateLink for를 참조하십시오. AWS 서비스

다음은 Amazon VPC 사용자를 위한 단계들입니다. 자세한 내용은 Amazon VPC 사용 설명서의 시작하기를 참조하세요.

가용성

VPC 엔드포인트는 Amazon Polly가 지원되는 모든 리전에서 지원됩니다. AWS 지역 및 가용 영역에 대한 자세한 내용은 AWS 글로벌 인프라를 참조하십시오.

Amazon Polly에 대한 VPC 엔드포인트 정책 생성

VPC에서 Amazon Polly를 사용하기 시작하려면 Amazon Polly에 대한 인터페이스 VPC 엔드포인트를 생성합니다. 선택할 서비스 이름은 com.amazonaws.***Region***.polly입니다. Amazon Polly의 설정은 변경할 필요가 없습니다. 자세한 내용은 Amazon VPC 사용 설명서의 인터페이스 엔드포인트 생성을 참조하세요.

VPC와 Amazon Polly 간의 연결 테스트

엔드포인트를 생성한 후에 연결을 테스트할 수 있습니다.

VPC와 Amazon Polly 엔드포인트 간 연결을 테스트하려면

1. VPC에 있는 Amazon EC2 인스턴스에 연결합니다. 연결에 대한 자세한 내용은 Amazon EC2 설명서의 Linux 인스턴스에 연결 또는 Windows 인스턴스에 연결을 참조하세요.
2. 인스턴스의 AWS CLI 에서 `aws polly describe-voices`를 사용하여 사용 가능한 Amazon Polly 음성을 나열합니다.

명령에 대한 응답에 사용 가능한 Amazon Polly 음성 목록이 포함되어 있다면 명령이 성공하고 VPC 엔드포인트가 작동합니다.

Amazon Polly 엔드포인트에 대한 액세스 제어

VPC 엔드포인트 정책은 엔드포인트를 만들거나 수정 시 엔드포인트에 연결하는 IAM 리소스 정책입니다. 엔드포인트를 만들 때 정책을 추가하지 않으면 서비스에 대한 모든 액세스를 허용하는 기본 정책이 추가됩니다. 엔드포인트 정책은 IAM 사용자 정책 또는 서비스별 정책을 재정의하거나 대체하지 않습니다. 이는 엔드포인트에서 지정된 서비스로의 액세스를 제어하기 위한 별도의 정책입니다.

엔드포인트 정책은 JSON 형식으로 작성해야 합니다.

자세한 내용은 Amazon VPC 사용 설명서의 VPC 엔드포인트를 통해 서비스에 대한 액세스 제어를 참조하세요.

다음은 Amazon Polly에 대한 엔드포인트 정책의 예입니다. 이 정책은 VPC를 통해 Amazon Polly에 연결한 사용자가 Amazon Polly를 통해 스피치를 설명하고 합성하도록 허용하고 다른 Amazon Polly 작업을 수행하지 못하게 금지합니다.

```
{
  "Statement": [
    {
      "Sid": "SynthesisAndDescribeVoicesOnly",
      "Principal": "*",
      "Action": [
        "polly:DescribeVoices",
        "polly:SynthesizeSpeech"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Amazon Polly에 대한 VPC 엔드포인트 정책을 수정하는 방법

1. https://console.aws.amazon.com/vpc 에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 엔드포인트를 선택합니다.
3. Amazon Polly에 대한 엔드포인트를 아직 생성하지 않았다면 엔드포인트 생성을 선택합니다. 그런 다음 com.amazonaws.***Region***.polly를 선택하고 엔드포인트 생성을 선택합니다.
4. com.amazonaws.***Region***.polly 엔드포인트를 선택하고 화면 하단의 정책 탭을 선택합니다.
5. 정책 편집(Edit Policy)을 선택하고 정책을 변경합니다.

VPC 컨텍스트 키에 대한 지원

Amazon Polly는 특정 VPC 또는 특정 VPC 엔드포인트에 대한 액세스를 제한할 수 있는 `aws:SourceVpc` 및 `aws:SourceVpce` 컨텍스트 키를 지원합니다. 이러한 키는 사용자가 VPC 엔드포인트를 사용하고 있을 때만 작동합니다. 자세한 내용은 IAM 사용 설명서의 일부 서비스에 사용 가능한 키를 참조하세요.

를 사용하여 Amazon Polly API 호출을 로깅합니다. AWS CloudTrail

Amazon Polly는 Amazon Polly에서 사용자 AWS CloudTrail, 역할 또는 서비스가 수행한 작업의 기록을 제공하는 AWS 서비스와 통합되어 있습니다. CloudTrail Amazon Polly에 대한 모든 API 호출을 이벤트로 캡처합니다. 캡처되는 호출에는 Amazon Polly 콘솔로부터의 호출과 Amazon Polly API 작업에 대한 코드 호출이 포함됩니다. 트레일을 생성하면 Amazon Polly용 CloudTrail 이벤트를 포함하여 Amazon S3 버킷으로 이벤트를 지속적으로 전송할 수 있습니다. 트레일을 구성하지 않아도 CloudTrail 콘솔의 이벤트 기록에서 가장 최근 이벤트를 계속 볼 수 있습니다. 에서 수집한 CloudTrail 정보를 사용하여 Amazon Polly에 이루어진 요청, 요청이 이루어진 IP 주소, 요청한 사람, 요청 시기 및 추가 세부 정보를 확인할 수 있습니다.

구성 및 활성화 방법을 CloudTrail 포함하여 자세한 내용은 사용 AWS CloudTrail 설명서를 참조하십시오.

Amazon Polly 정보: CloudTrail

CloudTrail 계정을 생성하면 AWS 계정에서 활성화됩니다. Amazon Polly에서 지원되는 이벤트 활동이 발생하면 해당 활동이 이벤트 기록의 다른 AWS 서비스 CloudTrail 이벤트와 함께 이벤트에 기록됩니다. AWS 계정에서 최근 이벤트를 보고, 검색하고, 다운로드할 수 있습니다. 자세한 내용은 이벤트 기록으로 CloudTrail 이벤트 보기를 참조하십시오.

Amazon Polly의 이벤트를 포함하여 AWS 계정에서 진행 중인 이벤트 기록을 보려면 트레일을 생성하십시오. 트레일을 사용하면 CloudTrail Amazon S3 버킷으로 로그 파일을 전송할 수 있습니다. 기본적으로 콘솔에서 트레일을 생성하면 트레일이 모든 AWS 지역에 적용됩니다. 트레일은 AWS 파티션에 있는 모든 지역의 이벤트를 기록하고 지정한 Amazon S3 버킷으로 로그 파일을 전송합니다. 또한 CloudTrail 로그에서 수집된 이벤트 데이터를 추가로 분석하고 이에 따라 조치를 취하도록 다른 AWS 서비스를 구성할 수 있습니다. 자세한 내용은 다음을 참조하십시오.

- 추적 생성 개요
- CloudTrail 지원되는 서비스 및 통합
- 에 대한 Amazon SNS 알림 구성 CloudTrail
- 여러 지역에서 CloudTrail 로그 파일 수신 및 여러 계정으로부터 CloudTrail 로그 파일 수신

Amazon Polly는 다음 작업을 CloudTrail 로그 파일에 이벤트로 기록하는 것을 지원합니다.

- DeleteLexicon
- DescribeVoices
- GetLexicon
- GetSpeechSynthesisTask
- ListLexicons
- ListSpeechSynthesisTasks
- PutLexicon
- StartSpeechSynthesisTask
- SynthesizeSpeech

모든 이벤트 및 로그 항목에는 요청을 생성한 사용자에 대한 정보가 들어 있습니다. 신원 정보를 이용하면 다음을 쉽게 알아볼 수 있습니다.

- 요청이 루트 사용자 자격 증명으로 이루어졌는지 AWS Identity and Access Management (IAM) 사용자 자격 증명으로 이루어졌는지 여부.
- 역할 또는 연동 사용자를 위한 임시 보안 인증으로 요청을 생성하였는지.
- 다른 AWS 서비스에서 요청했는지 여부.

자세한 내용은 CloudTrail 사용자 ID 요소를 참조하십시오.

예: Amazon Polly 로그 파일 항목

트레일은 지정한 Amazon S3 버킷에 이벤트를 로그 파일로 전송할 수 있는 구성입니다. CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함되어 있습니다. 이벤트는 모든 소스의 단일 요청을 나타내며 요청된 작업, 작업 날짜 및 시간, 요청 매개 변수 등에 대한 정보를 포함합니다. CloudTrail 로그 파일은 공개 API 호출의 정렬된 스택 트레이스가 아니므로 특정 순서로 표시되지 않습니다.

다음 예제는 를 보여주는 CloudTrail 로그 항목을 보여줍니다. `SynthesizeSpeech`

```
{
"Records": [
        {
"awsRegion": "us-east-2",
            "eventID": "19bd70f7-5e60-4cdc-9825-936c552278ae",
            "eventName": "SynthesizeSpeech",
            "eventSource": "polly.amazonaws.com",
```

```
            "eventTime": "2016-11-02T03:49:39Z",
            "eventType": "AwsApiCall",
            "eventVersion": "1.05",
            "recipientAccountId": "123456789012",
            "requestID": "414288c2-a1af-11e6-b17f-d7cfc06cb461",
            "requestParameters": {
"lexiconNames": [
                    "SampleLexicon"
                ],
                "engine": "neural",
                "outputFormat": "mp3",
                "sampleRate": "22050",
                "text": "**********",
                "textType": "text",
                "voiceId": "Kendra"
            },
            "responseElements": null,
            "sourceIPAddress": "1.2.3.4",
            "userAgent": "Amazon CLI/Polly 1.10 API 2016-06-10",
            "userIdentity": {
"accessKeyId": "EXAMPLE_KEY_ID",
                "accountId": "123456789012",
                "arn": "arn:aws:iam::123456789012:user/Alice",
                "principalId": "EX_PRINCIPAL_ID",
                "type": "IAMUser",
                "userName": "Alice"
            }
        }

    ]
}
```

아마존 CloudWatch Polly와 통합하기

Amazon Polly와 상호 작용하면 다음 지표와 차원이 CloudWatch 1분마다 전송됩니다. 다음 절차에 따라 Amazon Polly에 대한 지표를 볼 수 있습니다.

Amazon Polly의 원시 데이터를 수집하여 읽기 쉬운 거의 실시간 지표로 처리하는 를 사용하여 CloudWatch Amazon Polly를 모니터링할 수 있습니다. 이러한 통계는 2주간 기록되므로 `historical information`에 액세스해 웹 애플리케이션이나 서비스가 어떻게 실행되고 있는지 전체적으로 더 잘 파악할 수 있습니다. 기본적으로 Amazon Polly 측정치 데이터는 1분 간격으로 CloudWatch 전송됩니다. 자세한 내용은 Amazon CloudWatch 사용 설명서의 CloudWatch Amazon이란 무엇입니까? 를 참조하십시오.

CloudWatch 지표 가져오기 (콘솔)

1. https://console.aws.amazon.com/cloudwatch/ 에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 지표(Metrics)를 선택합니다.
3. 범주별 CloudWatch 지표 창의 Amazon Polly의 지표 범주에서 지표 범주를 선택한 다음 위쪽 창에서 아래로 스크롤하여 전체 지표 목록을 확인합니다.

에 대한 CloudWatch 지표 가져오기 AWS CLI

다음 코드는 Amazon Polly에 사용할 수 있는 지표를 표시합니다.

```
aws cloudwatch list-metrics --namespace "AWS/Polly"
```

위의 명령은 다음과 유사한 Amazon Polly에 대한 지표 목록을 반환합니다. `MetricName` 요소는 지표가 무엇인지 식별합니다.

```
{
    "Metrics": [
        {
            "Namespace": "AWS/Polly",
            "Dimensions": [
                {
                    "Name": "Operation",
                    "Value": "SynthesizeSpeech"
                }
```

```
            ],
            "MetricName": "ResponseLatency"
        },
        {
            "Namespace": "AWS/Polly",
            "Dimensions": [
                {
                    "Name": "Operation",
                    "Value": "SynthesizeSpeech"
                }
            ],
            "MetricName": "RequestCharacters"
        }
```

자세한 내용은 Amazon CloudWatch API 레퍼런스를 참조하십시오 GetMetricStatistics.

Amazon Polly 지표

Amazon Polly는 각 요청에 대한 다음 지표를 만듭니다. 이러한 지표는 집계되어 1분 간격으로 사용 가능한 CloudWatch 곳으로 전송됩니다.

지표	설명
RequestCharacters	요청의 문자 수. 청구 가능한 문자이며 SSML 태그가 포함되지 않습니다. 유효한 차원: Operation 유효한 통계: 최소값, 최대값, 평균 SampleCount, 합계 단위: 수
ResponseLatency	요청 시간과 스트리밍 응답이 시작되는 시간 사이의 지연 시간 유효한 차원: Operation 유효한 통계: 최소, 최대, 평균, SampleCount 단위: 밀리초

지표	설명
2XXCount	성공적 응답 시 반환되는 HTTP 200 레벨 코드 유효한 차원: Operation 유효 통계: 평균 SampleCount, 합계 단위: 수
4XXCount	오류 발생 시 반환되는 HTTP 400 레벨 오류 코드. 성공적 응답이 발생할 때마다 0을 내보냅니다. 유효한 차원: Operation 유효 통계: 평균 SampleCount, 합계 단위: 수
5XXCount	오류 발생 시 반환되는 HTTP 500 레벨 오류 코드. 성공적 응답이 발생할 때마다 0을 내보냅니다. 유효한 차원: Operation 유효 통계: 평균 SampleCount, 합계 단위: 수

Amazon Polly 지표의 차원

Amazon Polly 지표는 AWS/Polly 네임스페이스를 사용하며 다음 차원에 대한 지표를 제공합니다.

측정기준	설명
Operation	지표는 참조하는 API 메서드별로 그룹화됩니다. 가능한 값은 SynthesizeSpeech , PutLexicon , DescribeVoices 등입니다.

Amazon Polly API 참조

이 섹션에는 Amazon Polly API 참조가 포함되어 있습니다.

> **Note**
>
> 인증된 API 호출은 서명 버전 4 서명 프로세스를 사용하여 서명해야 합니다. 자세한 내용은 의 AWS API 요청 서명을 참조하십시오. Amazon Web Services 일반 참조

주제

- 작업
- 데이터 유형

작업

다음 작업이 지원됩니다.

- DeleteLexicon
- DescribeVoices
- GetLexicon
- GetSpeechSynthesisTask
- ListLexicons
- ListSpeechSynthesisTasks
- PutLexicon
- StartSpeechSynthesisTask
- SynthesizeSpeech

DeleteLexicon

AWS 리전에 저장된 지정된 발음 어휘를 삭제합니다. 삭제된 어휘는 스피치 합성에 사용할 수 없으며 GetLexicon 또는 ListLexicon API를 사용하여 검색할 수도 없습니다.

자세한 내용은 어휘 관리를 참조하세요.

Request Syntax

```
DELETE /v1/lexicons/LexiconName HTTP/1.1
```

URI 요청 파라미터

요청은 다음 URI 파라미터를 사용합니다.

LexiconName

삭제할 어휘의 이름입니다. 해당 리전에 존재하는 어휘여야 합니다.

패턴: [0-9A-Za-z]{1,20}

필수 사항 여부: Yes

Request Body

해당 요청에는 본문이 없습니다.

Response Syntax

```
HTTP/1.1 200
```

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반송합니다.

Errors

LexiconNotFoundException

Amazon Polly에서 지정한 어휘를 찾을 수 없습니다. 이는 어휘가 없거나, 이름의 철자가 틀렸거나, 다른 리전에 있는 어휘를 지정했기 때문일 수 있습니다.

어휘가 존재하고, 해당 리전(ListLexicons 참조)에 있고, 입력한 이름의 철자가 올바른지 확인하세요. 그런 다음 다시 시도하세요.

HTTP 상태 코드: 404

ServiceFailureException

알 수 없는 상태로 인해 서비스 장애가 발생했습니다.

HTTP 상태 코드: 500

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

DescribeVoices

스피치 합성을 요청할 때 사용할 수 있는 음성 목록을 반환합니다. 각 음성은 남성 또는 여성의 목소리로 지정된 언어를 사용하며 음성 이름의 ASCII 버전인 ID로 식별합니다.

스피치(SynthesizeSpeech)를 합성할 때는 DescribeVoices에서 반환한 음성 목록에서 원하는 음성의 음성 ID를 입력합니다.

예를 들어 뉴스를 읽어주는 애플리케이션에서 특정 언어로 뉴스를 읽도록 하되 사용자에게 음성을 선택할 수 있는 옵션을 제공할 수 있습니다. 이 DescribeVoices 작업을 사용하여 선택할 수 있는 음성 목록을 사용자에게 제공할 수 있습니다.

필요한 경우 언어 코드를 지정하여 사용 가능한 음성을 필터링할 수 있습니다. 예를 들어, en-US를 지정하는 경우 작업은 사용 가능한 모든 미국 영어 음성의 목록을 반환합니다.

이 작업에는 polly:DescribeVoices 조치를 수행할 권한이 요구됩니다.

Request Syntax

```
GET /v1/voices?
Engine=Engine&IncludeAdditionalLanguageCodes=IncludeAdditionalLanguageCodes&LanguageCode=Langua
 HTTP/1.1
```

URI 요청 파라미터

요청은 다음 URI 파라미터를 사용합니다.

Engine

Amazon Polly에서 음성 합성을 위한 입력 텍스트를 처리할 때 사용하는 엔진 (standard, neural, long-form 또는generative) 을 지정합니다.

유효 값: standard | neural | long-form | generative

IncludeAdditionalLanguageCodes

지정한 언어를 추가 언어로 사용하는 이중 언어 음성의 반환 여부를 나타내는 부울 값입니다. 예를 들어 미국 영어(es-US)를 사용하는 모든 언어를 요청했는데 이탈리아어(it-IT)와 미국 영어를 모두 구사하는 이탈리아어 음성이 있는 경우 yes를 지정하는 경우 해당 음성이 포함되지만 no를 지정하면 포함되지 않습니다.

LanguageCode

반환할 음성 목록을 필터링하기 위한 언어 식별 태그(언어 이름의 경우 ISO 639 코드-ISO 3166 국가 코드)입니다. 이 옵션 파라미터를 지정하지 않으면 사용 가능한 모든 음성을 반환합니다.

유효 값: `arb | cmn-CN | cy-GB | da-DK | de-DE | en-AU | en-GB | en-GB-WLS | en-IN | en-US | es-ES | es-MX | es-US | fr-CA | fr-FR | is-IS | it-IT | ja-JP | hi-IN | ko-KR | nb-NO | nl-NL | pl-PL | pt-BR | pt-PT | ro-RO | ru-RU | sv-SE | tr-TR | en-NZ | en-ZA | ca-ES | de-AT | yue-CN | ar-AE | fi-FI | en-IE | nl-BE | fr-BE`

NextToken

이전 `DescribeVoices` 작업에서 반환한 불투명한 페이지 매김 토큰입니다. 존재하는 경우 이는 목록을 계속해야 할 위치를 나타냅니다.

길이 제약: 최소 길이는 0입니다. 최대 길이는 4096자입니다.

Request Body

해당 요청에는 본문이 없습니다.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "NextToken": "string",
   "Voices": [
      {
         "AdditionalLanguageCodes": [ "string" ],
         "Gender": "string",
         "Id": "string",
         "LanguageCode": "string",
         "LanguageName": "string",
         "Name": "string",
         "SupportedEngines": [ "string" ]
      }
   ]
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

NextToken

음성 목록을 계속하기 위해 다음 요청에 사용할 페이지 매김 토큰입니다. 응답이 생략되지 않는 경우에만 NextToken을 반환합니다.

타입: 문자열

길이 제약: 최소 길이는 0입니다. 최대 길이는 4096자입니다.

Voices

음성 및 해당 속성 목록입니다.

타입: Voice 객체 배열

Errors

InvalidNextTokenException

NextToken 유효하지 않습니다. 철자가 정확한지 확인한 다음 다시 시도하세요.

HTTP 상태 코드: 400

ServiceFailureException

알 수 없는 상태로 인해 서비스 장애가 발생했습니다.

HTTP 상태 코드: 500

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++

- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

GetLexicon

AWS 리전에 저장되고 지정된 발음 어휘의 내용을 반환합니다. 자세한 내용은 어휘 관리를 참조하세요.

Request Syntax

```
GET /v1/lexicons/LexiconName HTTP/1.1
```

URI 요청 파라미터

요청은 다음 URI 파라미터를 사용합니다.

LexiconName

어휘의 이름입니다.

패턴: `[0-9A-Za-z]{1,20}`

필수 사항 여부: Yes

Request Body

해당 요청에는 본문이 없습니다.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "Lexicon": {
      "Content": "string",
      "Name": "string"
   },
   "LexiconAttributes": {
      "Alphabet": "string",
      "LanguageCode": "string",
      "LastModified": number,
      "LexemesCount": number,
      "LexiconArn": "string",
      "Size": number
```

```
    }
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

Lexicon

어휘의 이름과 문자열 내용을 제공하는 어휘 객체입니다.

유형: Lexicon객체

LexiconAttributes

사용된 음성 알파벳, 언어 코드, 어휘 ARN, 어휘에 정의된 어휘소 수, 바이트 단위의 어휘 크기 등 어휘의 메타데이터입니다.

유형: LexiconAttributes객체

Errors

LexiconNotFoundException

Amazon Polly에서 지정한 어휘를 찾을 수 없습니다. 이는 어휘가 없거나, 이름의 철자가 틀렸거나, 다른 리전에 있는 어휘를 지정했기 때문일 수 있습니다.

어휘가 존재하고, 해당 리전(ListLexicons 참조)에 있고, 입력한 이름의 철자가 올바른지 확인하세요. 그런 다음 다시 시도하세요.

HTTP 상태 코드: 404

ServiceFailureException

알 수 없는 상태로 인해 서비스 장애가 발생했습니다.

HTTP 상태 코드: 500

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

GetSpeechSynthesisTask

TaskID를 기반으로 특정 SpeechSynthesisTask 객체를 검색합니다. 이 객체에는 작업 상태, 작업 출력물의 S3 버킷의 링크 등 주어진 스피치 합성 작업에 대한 정보가 포함되어 있습니다.

Request Syntax

```
GET /v1/synthesisTasks/TaskId HTTP/1.1
```

URI 요청 파라미터

요청은 다음 URI 파라미터를 사용합니다.

TaskId

Amazon Polly에서 생성한 스피치 합성 작업의 고유 식별자입니다.

패턴: `^[a-zA-Z0-9_-]{1,100}$`

필수 사항 여부: Yes

Request Body

해당 요청에는 본문이 없습니다.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "SynthesisTask": {
      "CreationTime": number,
      "Engine": "string",
      "LanguageCode": "string",
      "LexiconNames": [ "string" ],
      "OutputFormat": "string",
      "OutputUri": "string",
      "RequestCharacters": number,
      "SampleRate": "string",
```

```
      "SnsTopicArn": "string",
      "SpeechMarkTypes": [ "string" ],
      "TaskId": "string",
      "TaskStatus": "string",
      "TaskStatusReason": "string",
      "TextType": "string",
      "VoiceId": "string"
   }
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

SynthesisTask

SynthesisTask 출력 형식, 생성 시간, 작업 상태 등을 포함하여 요청된 작업의 정보를 제공하는 객체입니다.

유형: SynthesisTask객체

Errors

InvalidTaskIdException

제공한 작업 ID가 유효하지 않습니다. 유효한 작업 ID를 제공하고 다시 시도하세요.

HTTP 상태 코드: 400

ServiceFailureException

알 수 없는 상태로 인해 서비스 장애가 발생했습니다.

HTTP 상태 코드: 500

SynthesisTaskNotFoundException

요청된 작업 ID가 있는 스피치 합성 작업을 찾을 수 없습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

ListLexicons

AWS 리전에 저장된 발음 어휘의 목록을 반환합니다. 자세한 내용은 어휘 관리를 참조하세요.

Request Syntax

```
GET /v1/lexicons?NextToken=NextToken HTTP/1.1
```

URI 요청 파라미터

요청은 다음 URI 파라미터를 사용합니다.

NextToken

이전 ListLexicons 작업에서 반환한 불투명한 페이지 매김 토큰입니다. 토큰이 존재하는 경우 어휘 목록을 계속할 위치를 나타냅니다.

길이 제약: 최소 길이는 0입니다. 최대 길이는 4096자입니다.

Request Body

해당 요청에는 본문이 없습니다.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "Lexicons": [
      {
         "Attributes": {
            "Alphabet": "string",
            "LanguageCode": "string",
            "LastModified": number,
            "LexemesCount": number,
            "LexiconArn": "string",
            "Size": number
         },
         "Name": "string"
      }
```

```
    ],
    "NextToken": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

Lexicons

어휘 이름 및 속성 목록입니다.

유형: LexiconDescription객체 어레이

NextToken

다음 요청에서 어휘 목록을 계속 나열하는 데 사용할 페이지 매김 토큰입니다. 응답이 생략되는 경우에만 NextToken을 반환합니다.

타입: 문자열

길이 제약: 최소 길이는 0입니다. 최대 길이는 4096자입니다.

Errors

InvalidNextTokenException

이 (NextToken 가) 유효하지 않습니다. 철자가 정확한지 확인한 다음 다시 시도하세요.

HTTP 상태 코드: 400

ServiceFailureException

알 수 없는 상태로 인해 서비스 장애가 발생했습니다.

HTTP 상태 코드: 500

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

ListSpeechSynthesisTasks

생성 날짜별로 정렬된 SpeechSynthesisTask 객체 목록을 반환합니다. 이 작업을 통해 상태를 기준으로 작업을 필터링할 수 있습니다. 예를 들어 완료된 작업만 표시되도록 할 수 있습니다.

Request Syntax

```
GET /v1/synthesisTasks?MaxResults=MaxResults&NextToken=NextToken&Status=Status HTTP/1.1
```

URI 요청 파라미터

요청은 다음 URI 파라미터를 사용합니다.

MaxResults

목록 작업에서 반환되는 스피치 합성 작업의 최대 수입니다.

유효 범위: 최소값은 1입니다. 최댓값은 100입니다.

NextToken

스피치 합성 작업 목록을 계속 나열하기 위해 다음 요청에서 사용할 페이지 매김 토큰입니다.

길이 제약: 최소 길이는 0입니다. 최대 길이는 4096자입니다.

Status

목록 작업에서 반환된 스피치 합성 작업의 상태입니다.

유효 값: `scheduled | inProgress | completed | failed`

Request Body

해당 요청에는 본문이 없습니다.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
```

```
   "NextToken": "string",
   "SynthesisTasks": [
      {
         "CreationTime": number,
         "Engine": "string",
         "LanguageCode": "string",
         "LexiconNames": [ "string" ],
         "OutputFormat": "string",
         "OutputUri": "string",
         "RequestCharacters": number,
         "SampleRate": "string",
         "SnsTopicArn": "string",
         "SpeechMarkTypes": [ "string" ],
         "TaskId": "string",
         "TaskStatus": "string",
         "TaskStatusReason": "string",
         "TextType": "string",
         "VoiceId": "string"
      }
   ]
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

NextToken

이 요청의 이전 목록 작업에서 반환된 불투명한 페이지 매김 토큰입니다. 존재하는 경우 이는 목록을 계속해야 할 위치를 나타냅니다.

타입: 문자열

길이 제약: 최소 길이는 0입니다. 최대 길이는 4096자입니다.

SynthesisTasks

출력 형식, 생성 시간, 작업 상태 등을 포함하여 목록 요청에서 지정된 작업의 정보를 제공하는 SynthesisTask 객체 목록입니다.

타입: SynthesisTask 객체 배열

Errors

InvalidNextTokenException

이 (NextToken 가) 유효하지 않습니다. 철자가 정확한지 확인한 다음 다시 시도하세요.

HTTP 상태 코드: 400

ServiceFailureException

알 수 없는 상태로 인해 서비스 장애가 발생했습니다.

HTTP 상태 코드: 500

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

PutLexicon

발음 어휘를 AWS 리전리전에 저장합니다. 같은 이름의 어휘가 해당 리전에 이미 있는 경우 새 어휘가 해당 어휘를 덮어씁니다. 사전 작업은 최종 일관성을 유지하므로 작업에 어휘를 사용할 수 있을 때까지 시간이 걸릴 수 있습니다. SynthesizeSpeech

자세한 내용은 어휘 관리를 참조하세요.

Request Syntax

```
PUT /v1/lexicons/LexiconName HTTP/1.1
Content-type: application/json

{
   "Content": "string"
}
```

URI 요청 파라미터

요청은 다음 URI 파라미터를 사용합니다.

LexiconName

어휘의 이름입니다. 이름은 일반 표현 형식 [0-9A-Za-z]{1,20}을 따라야 합니다. 즉, 이름은 최대 20자 길이의 대소문자를 구분하는 영숫자 문자열입니다.

패턴: `[0-9A-Za-z]{1,20}`

필수 사항 여부: Yes

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

Content

PLS 어휘의 내용을 문자열 데이터로 표현한 내용입니다.

타입: 문자열

필수 여부: 예

응답 구문

```
HTTP/1.1 200
```

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반송합니다.

Errors

InvalidLexiconException

Amazon Polly에서 지정한 어휘를 찾을 수 없습니다. 어휘 이름의 철자가 정확한지 확인한 다음 다시 시도하세요.

HTTP 상태 코드: 400

LexiconSizeExceededException

이 작업을 수행하면 지정된 어휘의 최대 크기를 초과하게 됩니다.

HTTP 상태 코드: 400

MaxLexemeLengthExceededException

이 작업을 수행하면 어휘소의 최대 크기를 초과할 수 있습니다.

HTTP 상태 코드: 400

MaxLexiconsNumberExceededException

이 작업을 수행하면 최대 어휘 수를 초과할 수 있습니다.

HTTP 상태 코드: 400

ServiceFailureException

알 수 없는 상태로 인해 서비스 장애가 발생했습니다.

HTTP 상태 코드: 500

UnsupportedPlsAlphabetException

어휘에서 지정한 알파벳은 지원되는 알파벳이 아닙니다. 유효 값은 `x-sampa`및 `ipa`입니다.

HTTP 상태 코드: 400

UnsupportedPlsLanguageException

어휘에 지정된 언어를 지원하지 않습니다. 지원 언어의 목록은 어휘 속성을 참조하세요.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

StartSpeechSynthesisTask

새로운 SpeechSynthesisTask를 시작하여 비동기 합성 작업의 생성을 허용합니다. 이 작업을 수행하려면 스피치 합성에 필요한 모든 표준 정보와 함께 서비스가 합성 작업의 출력을 저장하는 Amazon S3 버킷의 이름과 두 개의 옵션 파라미터(OutputS3KeyPrefix 및 SnsTopicArn)가 필요합니다. 합성 작업이 생성되면 이 작업은 이 작업의 식별자와 현재 상태를 포함하는 SpeechSynthesisTask 객체를 반환합니다. 비동기 합성 작업을 시작한 후 72시간 동안 SpeechSynthesisTask 객체를 사용할 수 있습니다.

Request Syntax

```
POST /v1/synthesisTasks HTTP/1.1
Content-type: application/json

{
   "Engine": "string",
   "LanguageCode": "string",
   "LexiconNames": [ "string" ],
   "OutputFormat": "string",
   "OutputS3BucketName": "string",
   "OutputS3KeyPrefix": "string",
   "SampleRate": "string",
   "SnsTopicArn": "string",
   "SpeechMarkTypes": [ "string" ],
   "Text": "string",
   "TextType": "string",
   "VoiceId": "string"
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

Engine

Amazon Polly가 음성 합성을 위한 입력 텍스트를 처리할 때 사용할 엔진 (standardneural, long-form 또는generative) 을 지정합니다. 선택한 엔진에서 지원되지 않는 음성을 사용하면 오류가 발생합니다.

타입: 문자열

유효 값: `standard | neural | long-form | generative`

필수 여부: 아니요

LanguageCode

스피치 합성 요청을 위한 옵션 언어 코드입니다. 이는 인도 영어(en-IN) 또는 힌디어(hi-IN)에 사용할 수 있는 Aditi와 같은 이중 언어 음성을 사용하는 경우에만 필요합니다.

이중 언어 음성을 사용하지만 언어 코드를 지정하지 않은 경우 Amazon Polly는 이중 언어 음성의 기본 언어를 사용합니다. 모든 음성의 기본 언어는 LanguageCode 파라미터 DescribeVoices연산에서 반환되는 언어입니다. 예를 들어 언어 코드를 지정하지 않은 경우 Aditi는 힌디어 대신 인도 영어를 사용합니다.

타입: 문자열

유효 값: `arb | cmn-CN | cy-GB | da-DK | de-DE | en-AU | en-GB | en-GB-WLS | en-IN | en-US | es-ES | es-MX | es-US | fr-CA | fr-FR | is-IS | it-IT | ja-JP | hi-IN | ko-KR | nb-NO | nl-NL | pl-PL | pt-BR | pt-PT | ro-RO | ru-RU | sv-SE | tr-TR | en-NZ | en-ZA | ca-ES | de-AT | yue-CN | ar-AE | fi-FI | en-IE | nl-BE | fr-BE`

필수 여부: 아니요

LexiconNames

합성 과정에서 서비스에 적용하려는 하나 이상의 발음 어휘 이름의 목록입니다. 어휘는 어휘의 언어가 음성의 언어와 동일한 경우에만 적용됩니다.

유형: 문자열 배열

배열 멤버: 최대 항목 수는 5개입니다.

패턴: `[0-9A-Za-z]{1,20}`

Required: No

OutputFormat

반환된 출력이 인코딩되는 형식입니다. 오디오 스트림의 경우 mp3, ogg_vorbis 또는 pcm이 됩니다. 스피치 마크의 경우 json이 됩니다.

타입: 문자열

유효 값: `json | mp3 | ogg_vorbis | pcm`

필수 사항 여부: 예

OutputS3BucketName

출력 파일이 저장될 Amazon S3 버킷 이름입니다.

유형: String

패턴: `^[a-z0-9][\.\-a-z0-9]{1,61}[a-z0-9]$`

필수 사항 여부: Yes

OutputS3KeyPrefix

출력 스피치 파일의 Amazon S3 키 접두사입니다.

유형: String

패턴: `^[0-9a-zA-Z\/\!\-_\.\*\'\(\):;\$@=+\,\?&]{0,800}$`

Required: No

SampleRate

Hz로 지정된 오디오 주파수입니다.

mp3 및 ogg_vorbis의 유효한 값은 “8000”, “16000”, “22050”, “24000”입니다. 표준 음성의 기본값은 “22050”이고 신경 음성의 기본값은 “24000”입니다. 롱폼 음성의 기본값은 ‘24000’입니다. 제너레이티브 보이스의 기본값은 “24000"입니다.

pcm의 유효한 값은 “8000”과 “16000”입니다. 기본값은 “16000”입니다.

타입: 문자열

필수사항: 아니요

SnsTopicArn

스피치 합성 작업에 대한 상태 알림을 제공할 때 옵션으로 사용되는 SNS 주제용 ARN입니다.

유형: String

패턴: `^arn:aws(-(cn|iso(-b)?|us-gov))?:sns:[a-z0-9_-]{1,50}:\d{12}:[a-zA-Z0-9_-]{1,251}([a-zA-Z0-9_-]{0,5}|\.fifo)$`

Required: No

SpeechMarkTypes

입력 텍스트에 대해 반환되는 스피치 마크의 유형입니다.

유형: 문자열 어레이

배열 멤버: 최대 항목 수는 4개입니다.

유효 값: `sentence | ssml | viseme | word`

필수 여부: 아니요

Text

합성할 입력 텍스트입니다. ssml을 로 지정하는 경우 입력 텍스트의 TextType SSML 형식을 따르십시오.

타입: 문자열

필수 항목 여부: 예

TextType

입력 텍스트가 일반 텍스트인지 SSML인지 지정합니다. 기본값은 일반 텍스트입니다.

타입: 문자열

유효 값: `ssml | text`

필수 여부: 아니요

VoiceId

합성에 사용할 음성 ID입니다.

타입: 문자열

유효 값: `Aditi | Amy | Astrid | Bianca | Brian | Camila | Carla | Carmen | Celine | Chantal | Conchita | Cristiano | Dora | Emma | Enrique | Ewa | Filiz | Gabrielle | Geraint | Giorgio | Gwyneth | Hans | Ines | Ivy | Jacek | Jan | Joanna | Joey | Justin | Karl | Kendra | Kevin | Kimberly`

| Lea | Liv | Lotte | Lucia | Lupe | Mads | Maja | Marlene | Mathieu | Matthew | Maxim | Mia | Miguel | Mizuki | Naja | Nicole | Olivia | Penelope | Raveena | Ricardo | Ruben | Russell | Salli | Seoyeon | Takumi | Tatyana | Vicki | Vitoria | Zeina | Zhiyu | Aria | Ayanda | Arlet | Hannah | Arthur | Daniel | Liam | Pedro | Kajal | Hiujin | Laura | Elin | Ida | Suvi | Ola | Hala | Andres | Sergio | Remi | Adriano | Thiago | Ruth | Stephen | Kazuha | Tomoko | Niamh | Sofie | Lisa | Isabelle | Zayd | Danielle | Gregory | Burcu

필수 여부: 예

응답 구문

```
HTTP/1.1 200
Content-type: application/json

{
   "SynthesisTask": {
      "CreationTime": number,
      "Engine": "string",
      "LanguageCode": "string",
      "LexiconNames": [ "string" ],
      "OutputFormat": "string",
      "OutputUri": "string",
      "RequestCharacters": number,
      "SampleRate": "string",
      "SnsTopicArn": "string",
      "SpeechMarkTypes": [ "string" ],
      "TaskId": "string",
      "TaskStatus": "string",
      "TaskStatusReason": "string",
      "TextType": "string",
      "VoiceId": "string"
   }
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

SynthesisTask

SynthesisTask 새로 제출된 음성 합성 작업에 대한 정보와 속성을 제공하는 객체입니다.

유형: SynthesisTask객체

Errors

EngineNotSupportedException

이 엔진은 지정한 음성과 호환되지 않습니다. 엔진과 호환되는 새 음성을 선택하거나 엔진을 교체하고 작업을 다시 시작하세요.

HTTP 상태 코드: 400

InvalidS3BucketException

제공된 Amazon S3 버킷 이름이 유효하지 않습니다. S3 버킷의 명명 규칙을 확인하고 다시 시도하세요.

HTTP 상태 코드: 400

InvalidS3KeyException

제공된 Amazon S3 키 접두사가 유효하지 않습니다. 유효한 S3 객체 키 이름을 제공하세요.

HTTP 상태 코드: 400

InvalidSampleRateException

지정한 샘플 속도가 유효하지 않습니다.

HTTP 상태 코드: 400

InvalidSnsTopicArnException

제공된 SNS 주제 ARN이 유효하지 않습니다. 유효한 SNS 주제 ARN을 제공하고 다시 시도하세요.

HTTP 상태 코드: 400

InvalidSsmlException

제공한 SSML이 유효하지 않습니다. SSML 구문, 태그의 철자 및 값을 확인한 다음 다시 시도하세요.

HTTP 상태 코드: 400

LanguageNotSupportedException

Amazon Polly의 현재 용량에서는 지정된 언어를 지원하지 않습니다.

HTTP 상태 코드: 400

LexiconNotFoundException

Amazon Polly에서 지정한 어휘를 찾을 수 없습니다. 이는 어휘가 없거나, 이름의 철자가 틀렸거나, 다른 리전에 있는 어휘를 지정했기 때문일 수 있습니다.

어휘가 존재하고, 해당 리전(ListLexicons 참조)에 있고, 입력한 이름의 철자가 올바른지 확인하세요. 그런 다음 다시 시도하세요.

HTTP 상태 코드: 404

MarksNotSupportedForFormatException

선택한 `OutputFormat`에서 스피치 마크를 지원하지 않습니다. 스피치 마크는 `json` 형식의 내용에서만 사용할 수 있습니다.

HTTP 상태 코드: 400

ServiceFailureException

알 수 없는 상태로 인해 서비스 장애가 발생했습니다.

HTTP 상태 코드: 500

SsmlMarksNotSupportedForTextTypeException

일반 텍스트 유형 입력에는 SSML 스피치 마크가 지원되지 않습니다.

HTTP 상태 코드: 400

TextLengthExceededException

"Text" 파라미터 값이 허용된 한도보다 깁니다. `SynthesizeSpeech` API의 경우 입력 텍스트 한도는 최대 총 6,000자이며, 이 중 청구 가능한 문자 수는 3,000자를 초과할 수 없습니다. `StartSpeechSynthesisTask` API의 경우 최대 200,000자이며, 이 중 청구 가능한 문자 수는 100,000자를 초과할 수 없습니다. SSML 태그는 청구 문자 수로 계산되지 않습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

SynthesizeSpeech

UTF-8 입력, 일반 텍스트 또는 SSML을 바이트 스트림으로 합성합니다. SSML 입력은 유효하고 올바른 형식의 SSML이어야 합니다. 음소 매핑을 사용하지 않는 한 일부 음성에서 일부 알파벳을 사용할 수 없습니다(예: 키릴 문자는 영어 음성으로 전혀 읽히지 않을 수 있음). 자세한 내용은 작동 방식을 참조하세요.

Request Syntax

```
POST /v1/speech HTTP/1.1
Content-type: application/json

{
   "Engine": "string",
   "LanguageCode": "string",
   "LexiconNames": [ "string" ],
   "OutputFormat": "string",
   "SampleRate": "string",
   "SpeechMarkTypes": [ "string" ],
   "Text": "string",
   "TextType": "string",
   "VoiceId": "string"
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

Engine

Amazon Polly가 음성 합성을 위한 입력 텍스트를 처리할 때 사용할 엔진 (`standard`,`neural`, `long-form`, 또는`generative`) 을 지정합니다. 선택한 음성이 지원하는 엔진을 제공하십시오. 엔진을 제공하지 않는 경우 기본적으로 표준 엔진이 선택됩니다. 선택한 음성이 표준 엔진에서 지원되지 않는 경우 오류가 발생합니다. Amazon Polly 음성 및 각 엔진에서 사용할 수 있는 음성에 대한 자세한 내용은 사용 가능한 음성을 참조하세요.

타입: 문자열

유효한 값: `standard | neural | long-form | generative`

필수 항목 여부: 예

타입: 문자열

유효 값: `standard | neural | long-form | generative`

필수 여부: 아니요

LanguageCode

스피치 합성 요청을 위한 옵션 언어 코드입니다. 이는 인도 영어(en-IN) 또는 힌디어(hi-IN)에 사용할 수 있는 Aditi와 같은 이중 언어 음성을 사용하는 경우에만 필요합니다.

이중 언어 음성을 사용하지만 언어 코드를 지정하지 않은 경우 Amazon Polly는 이중 언어 음성의 기본 언어를 사용합니다. 모든 음성의 기본 언어는 LanguageCode 파라미터 DescribeVoices조작에서 반환되는 언어입니다. 예를 들어 언어 코드를 지정하지 않은 경우 Aditi는 힌디어 대신 인도 영어를 사용합니다.

타입: 문자열

유효 값: `arb | cmn-CN | cy-GB | da-DK | de-DE | en-AU | en-GB | en-GB-WLS | en-IN | en-US | es-ES | es-MX | es-US | fr-CA | fr-FR | is-IS | it-IT | ja-JP | hi-IN | ko-KR | nb-NO | nl-NL | pl-PL | pt-BR | pt-PT | ro-RO | ru-RU | sv-SE | tr-TR | en-NZ | en-ZA | ca-ES | de-AT | yue-CN | ar-AE | fi-FI | en-IE | nl-BE | fr-BE`

필수 여부: 아니요

LexiconNames

합성 과정에서 서비스에 적용하려는 하나 이상의 발음 어휘 이름의 목록입니다. 어휘는 어휘의 언어가 음성의 언어와 동일한 경우에만 적용됩니다. 어휘집 저장에 대한 자세한 내용은 을 참조하십시오 PutLexicon.

유형: 문자열 배열

배열 멤버: 최대 항목 수는 5개입니다.

패턴: `[0-9A-Za-z]{1,20}`

Required: No

OutputFormat

반환된 출력이 인코딩되는 형식입니다. 오디오 스트림의 경우 mp3, ogg_vorbis 또는 pcm이 됩니다. 스피치 마크의 경우 json이 됩니다.

pcm을 사용하면 반환되는 콘텐츠는 부호가 있는 16비트, 1채널(모노), 리틀 엔디안 형식의 오디오/pcm입니다.

타입: 문자열

유효 값: `json | mp3 | ogg_vorbis | pcm`

필수 사항 여부: 예

SampleRate

Hz로 지정된 오디오 주파수입니다.

mp3 및 ogg_vorbis의 유효한 값은 “8000”, “16000”, “22050”, “24000”입니다. 표준 음성의 기본값은 “22050”이고 신경 음성의 기본값은 “24000”입니다. 롱폼 음성의 기본값은 ‘24000’입니다. 제너레이티브 음성의 기본값은 “24000"입니다.

pcm의 유효한 값은 “8000”과 “16000”입니다. 기본값은 “16000”입니다.

타입: 문자열

필수사항: 아니요

SpeechMarkTypes

입력 텍스트에 대해 반환되는 스피치 마크의 유형입니다.

유형: 문자열 어레이

배열 멤버: 최대 항목 수는 4개입니다.

유효 값: `sentence | ssml | viseme | word`

필수 여부: 아니요

Text

합성할 입력 텍스트입니다. `ssml`을 `TextType`으로 지정하는 경우 입력 텍스트의 SSML 형식을 따라야 합니다.

타입: 문자열

필수 항목 여부: 예

TextType

입력 텍스트가 일반 텍스트인지 SSML인지 지정합니다. 기본값은 일반 텍스트입니다. 자세한 내용은 SSML 사용을 참조하세요.

타입: 문자열

유효 값: `ssml | text`

필수 여부: 아니요

VoiceId

합성에 사용할 음성 ID입니다. 오퍼레이션을 호출하여 사용 가능한 음성 ID 목록을 가져올 수 있습니다. DescribeVoices

타입: 문자열

유효 값: `Aditi | Amy | Astrid | Bianca | Brian | Camila | Carla | Carmen | Celine | Chantal | Conchita | Cristiano | Dora | Emma | Enrique | Ewa | Filiz | Gabrielle | Geraint | Giorgio | Gwyneth | Hans | Ines | Ivy | Jacek | Jan | Joanna | Joey | Justin | Karl | Kendra | Kevin | Kimberly | Lea | Liv | Lotte | Lucia | Lupe | Mads | Maja | Marlene | Mathieu | Matthew | Maxim | Mia | Miguel | Mizuki | Naja | Nicole | Olivia | Penelope | Raveena | Ricardo | Ruben | Russell | Salli | Seoyeon | Takumi | Tatyana | Vicki | Vitoria | Zeina | Zhiyu | Aria | Ayanda | Arlet | Hannah | Arthur | Daniel | Liam | Pedro | Kajal | Hiujin | Laura | Elin | Ida | Suvi | Ola | Hala | Andres | Sergio | Remi | Adriano | Thiago | Ruth | Stephen | Kazuha | Tomoko | Niamh | Sofie | Lisa | Isabelle | Zayd | Danielle | Gregory | Burcu`

필수 여부: 예

응답 구문

```
HTTP/1.1 200
Content-Type: ContentType
x-amzn-RequestCharacters: RequestCharacters
```

```
AudioStream
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

응답에 다음 HTTP 헤더가 반환됩니다.

ContentType

오디오 스트림 유형을 지정합니다. 여기에는 요청의 OutputFormat 파라미터가 반영되어야 합니다.

- mp3를 OutputFormat으로 요청하면 ContentType이 audio/mpeg를 반환합니다.
- ogg_vorbis를 OutputFormat으로 요청하면 ContentType이 audio/ogg를 반환합니다.
- pcm을 OutputFormat으로 요청하면 ContentType이 부호가 있는 16비트, 1채널(모노), 리틀 엔디안 형식의 audio/pcm을 반환합니다.
- json로 요청하면 x-json-stream application/이 ContentType 반환됩니다. OutputFormat

RequestCharacters

합성된 문자 수입니다.

응답은 다음 내용을 HTTP 본문으로 반환합니다.

AudioStream

합성된 스피치가 포함된 스트림입니다.

Errors

EngineNotSupportedException

이 엔진은 지정한 음성과 호환되지 않습니다. 엔진과 호환되는 새 음성을 선택하거나 엔진을 교체하고 작업을 다시 시작하세요.

HTTP 상태 코드: 400

InvalidSampleRateException

지정한 샘플 속도가 유효하지 않습니다.

HTTP 상태 코드: 400

InvalidSsmlException

제공한 SSML이 유효하지 않습니다. SSML 구문, 태그의 철자 및 값을 확인한 다음 다시 시도하세요.

HTTP 상태 코드: 400

LanguageNotSupportedException

Amazon Polly의 현재 용량에서는 지정된 언어를 지원하지 않습니다.

HTTP 상태 코드: 400

LexiconNotFoundException

Amazon Polly에서 지정한 어휘를 찾을 수 없습니다. 이는 어휘가 없거나, 이름의 철자가 틀렸거나, 다른 리전에 있는 어휘를 지정했기 때문일 수 있습니다.

어휘가 존재하고, 해당 리전(ListLexicons 참조)에 있고, 입력한 이름의 철자가 올바른지 확인하세요. 그런 다음 다시 시도하세요.

HTTP 상태 코드: 404

MarksNotSupportedForFormatException

선택한 `OutputFormat`에서 스피치 마크를 지원하지 않습니다. 스피치 마크는 `json` 형식의 내용에서만 사용할 수 있습니다.

HTTP 상태 코드: 400

ServiceFailureException

알 수 없는 상태로 인해 서비스 장애가 발생했습니다.

HTTP 상태 코드: 500

SsmlMarksNotSupportedForTextTypeException

일반 텍스트 유형 입력에는 SSML 스피치 마크가 지원되지 않습니다.

HTTP 상태 코드: 400

TextLengthExceededException

“Text” 파라미터 값이 허용된 한도보다 깁니다. `SynthesizeSpeech` API의 경우 입력 텍스트 한도는 최대 총 6,000자이며, 이 중 청구 가능한 문자 수는 3,000자를 초과할 수 없습니다.

`StartSpeechSynthesisTask` API의 경우 최대 200,000자이며, 이 중 청구 가능한 문자 수는 100,000자를 초과할 수 없습니다. SSML 태그는 청구 문자 수로 계산되지 않습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

데이터 유형

다음 데이터 유형이 지원됩니다.

- Lexicon
- LexiconAttributes
- LexiconDescription
- SynthesisTask
- Voice

Lexicon

어휘 이름과 문자열 형식의 어휘 내용을 제공합니다. 자세한 내용은 발음 어휘 사양(PLS) 버전 1.0을 참조하세요.

내용

Content

문자열 형식의 어휘 내용입니다. 어휘의 내용은 PLS 형식이어야 합니다.

타입: 문자열

필수사항: 아니요

Name

어휘의 이름입니다.

유형: String

패턴: `[0-9A-Za-z]{1,20}`

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

LexiconAttributes

어휘소의 수, 언어 코드 등과 같이 어휘를 설명하는 메타데이터를 포함합니다. 자세한 내용은 어휘 관리를 참조하세요.

내용

Alphabet

어휘에서 사용하는 발음기호입니다. 유효 값은 `ipa`및 `x-sampa`입니다.

타입: 문자열

필수사항: 아니요

LanguageCode

어휘가 적용되는 언어 코드입니다. 언어 코드가 “en”과 같은 어휘는 모든 영어(en-GB, en-US, en-AUS, en-AUS, en-WLS 등)에 적용됩니다.

타입: 문자열

유효 값: `arb | cmn-CN | cy-GB | da-DK | de-DE | en-AU | en-GB | en-GB-WLS | en-IN | en-US | es-ES | es-MX | es-US | fr-CA | fr-FR | is-IS | it-IT | ja-JP | hi-IN | ko-KR | nb-NO | nl-NL | pl-PL | pt-BR | pt-PT | ro-RO | ru-RU | sv-SE | tr-TR | en-NZ | en-ZA | ca-ES | de-AT | yue-CN | ar-AE | fi-FI | en-IE | nl-BE | fr-BE`

필수 여부: 아니요

LastModified

어휘가 마지막으로 수정된 날짜(타임스탬프 값)입니다.

유형: 타임스탬프

필수 여부: 아니요

LexemesCount

어휘에 포함된 어휘소의 수입니다.

유형: 정수

필수 항목 여부: 아니요

LexiconArn

어휘의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

필수사항: 아니요

Size

어휘의 전체 크기(문자 수)입니다.

유형: 정수

필수 항목 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

LexiconDescription

어휘의 내용을 설명합니다.

내용

Attributes

어휘 메타데이터를 제공합니다.

유형: LexiconAttributes객체

필수 항목 여부: 아니요

Name

어휘의 이름입니다.

유형: String

패턴: `[0-9A-Za-z]{1,20}`

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

SynthesisTask

SynthesisTask 음성 합성 작업에 대한 정보를 제공하는 객체입니다.

내용

CreationTime

합성 작업이 시작된 시각의 타임스탬프입니다.

유형: 타임스탬프

필수 여부: 아니요

Engine

Amazon Polly가 음성 합성을 위한 입력 텍스트를 처리할 때 사용할 엔진 (`standard`, `neural`, `long-form` 또는`generative`) 을 지정합니다. 선택한 엔진에서 지원되지 않는 음성을 사용하면 오류가 발생합니다.

타입: 문자열

유효 값: `standard | neural | long-form | generative`

필수 여부: 아니요

LanguageCode

합성 작업을 위한 옵션 언어 코드입니다. 이는 인도 영어(en-IN) 또는 힌디어(hi-IN)에 사용할 수 있는 Aditi와 같은 이중 언어 음성을 사용하는 경우에만 필요합니다.

이중 언어 음성을 사용하지만 언어 코드를 지정하지 않은 경우 Amazon Polly는 이중 언어 음성의 기본 언어를 사용합니다. 모든 음성의 기본 언어는 LanguageCode 파라미터 DescribeVoices연산에서 반환되는 언어입니다. 예를 들어 언어 코드를 지정하지 않은 경우 Aditi는 힌디어 대신 인도 영어를 사용합니다.

타입: 문자열

유효 값: `arb | cmn-CN | cy-GB | da-DK | de-DE | en-AU | en-GB | en-GB-WLS | en-IN | en-US | es-ES | es-MX | es-US | fr-CA | fr-FR | is-IS | it-IT | ja-JP | hi-IN | ko-KR | nb-NO | nl-NL | pl-PL | pt-BR | pt-PT | ro-RO | ru-RU | sv-SE | tr-TR | en-NZ | en-ZA | ca-ES | de-AT | yue-CN | ar-AE | fi-FI | en-IE | nl-BE | fr-BE`

필수 여부: 아니요

LexiconNames

합성 과정에서 서비스에 적용하려는 하나 이상의 발음 어휘 이름의 목록입니다. 어휘는 어휘의 언어가 음성의 언어와 동일한 경우에만 적용됩니다.

유형: 문자열 배열

배열 멤버: 최대 항목 수는 5개입니다.

패턴: `[0-9A-Za-z]{1,20}`

Required: No

OutputFormat

반환된 출력이 인코딩되는 형식입니다. 오디오 스트림의 경우 mp3, ogg_vorbis 또는 pcm이 됩니다. 스피치 마크의 경우 json이 됩니다.

타입: 문자열

유효 값: `json | mp3 | ogg_vorbis | pcm`

필수 여부: 아니요

OutputUri

출력 스피치 파일의 경로입니다.

타입: 문자열

필수사항: 아니요

RequestCharacters

합성된 청구 가능 문자 수입니다.

유형: 정수

필수 항목 여부: 아니요

SampleRate

Hz로 지정된 오디오 주파수입니다.

mp3 및 ogg_vorbis의 유효한 값은 “8000”, “16000”, “22050”, “24000”입니다. 표준 음성의 기본값은 “22050”이고 신경 음성의 기본값은 “24000”입니다. 롱폼 음성의 기본값은 ‘24000’입니다. 제너레이티브 보이스의 기본값은 “24000"입니다.

pcm의 유효한 값은 “8000”과 “16000”입니다. 기본값은 “16000”입니다.

타입: 문자열

필수사항: 아니요

SnsTopicArn

스피치 합성 작업에 대한 상태 알림을 제공할 때 옵션으로 사용되는 SNS 주제용 ARN입니다.

유형: String

패턴: `^arn:aws(-(cn|iso(-b)?|us-gov))?:sns:[a-z0-9_-]{1,50}:\d{12}:[a-zA-Z0-9_-]{1,251}([a-zA-Z0-9_-]{0,5}|\.fifo)$`

Required: No

SpeechMarkTypes

입력 텍스트에 대해 반환되는 스피치 마크의 유형입니다.

유형: 문자열 어레이

배열 멤버: 최대 항목 수는 4개입니다.

유효 값: `sentence | ssml | viseme | word`

필수 여부: 아니요

TaskId

Amazon Polly에서 생성한 스피치 합성 작업의 고유 식별자입니다.

유형: String

패턴: `^[a-zA-Z0-9_-]{1,100}$`

Required: No

TaskStatus

개별 스피치 합성 작업의 현재 상태입니다.

타입: 문자열

유효 값: `scheduled | inProgress | completed | failed`

필수 여부: 아니요

TaskStatusReason

특정 스피치 합성 작업의 현재 상태에 대한 이유(작업 실패 시 오류 포함)입니다.

타입: 문자열

필수사항: 아니요

TextType

입력 텍스트가 일반 텍스트인지 SSML인지 지정합니다. 기본값은 일반 텍스트입니다.

타입: 문자열

유효 값: `ssml | text`

필수 여부: 아니요

VoiceId

합성에 사용할 음성 ID입니다.

타입: 문자열

유효 값: `Aditi | Amy | Astrid | Bianca | Brian | Camila | Carla | Carmen | Celine | Chantal | Conchita | Cristiano | Dora | Emma | Enrique | Ewa | Filiz | Gabrielle | Geraint | Giorgio | Gwyneth | Hans | Ines | Ivy | Jacek | Jan | Joanna | Joey | Justin | Karl | Kendra | Kevin | Kimberly | Lea | Liv | Lotte | Lucia | Lupe | Mads | Maja | Marlene | Mathieu | Matthew | Maxim | Mia | Miguel | Mizuki | Naja | Nicole | Olivia | Penelope | Raveena | Ricardo | Ruben | Russell | Salli | Seoyeon | Takumi | Tatyana | Vicki | Vitoria | Zeina | Zhiyu | Aria | Ayanda | Arlet | Hannah | Arthur | Daniel | Liam | Pedro | Kajal | Hiujin | Laura | Elin | Ida | Suvi | Ola | Hala | Andres | Sergio | Remi | Adriano | Thiago | Ruth | Stephen | Kazuha | Tomoko | Niamh | Sofie | Lisa | Isabelle | Zayd | Danielle | Gregory | Burcu`

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

Voice

음성에 대한 설명입니다.

내용

AdditionalLanguageCodes

기본 언어 외에 지정된 음성에서 사용할 수 있는 언어에 대한 추가 코드입니다.

예를 들어 Aditi의 기본 언어는 인도 영어(en-IN)인데, 이는 해당 언어에 처음 사용되었기 때문입니다. Aditi는 인도 영어와 힌디어를 모두 유창하게 구사하는 이중 언어를 구사하므로 이 파라미터는 `hi-IN` 코드를 표시합니다.

유형: 문자열 어레이

유효 값: `arb | cmn-CN | cy-GB | da-DK | de-DE | en-AU | en-GB | en-GB-WLS | en-IN | en-US | es-ES | es-MX | es-US | fr-CA | fr-FR | is-IS | it-IT | ja-JP | hi-IN | ko-KR | nb-NO | nl-NL | pl-PL | pt-BR | pt-PT | ro-RO | ru-RU | sv-SE | tr-TR | en-NZ | en-ZA | ca-ES | de-AT | yue-CN | ar-AE | fi-FI | en-IE | nl-BE | fr-BE`

필수 여부: 아니요

Gender

음성의 성별입니다.

타입: 문자열

유효 값: `Female | Male`

필수 여부: 아니요

Id

Amazon Polly에서 할당한 음성 ID입니다. `SynthesizeSpeech` 작업을 호출할 때 지정하는 ID입니다.

타입: 문자열

유효 값: `Aditi | Amy | Astrid | Bianca | Brian | Camila | Carla | Carmen | Celine | Chantal | Conchita | Cristiano | Dora | Emma | Enrique | Ewa`

| Filiz | Gabrielle | Geraint | Giorgio | Gwyneth | Hans | Ines | Ivy | Jacek | Jan | Joanna | Joey | Justin | Karl | Kendra | Kevin | Kimberly | Lea | Liv | Lotte | Lucia | Lupe | Mads | Maja | Marlene | Mathieu | Matthew | Maxim | Mia | Miguel | Mizuki | Naja | Nicole | Olivia | Penelope | Raveena | Ricardo | Ruben | Russell | Salli | Seoyeon | Takumi | Tatyana | Vicki | Vitoria | Zeina | Zhiyu | Aria | Ayanda | Arlet | Hannah | Arthur | Daniel | Liam | Pedro | Kajal | Hiujin | Laura | Elin | Ida | Suvi | Ola | Hala | Andres | Sergio | Remi | Adriano | Thiago | Ruth | Stephen | Kazuha | Tomoko | Niamh | Sofie | Lisa | Isabelle | Zayd | Danielle | Gregory | Burcu

필수 여부: 아니요

LanguageCode

음성의 언어 코드입니다.

타입: 문자열

유효 값: arb | cmn-CN | cy-GB | da-DK | de-DE | en-AU | en-GB | en-GB-WLS | en-IN | en-US | es-ES | es-MX | es-US | fr-CA | fr-FR | is-IS | it-IT | ja-JP | hi-IN | ko-KR | nb-NO | nl-NL | pl-PL | pt-BR | pt-PT | ro-RO | ru-RU | sv-SE | tr-TR | en-NZ | en-ZA | ca-ES | de-AT | yue-CN | ar-AE | fi-FI | en-IE | nl-BE | fr-BE

필수 여부: 아니요

LanguageName

사람이 읽을 수 있는 언어의 영어 이름입니다.

타입: 문자열

필수사항: 아니요

Name

목소리의 이름(예: Salli, Kendra 등)입니다. 이를 사용하여 애플리케이션에 사람이 읽을 수 있는 음성 이름을 표시할 수 있습니다.

타입: 문자열

필수사항: 아니요

SupportedEngines

지정된 음성이 지원하는 엔진 (`standard`, `neural`, `long-form` 또는 `generative`) 을 지정합니다.

유형: 문자열 어레이

유효 값: `standard | neural | long-form | generative`

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

Amazon Polly 문서 기록

다음 표는 Amazon Polly 개발자 안내서의 각 릴리스에서 변경된 중요 사항에 대해 설명합니다. 이 문서에 대한 업데이트 알림을 받으려면 RSS 피드를 구독하면 됩니다.

- 최신 설명서 업데이트: 2024년 3월 28일

변경 사항	설명	날짜
새로운 제너레이티브 음성 엔진 추가	Amazon Polly는 이제 에이미 (Amy), 매튜 (Matthew), 루스 (Ruth) 의 세 가지 영어 음성을 생성형 변형으로 사용하여 더 긴 콘텐츠를 위해 설계된 제너레이티브 음성 엔진을 제공합니다. 자세한 내용은 제너레이티브 보이스를 참조하십시오.	2024년 3월 28일
NTTS에 새 음성 추가	Amazon Polly는 이제 NTTS 터키어 음성 부르쿠를 제공합니다. NTTS 음성 목록은 신경 음성을 참조하세요.	2024년 2월 14일
새로운 롱폼 음성 엔진 추가	Amazon Polly는 이제 세 가지 en-US 음성(Danielle, Gregory, Ruth)을 사용하여 긴 콘텐츠를 위해 설계된 롱폼 음성 엔진을 제공합니다. 자세한 내용은 롱폼 음성을 참조하세요.	2023년 11월 16일
NTTS에 새 음성 추가	이제 Amazon Polly에서 Danielle과 Gregory라는 두 개의 새로운 NTTS 미국 영어 음성을 제공합니다. NTTS 음성 목록은 신경 음성을 참조하세요.	2023년 10월 5일

Windows용 Amazon Polly	Amazon Polly Windows SAPI(Speech Application Programming Interface) 플러그인은 더 이상 지원되지 않습니다.	2023년 9월 26일
Amazon Polly 할당량 가이드 업데이트	Amazon Polly 할당량 안내서가 업데이트되었습니다. 예제 및 용어 설명이 추가되었습니다. 업데이트는 Amazon Polly 할당량을 참조하세요.	2023년 8월 17일
NTTS에 새 음성 추가	이제 Amazon Polly에서 걸프 아랍어 NTTS 음성 Zayd를 제공합니다. NTTS 음성 목록은 신경 음성을 참조하세요.	2023년 8월 16일
NTTS에 새 음성 추가	이제 Amazon Polly에서 벨기에 프랑스어 NTTS 음성 Isabelle을 제공합니다. NTTS 음성 목록은 신경 음성을 참조하세요.	2023년 8월 1일
NTTS에 새 음성 추가	이제 Amazon Polly에서 벨기에 네덜란드어(플랑드르) NTTS 음성 Lisa를 제공합니다. NTTS 음성 목록은 신경 음성을 참조하세요.	2023년 6월 7일
NTTS에 새 음성 추가	이제 Amazon Polly에서 아일랜드 영어(Niamh)와 덴마크어(Sofie)의 두 가지 새로운 NTTS 음성을 제공합니다. NTTS 음성 목록은 신경 음성을 참조하세요.	2023년 5월 30일

Amazon Polly에 대한 IAM 가이드 업데이트	IAM 모범 사례에 따라 안내서가 업데이트되었습니다. 자세한 내용은 IAM의 보안 모범 사례를 참조하십시오.	2023년 4월 19일
WordPress 업데이트	Amazon Polly WordPress 플러그인은 더 이상 지원되지 않습니다.	2023년 4월 6일
새 리전 추가	이제 아시아 태평양(오사카) AWS 리전에서 Amazon Polly를 사용할 수 있습니다. 이 리전은 신경 TTS(NTTS)를 지원합니다. 자세한 내용은 기능 및 리전 호환성의 NTTS를 지원하는 리전 목록을 참조하세요.	2023년 4월 5일
NTTS에 새 음성 추가	이제 Amazon Polly에서 Kazuha와 Tomoko라는 두 개의 새로운 일본어 NTTS 음성을 제공합니다. NTTS 음성 목록은 신경 음성을 참조하세요.	2023년 2월 7일
NTTS에 새 음성 추가	이제 Amazon Polly에서 Stephen과 Ruth라는 두 개의 새로운 미국 영어 NTTS 음성을 제공합니다. NTTS 음성 목록은 신경 음성을 참조하세요.	2023년 1월 31일

NTTS에 새 음성 추가	이제 Amazon Polly에서 다음과 같은 새로운 NTTS 음성을 제공합니다. 이들은 브라질 포르투갈어(Thiago), 카스티야 스페인어(Sergio), 프랑스어(Rémi), 이탈리아어(Adriano), 멕시코 스페인어(Andrés)입니다. NTTS 음성 목록은 신경 음성을 참조하세요.	2023년 1월 24일
NTTS에 새 음성 추가	이제 Amazon Polly에서 아랍어(Hala)와 폴란드어(Ola) NTTS 음성을 제공합니다. NTTS 음성 목록은 신경 음성을 참조하세요.	2022년 11월 17일
릴리스 지원 AWS PrivateLink	Amazon Polly는 이제 지원을 제공합니다 AWS PrivateLink . 자세한 내용은 VPC 엔드포인트와 함께 Amazon Polly 사용을 참조하세요.	2022년 11월 9일
NTTS에 새 음성 및 언어 추가	이제 Amazon Polly에서 핀란드어(Suvi), 노르웨이어(Ida) 및 스웨덴어(Elin) NTTS 음성을 제공합니다. NTTS 음성 목록은 신경 음성을 참조하세요.	2022년 11월 8일
NTTS에 새 음성 추가	이제 Amazon Polly에서 네덜란드어 NTTS 음성 Laura를 제공합니다. NTTS 음성 목록은 신경 음성을 참조하세요.	2022년 11월 2일

새 리전 추가	이제 유럽(파리) AWS 리전에서 Amazon Polly를 사용할 수 있습니다. 이 리전은 신경 TTS(NTTS)를 지원합니다. 자세한 내용은 기능 및 리전 호환성의 NTTS를 지원하는 리전 목록을 참조하세요.	2022년 9월 22일
NTTS에 새 음성 및 언어 추가	이제 Amazon Polly에서 광동어 NTTS 음성 Hiujin을 제공합니다. NTTS 음성 목록은 신경 음성을 참조하세요.	2022년 9월 20일
새 리전 추가	이제 아시아 태평양(뭄바이) AWS 리전에서 Amazon Polly를 사용할 수 있습니다. 이 리전은 신경 TTS(NTTS)를 지원합니다. 자세한 내용은 기능 및 리전 호환성의 NTTS를 지원하는 리전 목록을 참조하세요.	2022년 9월 1일
NTTS에 새 음성 추가	이제 Amazon Polly에서 북경어 음성 Zhiyu를 NTTS 음성으로 제공합니다. NTTS 음성 목록은 신경 음성을 참조하세요.	2022년 8월 23일
NTTS에 새 음성 추가	이제 Amazon Polly에서 힌디어 NTTS 음성 Kajal을 제공합니다. NTTS 음성 목록은 신경 음성을 참조하세요.	2022년 7월 27일

NTTS에 새 음성 추가	이제 Amazon Polly에서 미국 스페인어(Pedro), 독일어(Daniel), 캐나다 프랑스어(Liam) 및 영국 영어(Arthur)에 대한 NTTS 음성을 제공합니다. NTTS 음성 목록은 신경 음성을 참조하세요.	2022년 6월 28일
NTTS에 새 음성 추가	이제 Amazon Polly에서 포르투갈어(브라질) 음성 Vitória를 NTTS 음성으로 제공합니다. NTTS 음성 목록은 신경 음성을 참조하세요.	2022년 4월 27일
NTTS에 새 음성 추가	이제 Amazon Polly에서 포르투갈어(유럽) 음성 Inês를 NTTS 음성으로 제공합니다. NTTS 음성 목록은 신경 음성을 참조하세요.	2022년 4월 26일
NTTS에 새 음성 및 언어 추가	이제 Amazon Polly에서 독일어(오스트리아)와 NTTS 음성 Hannah를 제공합니다. NTTS 음성 목록은 신경 음성을 참조하세요.	2022년 4월 19일
NTTS에 새 음성 및 언어 추가	이제 Amazon Polly에서 스페인어(멕시코) 음성 Mia를 NTTS 음성으로 제공합니다. 새로운 언어인 카탈루냐어가 NTTS 음성 Arlet과 함께 추가되었습니다. NTTS 음성 목록은 신경 음성을 참조하세요.	2022년 3월 22일

NTTS에 새 음성 추가	이제 Amazon Polly에서 일본어 음성 Takumi를 NTTS 음성으로 제공합니다. NTTS 음성 목록은 신경 음성을 참조하세요.	2021년 12월 6일
NTTS에 새 음성 추가	이제 Amazon Polly에서 프랑스어 음성 Léa를 NTTS 음성으로 제공합니다. NTTS 음성 목록은 신경 음성을 참조하세요.	2021년 11월 18일
NTTS에 새 음성 추가	이제 Amazon Polly에서 이탈리아어 음성 Bianca와 유럽 스페인어 음성 Lucia를 NTTS 음성으로 제공합니다. NTTS 음성 목록은 신경 음성을 참조하세요.	2021년 11월 8일
NTTS에 새 음성 추가	이제 Amazon Polly에서 새로운 남아프리카 영어 음성인 Ayanda를 제공합니다. 이 음성은 NTTS 음성으로만 사용할 수 있습니다. NTTS 음성 목록은 신경 음성을 참조하세요.	2021년 9월 1일
새 리전 추가	이제 아프리카(케이프타운) AWS 리전에서 Amazon Polly를 사용할 수 있습니다. 이 리전은 신경 TTS(NTTS)를 지원합니다. 자세한 내용은 기능 및 리전 호환성의 NTTS를 지원하는 리전 목록을 참조하세요.	2021년 9월 1일
새 언어 및 음성 추가	이제 Amazon Polly에서 뉴질랜드 영어(en-NZ)를 지원합니다. 새로운 NTTS 음성인 Aria는 뉴질랜드 영어와 일부 마오리 단어를 구사합니다.	2021년 8월 24일

새 기능	Amazon Polly는 대화식 말투에서 기본적으로 신경 Matthew 및 Joanna 음성을 사용합니다. 대화식 말투에 대한 참조를 삭제했습니다.	2021년 6월 28일
NTTS에 새 음성 추가	이제 Amazon Polly에서 독일어 음성 Vicki를 NTTS 음성으로 제공합니다.	2021년 6월 15일
새 음성 추가	프랑스(캐나다)(FR-CA) 로케일에 새로운 여성 목소리인 Gabrielle이 추가되었습니다. 이 음성은 품질이 매우 높으며 NTTS 음성으로만 사용할 수 있습니다. 다른 모든 신경 음성과 마찬가지로 이 음성도 특정 리전에서만 사용할 수 있습니다. 리전 목록은 기능 및 리전 호환성을 참조하세요.	2021년 6월 1일
NTTS에 새 음성 추가	이제 Amazon Polly에서 한국어 음성 Seoyeon을 NTTS 음성으로 제공합니다.	2021년 5월 11일
NTTS에 새 리전 추가	Amazon Polly는 이제 캐나다(중부) 지역에서 신경망 TTS(NTTS) 를 지원합니다. AWS 자세한 내용은 NTTS의 기능 및 리전 호환성을 참조하세요.	2021년 3월 17일

새로운 뉴스 진행자 말투 음성 제공	Matthew, Joanna 및 Lupe 음성의 뉴스 진행자 말투 외에도 Amazon Polly에서 이러한 말투에 대한 추가 옵션을 제공합니다. 신경 엔진을 사용하면 영국 영어 Amy 음성에 뉴스 진행자 스타일을 적용할 수 있습니다. 자세한 내용은 NTTS 말투를 참조하세요.	2020년 11월 10일
NTTS에 새 리전 추가	기존 NTTS 리전(us-east-1, us-west-2, eu-west-1, ap-southeast-2)에 더하여 이제 다음 리전에서 신경 음성을 지원합니다. 이들은 ap-northeast-1(도쿄), ap-southeast-1(싱가포르), eu-central-1(프랑크프루트) 및 eu-west-2(런던)입니다. 자세한 내용은 NTTS의 기능 및 리전 호환성을 참조하세요.	2020년 9월 3일
새 음성 추가	어린이 목소리 Ivy와 Justin 이외에도 새로운 남자 어린이 목소리인 Kevin이 미국 영어(en-US)에 추가되었습니다. 이 새로운 음성은 품질이 매우 높으며 NTTS 음성으로만 사용할 수 있습니다. 다른 모든 신경 음성과 마찬가지로 us-east-1(버지니아 북부), us-west-2(오레곤), eu-west-1(아일랜드) 및 ap-southeast-2(시드니)라는 4개 리전에서만 지원됩니다. 자세한 내용은 NTTS 음성을 참조하세요.	2020년 6월 16일

새로운 뉴스 진행자 말투 음성 제공	Matthew 및 Joanna 음성의 뉴스 진행자 말투 외에도 이제 Amazon Polly에서 이러한 말투에 대한 추가 옵션을 제공합니다. 신경 엔진을 사용하면 스페인어(미국) Lupe 음성에 뉴스 진행자 스타일을 사용할 수 있습니다. 자세한 내용은 NTTS 말투를 참조하세요.	2020년 4월 16일
새 기능	뉴스 진행자 말투 외에도 이제 Amazon Polly에서 더 나은 텍스트 투 스피치 구절을 합성하는 데 도움이 되는 두 번째 NTTS 말투를 제공합니다. 대화식 말투는 신경 시스템을 사용하여 많은 사용 사례에서 사용할 수 있는 보다 친근하고 표현적인 대화식 말투로 스피치를 생성합니다. 자세한 내용은 NTTS 말투를 참조하세요.	2019년 11월 25일
새 음성 추가	두 개의 새로운 음성인 Camila(여성, 포르투갈어-브라질)와 Lupe(여성, 스페인어-미국)가 추가되었습니다.	2019년 10월 23일
새 기능 추가	Windows 플러그인용 Amazon Polly를 추가하면 광범위한 Amazon Polly 음성을 Windows SAPI 규정 준수 애플리케이션에 통합할 수 있습니다.	2019년 9월 26일

새로운 주요 기능	Amazon Polly는 출시 이후 Amazon Polly에서 지원하는 표준 text-to-speech (TTS) 음성 외에도 이제 훨씬 더 높은 품질의 음성을 제공할 수 있는 개선된 Neural TTS (NTTS) 시스템을 제공하여 가능한 가장 자연스럽고 사람과 비슷한 음성을 제공합니다. text-to-speech 자세한 내용은 신경 텍스트 투 스피치를 참조하세요.	2019년 7월 30일
새 음성 추가	새 음성인 Lucia(여성, 스페인어) 및 Bianca(여성, 이탈리아어)가 추가되었습니다.	2018년 8월 2일
새 언어 추가	새 언어인 멕시코 스페인어(es-MX)가 추가되었습니다. 이 언어는 Mia의 여성 음성을 사용합니다.	2018년 8월 2일
새 언어 추가	새 언어인 힌디어(hi-IN)가 추가되었습니다. 이 음성은 Aditi의 여성 음성을 사용합니다. Aditi는 인도 영어에도 사용되며, 이는 Amazon Polly의 첫 번째 이중 언어 음성입니다.	2018년 8월 2일
새 기능 추가	긴 텍스트 구절의 스피치 합성이 추가되었습니다(최대 100,000자).	2018년 7월 17일
새 SSML 기능 추가	합성된 스피치의 최대 기간이 추가되었습니다.	2018년 7월 17일
새 음성 추가	새 음성인 Léa(여성, 프랑스어)가 추가되었습니다.	2018년 6월 5일

리전 확장	Amazon Polly 서비스가 모든 상용 리전으로 확장되었습니다.	2018년 6월 4일
새 언어 추가	새 언어인 한국어(ko-KR)가 추가되었습니다.	2018년 6월 4일
확장된 기능	Amazon Polly WordPress 플러그인 기능 (Amazon Translate 기능 추가 포함)	2018년 6월 4일
새 음성 추가	두 개의 새 음성인 Aditi(여성, 인도 영어) 및 Seoyeon(여성, 한국어)이 추가되었습니다.	2017년 11월 15일
새 기능	새로운 스피치 마크 기능이 추가되었으며, SSML 기능이 확장되었습니다.	2017년 4월 19일
새 안내서	이 안내서는 Amazon Polly 개발자 안내서의 최초 릴리스입니다.	2016년 11월 30일

AWS 용어집

최신 AWS 용어는 AWS 용어집 참조서의 AWS 용어집을 참조하세요.

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.