

---

# Amazon Polly

개발자 안내서



## Amazon Polly: 개발자 안내서

Copyright © 2020 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

## Table of Contents

Amazon Polly란 무엇입니까?	1
Amazon Polly를 처음 사용하십니까?	1
작동 방식	3
다음 단계	3
시작하기	4
1단계: 계정 및 사용자 설정	4
1.1단계: AWS에 가입	4
2단계: IAM 사용자 만들기	4
다음 단계	5
2단계: 시작하기(콘솔)	5
연습 1: 스피치 합성 빠른 시작(콘솔)	5
연습 2: 일반 텍스트 입력을 사용하여 스피치 합성(콘솔)	6
다음 단계	6
3단계: 시작하기(AWS CLI)	7
3.1단계: AWS CLI 설정	7
3.2 단계: 연습 시작하기	8
Python 예제	10
Python 설정 및 예제 테스트(SDK)	10
Amazon Polly의 음성	12
사용 가능한 음성	12
이중 언어 음성	14
음성 톤기	15
음성 속도	15
음성 속도 변경	16
Amazon Polly에서 지원하는 언어	17
지원되는 언어의 음소 및 Viseme 표	18
신경 TTS	72
기능 및 리전 호환성	72
음성 엔진	73
음성 엔진 선택(콘솔)	73
음성 엔진 선택(CLI)	75
신경 음성	75
NTTS 말투	76
뉴스 진행자 스타일 사용	76
대화 스타일 사용	77
스피치 마크	79
스피치 마크 형식	79
Viseme 및 Amazon Polly	79
스피치 마크 사용	80
스피치 마크 요청	80
스피치 마크 출력	81
스피치 마크 예제	81
스피치 마크 요청(콘솔)	83
SSML 사용	84
예약 문자	84
콘솔에서 SSML 사용	86
AWS CLI에서 SSML 사용	87
Synthesize-Speech 명령을 통한 SSML 사용	88
SSML로 확장된 문서 합성	88
일반 Amazon Polly 작업에 대한 SSML 사용	89
지원되는 SSML 태그	91
SSML로 확장된 텍스트 식별	92
일시 중지 추가	92
단어 강조	93

특정 단어에 대하여 다른 언어 지정 .....	94
텍스트에 사용자 지정 태그 배치 .....	95
단락 사이에 일시 중지 추가 .....	95
철자대로의 발음 사용 .....	95
볼륨, 말하기 속도 및 피치 제어 .....	96
합성된 스피치의 최대 기간 설정 .....	98
문장 사이에 일시 중지 추가 .....	100
특별한 유형의 단어를 말하는 방식 제어 .....	100
두문자어 및 약어 발음 .....	103
스피치의 일부를 구체적으로 지정함으로써 발음 개선 .....	103
숨소리 추가 .....	104
대화 말투 .....	106
뉴스 진행자 말투 .....	107
동적 범위 압축 추가 .....	107
부드러운 어투 .....	109
음색 제어 .....	109
속삭임 기능 .....	110
어휘 관리 .....	112
여러 어휘 적용 .....	112
콘솔을 사용하여 어휘 관리 .....	113
콘솔을 사용하여 어휘 업로드 .....	113
콘솔을 사용하여 어휘 적용(스피치 합성) .....	114
콘솔을 사용하여 어휘 목록 필터링 .....	115
콘솔을 사용하여 어휘 다운로드 .....	115
콘솔을 사용하여 어휘 삭제 .....	115
AWS CLI를 사용하여 어휘 관리 .....	116
PutLexicon .....	116
GetLexicon .....	120
ListLexicons .....	121
DeleteLexicon .....	121
긴 오디오 파일 생성 .....	123
비동기식 합성에 대한 IAM 정책 설정 .....	123
긴 오디오 파일 생성(콘솔) .....	124
긴 오디오 파일 생성(CLI) .....	125
코드 및 애플리케이션 예제 .....	128
샘플 코드 .....	128
Java 샘플 .....	128
Python 샘플 .....	134
예제 애플리케이션 .....	139
Python 예제 .....	139
Java 예제 .....	148
iOS 예제 .....	151
Android 예제 .....	152
Windows용 Amazon Polly(SAPI) .....	154
Windows용 Amazon Polly(SAPI) 설치 및 구성 .....	154
AWS 클라이언트용 IAM 사용자 생성 .....	154
Windows용 AWS CLI 설치 .....	155
AWS 클라이언트용 프로필 생성 .....	155
Windows용 Amazon Polly 플러그인 설치 .....	155
애플리케이션에서 Amazon Polly 사용 .....	156
AWS for WordPress 플러그인 .....	159
설치를 위한 사전 조건 .....	159
AWS 계정 생성 .....	159
IAM 사용자 생성 .....	159
WordPress 웹 사이트 만들기 .....	161
플러그인 설치 및 구성 .....	161
플러그인 구성 .....	161

WordPress 사용자 지정	162
오디오 파일 미세 조정을 위한 플러그인 설정의 조정	162
콘텐츠에서 SSML을 사용하여 읽혀지는 방식을 수정합니다.	163
콘텐츠에서 오디오 전용 및 워드 전용 태그 사용	163
게시물에 번역된 텍스트 추가	164
Amazon Pollycast	164
플레이어 위치 지정	165
오디오 파일 저장	165
할당량	167
지원하는 리전	167
Throttling	167
발음 어휘	168
SynthesizeSpeech API 작업	168
SpeechSynthesisTask API 작업	168
SSML(Speech Synthesis Markup Language)	168
보안	170
데이터 보호	170
저장 데이터 암호화	171
전송 중 데이터 암호화	171
인터넷워크 트래픽 개인 정보	171
자격 증명 및 액세스 관리	171
대상	171
자격 증명을 통한 인증	172
정책을 사용하여 액세스 관리	173
Amazon Polly에서 IAM을 사용하는 방법	174
자격 증명 기반 정책 예제	176
Amazon Polly API 권한 참조	180
로깅 및 모니터링	181
규정 준수 확인	182
복원성	182
인프라 보안	182
보안 모범 사례	183
AWS CloudTrail을 사용하여 Amazon Polly API 호출 로깅	184
CloudTrail의 Amazon Polly 정보	184
예제: Amazon Polly 로그 파일 항목	185
CloudWatch 통합	186
CloudWatch 측정치 가져오기(콘솔)	186
CloudWatch 측정치 가져오기(CLI)	186
Amazon Polly 지표	187
Amazon Polly 지표 차원	187
API 참조	189
Actions	189
DeleteLexicon	190
DescribeVoices	192
GetLexicon	195
GetSpeechSynthesisTask	197
ListLexicons	199
ListSpeechSynthesisTasks	201
PutLexicon	203
StartSpeechSynthesisTask	205
SynthesizeSpeech	211
Data Types	215
Lexicon	216
LexiconAttributes	217
LexiconDescription	219
SynthesisTask	220
Voice	223

문서 기록 .....	225
AWS Glossary .....	227

# Amazon Polly란 무엇입니까?

Amazon Polly는 텍스트를 생생한 음성으로 변환하는 서비스입니다. Amazon Polly를 사용하여 참여와 접근성을 높여주는 애플리케이션을 개발할 수 있습니다. Amazon Polly는 여러 언어를 지원하며 다양한 실제 음성이 포함되어 있어서 여러 위치에서 고객에게 적합한 음성을 사용하는 음성 지원 애플리케이션을 개발할 수 있습니다. Amazon Polly에서는 합성한 텍스트에 대해서만 비용을 지불합니다. 또한 Amazon Polly에서 생성된 음성을 캐시하여 추가 요금 없이 재생할 수 있습니다.

또한 Amazon Polly에는 여러 NTTS(신경 텍스트 투 스피치) 음성이 있어 새로운 기계 학습 방식을 통해 스피치 품질이 획기적으로 향상되므로 고객에게 최대한 자연스럽고 인간과 유사한 텍스트 투 스피치 음성을 제공할 수 있습니다. 신경 TTS 기술은 뉴스 해설 사용 사례에 맞는 뉴스 진행자 말투도 지원합니다.

Amazon Polly의 일반적인 사용 사례에는 뉴스리더, 게임, 전자 학습 플랫폼, 시각 장애가 있는 사람을 위한 접근성 애플리케이션, 빠르게 성장하는 사물 인터넷(IoT) 세그먼트 등과 같은 모바일 애플리케이션이 포함되거나 이에 국한되지 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.

Amazon Polly를 사용하면 다음과 같은 이점이 있습니다.

- **우수한 품질** – Amazon Polly에서는 새로운 신경 TTS와 동급최고의 표준 TTS 기술을 제공하여 자연 음성을 정확한 발음으로 합성합니다(약어, 두문자어 확장, 날짜/시간 해석, 명확한 동음이의어 구별 포함).
- **짧은 지연 시간** – Amazon Polly에서는 빠른 응답을 보장하므로 대화 시스템처럼 지연 시간이 짧아야 하는 사용 사례에 맞는 옵션입니다.
- **다양한 언어 및 음성 포트폴리오 지원** – Amazon Polly에서는 수십 가지 음성 언어를 지원하며 대부분의 언어에서 남성 및 여성 음성 옵션을 제공합니다. 신경 TTS는 현재 3가지 영국 영어 음성과 8가지 미국 영어 음성을 지원합니다. 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 효율성** – Amazon Polly는 종량제 모델을 사용하므로 설정 비용이 없습니다. 따라서 소규모로 시작하여 애플리케이션의 성장에 따라 규모를 확장할 수 있습니다.
- **클라우드 기반 솔루션** – 디바이스 기반 TTS 솔루션을 사용하려면 충분한 컴퓨팅 리소스, 매우 높은 CPU 전력, RAM 및 디스크 공간이 필요합니다. 따라서 개발 비용이 높고 태블릿, 스마트폰 등의 디바이스에서 전력 소비가 높아집니다. 반면에 AWS 클라우드에서 수행되는 TTS 변환은 로컬 리소스 요구 사항을 획기적으로 줄여줍니다. 따라서 모든 사용 가능한 언어와 음성을 최상의 품질로 지원할 수 있습니다. 또한 음성 개선 사항이 모든 최종 사용자에게 즉시 제공되므로 디바이스에 대한 추가 업데이트가 필요하지 않습니다.

## Amazon Polly를 처음 사용하십니까?

Amazon Polly를 처음 사용하는 경우, 먼저 다음 단원을 나열된 순서대로 읽어보십시오.

1. [Amazon Polly 작동 방식 \(p. 3\)](#) – 이 섹션에서는 종단 간 경험 생성에 사용 가능한 다양한 Amazon Polly 입력 및 옵션을 소개합니다.

2. [Amazon Polly 시작하기 \(p. 4\)](#) – 이 섹션에서는 계정을 설정하고 Amazon Polly 스피치 합성을 테스트합니다.
3. [예제 애플리케이션 \(p. 139\)](#) – 이 섹션에서는 Amazon Polly 탐색에 사용할 수 있는 추가 예제를 제공합니다.



# Amazon Polly 작동 방식

Amazon Polly는 입력 텍스트를 실제 같은 음성으로 변환합니다. 스피치 합성 메서드 중 하나를 호출하고, 합성할 텍스트를 제공한 후 NTTS(신경 텍스트 투 스피치) 또는 표준 TTS(텍스트 투 스피치) 음성 중 하나를 선택하고, 오디오 출력 형식을 지정합니다. 그러면 Amazon Polly에서 제공된 텍스트를 고품질 음성 오디오 스트림으로 합성합니다.

- **입력 텍스트** – 합성하려는 텍스트를 제공하면 Amazon Polly에서 오디오 스트림을 반환합니다. 입력은 일반 텍스트 또는 SSML(Speech Synthesis Markup Language) 형식으로 제공할 수 있습니다. SSML을 사용하면 발음, 볼륨, 음조 및 말하기 속도 등 음성의 다양한 측면을 제어할 수 있습니다. 자세한 내용은 [SSML 문서로부터 스피치 생성 \(p. 84\)](#)를 참조하십시오.
- **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 Polly가 이 음성을 사용하여 텍스트를 스피치로 변환합니다. Amazon Polly는 번역 서비스가 아닙니다. 합성된 스피치는 텍스트와 같은 언어입니다. 하지만 음성에 지정된 것과 다른 언어로 텍스트가 제공되면 아라비아 숫자(예: 쉰셋이 아니라 53)가 텍스트가 아닌 음성 언어로 합성됩니다. 자세한 내용은 [Amazon Polly의 음성](#)을 참조하십시오.
- **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, AWS IoT 디바이스 및 텔레포니 솔루션에 사용할 수 있게 PCM 출력 형식을 요청할 수 있습니다.

## 다음 단계

Amazon Polly를 처음 사용하는 경우, 먼저 다음 항목을 순서대로 읽어보십시오.

- [Amazon Polly 시작하기 \(p. 4\)](#)
- [예제 애플리케이션 \(p. 139\)](#)
- [Amazon Polly의 할당량 \(p. 167\)](#)

# Amazon Polly 시작하기

Amazon Polly는 기존 애플리케이션과 쉽게 통합할 수 있는 간단한 API 작업을 제공합니다. 지원되는 작업 목록은 [Actions \(p. 189\)](#) 항목을 참조하십시오. 다음 두 가지 옵션 중 하나를 사용할 수 있습니다.

- AWS SDK – SDK를 사용할 때 Amazon Polly에 대한 요청은 사용자가 제공한 자격 증명을 사용하여 자동으로 서명되고 인증됩니다. 이것은 애플리케이션을 구축할 때 권장되는 선택 사항입니다.
- AWS CLI – AWS CLI를 사용하여 코드를 작성하지 않고 Amazon Polly 기능에 액세스할 수 있습니다.

다음 섹션에서는 설정 및 입문용 실습을 제공하는 방법에 대해 설명합니다.

주제

- 1단계: AWS 계정 설정 및 사용자 만들기 (p. 4)
- 2단계: 시작하기(콘솔) (p. 5)
- 3단계: 시작하기(AWS CLI) (p. 7)
- Python 예제 (p. 10)

## 1단계: AWS 계정 설정 및 사용자 만들기

Amazon Polly를 처음 사용하는 경우, 먼저 다음 작업을 완료해야 합니다.

1. 1.1단계: AWS에 가입 (p. 4)
2. 2단계: IAM 사용자 만들기 (p. 4)

### 1.1단계: AWS에 가입

Amazon Web Services(AWS)에 가입하면 Amazon Polly를 포함한 AWS의 모든 서비스에 AWS 계정이 자동으로 등록됩니다. 사용한 서비스에 대해서만 청구됩니다.

Amazon Polly에서는 사용한 리소스에 대해서만 비용을 지불합니다. AWS를 처음 사용하는 고객인 경우 Amazon Polly를 무료로 시작할 수 있습니다. 자세한 내용은 [AWS 프리 티어](#)를 참조하십시오.

이미 AWS 계정이 있다면 다음 단계로 건너뛰십시오. AWS 계정이 없는 경우에는 다음 절차의 단계를 수행하여 계정을 만듭니다.

AWS 계정을 생성하려면

1. <https://portal.aws.amazon.com/billing/signup>을 엽니다.
2. 온라인 지시 사항을 따릅니다.

등록 절차 중 전화를 받고 전화 키패드를 사용하여 확인 코드를 입력하는 과정이 있습니다.

다음 단계에 필요하므로 AWS 계정 ID를 기록합니다.

### 2단계: IAM 사용자 만들기

Amazon Polly 등의 AWS 서비스에 액세스하려면 자격 증명을 제공해야 하는데, 이는 그 서비스가 소유한 리소스에 대한 액세스 권한이 있는지 확인할 수 있기 위한 것입니다. 콘솔은 암호를 요구합니다. AWS 계정에 대한 액세스 키를 생성하면 AWS CLI 또는 API에 액세스할 수 있습니다. 그러나 AWS 계정용 자격 증명을 사용하여 AWS에 액세스하지 않는 것이 좋습니다. 대신 AWS Identity and Access Management(IAM) 사용을

권장합니다. 관리자 권한을 사용하여 IAM 사용자를 생성하고 IAM 그룹에 추가한 다음 생성한 IAM 사용자에게 관리자 권한을 부여합니다. 그러면 특정 URL과 해당 IAM 사용자의 자격 증명을 사용하여 AWS에 액세스할 수 있습니다.

AWS에 가입했지만 IAM 사용자를 만들지 않았다면, IAM 콘솔을 사용하여 사용자를 만들 수 있습니다.

이 안내서의 연습에서는 관리자 권한을 가진 사용자(adminuser)가 있다고 가정합니다. 절차에 따라 계정에 adminuser를 만듭니다.

관리자 사용자를 만들고 콘솔에 로그인하려면

1. AWS 계정에서 adminuser라는 관리자 사용자를 만듭니다. 지침은 IAM 사용 설명서의 [첫 번째 IAM 사용자 및 관리자 그룹 생성](#) 단원을 참조하십시오.
2. 사용자는 특별 URL을 사용하여 AWS Management 콘솔에 로그인할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 [사용자의 계정 로그인 방법](#) 단원을 참조하십시오.

### Important

관리자 자격증명을 사용하는 연습 시작하기 보안 강화를 위해 프로덕션 애플리케이션을 구축하고 테스트할 때 Amazon Polly 작업에 대해서만 권한이 있는 서비스 관련 관리자를 생성하는 것이 좋습니다. Amazon Polly 특정 권한을 부여하는 정책에 대한 예제는 [예제 1: 모든 Amazon Polly 작업 허용](#) (p. 179) 항목을 참조하십시오.

IAM에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Identity and Access Management\(IAM\)](#)
- [시작하기](#)
- [IAM 사용 설명서](#)

## 다음 단계

2단계: 시작하기(콘솔) (p. 5)

## 2단계: 시작하기(콘솔)

Amazon Polly 콘솔은 Amazon Polly의 스피치 합성을 테스트하고 사용하기 위한 가장 간단한 방법입니다. Amazon Polly 콘솔은 일반 텍스트 또는 SSML 입력에서 스피치 합성을 지원합니다.

주제

- [연습 1: 스피치 합성 빠른 시작\(콘솔\) \(p. 5\)](#)
- [연습 2: 일반 텍스트 입력을 사용하여 스피치 합성\(콘솔\) \(p. 6\)](#)
- [다음 단계 \(p. 6\)](#)

## 연습 1: 스피치 합성 빠른 시작(콘솔)

빠른 시작은 스피치 품질을 위한 Amazon Polly 스피치 합성을 테스트하는 가장 빠른 방법을 안내합니다. 텍스트 두 스피치 탭을 선택하면 텍스트를 입력하기 위한 텍스트 필드에 예제 텍스트가 미리 로드되어 Amazon Polly를 신속하게 사용해 볼 수 있습니다.

Amazon Polly를 빠르게 테스트하려면(콘솔)

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/polly/>에서 Amazon Polly 콘솔을 엽니다.

2. [Text-to-Speech] 탭을 선택합니다.
3. 일반 텍스트를 선택합니다.
4. 엔진에서 Neural의 Standard를 선택합니다.
5. 언어 및 AWS 리전을 선택한 후 음성을 선택합니다. 엔진에 대해 Neural을 선택하면 미국 영어만 언어에 사용할 수 있고 NTTS가 지원되는 리전만 표시됩니다. 모든 표준 음성은 비활성화됩니다.
6. [Listen to speech]을 선택합니다.

심층 테스트에 대한 자세한 내용은 다음 항목을 참조하십시오.

- [연습 2: 일반 텍스트 입력을 사용하여 스피치 합성\(콘솔\) \(p. 6\)](#)
- [SSML\(콘솔\) 사용 \(p. 86\)](#)
- [콘솔을 사용하여 어휘 적용\(스피치 합성\) \(p. 114\)](#)

## 연습 2: 일반 텍스트 입력을 사용하여 스피치 합성(콘솔)

다음 절차에서는 일반 텍스트를 입력하여 스피치를 합성합니다. "W3C"와 날짜 "10/3"(10월 3일)이 어떻게 합성되는지 확인하십시오.

일반 텍스트 입력을 사용하여 스피치를 합성하려면(콘솔)

1. Amazon Polly 콘솔에 로그인하고 시작하기를 선택한 다음 텍스트 투 스피치 탭을 선택합니다.
2. [Plain text] 탭을 선택합니다.
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. 엔진에서 Standard 또는 Neural을 선택합니다.
5. 언어 및 AWS 리전을 선택한 후 음성을 선택합니다. 엔진에 대해 Neural을 선택하면 미국 영어만 언어에 사용할 수 있고 NTTS가 지원되는 리전만 표시됩니다. 모든 표준 음성은 비활성화됩니다.
6. 스피치를 즉시 들으려면 [Listen to speech]를 선택합니다.
7. 스피치를 파일로 저장하려면 다음 중 한 가지를 실행합니다.
  - a. [Save speech to MP3]를 선택합니다.
  - b. 다른 파일 형식으로 변경하려면 파일 형식 변경을 선택하여 원하는 파일 형식을 선택한 다음 변경을 선택합니다.

심층 예제에 대한 자세한 내용은 다음 항목을 참조하십시오.

- [콘솔을 사용하여 어휘 적용\(스피치 합성\) \(p. 114\)](#)
- [SSML\(콘솔\) 사용 \(p. 86\)](#)

## 다음 단계

3단계: 시작하기(AWS CLI) (p. 7)

## 3단계: 시작하기(AWS CLI)

AWS Command Line Interface(AWS CLI)를 사용하면 Amazon Polly 콘솔에서 할 수 있는 거의 모든 Amazon Polly 작업을 수행할 수 있습니다. AWS CLI를 사용하여 합성된 스피치를 들을 수 없습니다. 대신에 스피치를 파일에 저장한 후 재생할 수 있는 애플리케이션에서 파일을 열어야 합니다.

주제

- [3.1단계: AWS Command Line Interface\(AWS CLI\) 설정 \(p. 7\)](#)
- [3.2 단계: AWS CLI 사용 연습 시작하기 \(p. 8\)](#)

### 3.1단계: AWS Command Line Interface(AWS CLI) 설정

단계에 따라 AWS CLI를 다운로드하고 구성합니다.

Important

AWS CLI가 없어도 이 연습에 나오는 단계를 수행할 수 있지만 이 가이드의 일부 연습에서는 AWS CLI를 사용합니다. 이 단계를 건너뛰어 [3.2 단계: AWS CLI 사용 연습 시작하기 \(p. 8\)](#)으로 이동한 다음 나중에 필요할 때 AWS CLI를 설정할 수 있습니다.

AWS CLI을(를) 설정하려면

1. AWS CLI을(를) 다운로드하고 구성합니다. 지침은 AWS Command Line Interface 사용 설명서에서 다음 항목을 참조하십시오.
  - [AWS Command Line Interface를 이용한 설정](#)
  - [AWS Command Line Interface 구성](#)
2. AWS CLI 구성 파일에서 관리자 사용자의 명명된 프로필을 추가합니다. AWS CLI 명령을 실행할 때 이 프로파일을 사용합니다. 명명된 프로필에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서의 [명명된 프로필](#)을 참조하십시오.

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

Amazon Polly에서 지원되는 사용 가능한 AWS 리전의 목록은 Amazon Web Services 일반 참조의 [리전 및 종단점](#) 단원을 참조하십시오.

Note

AWS CLI를 구성할 때 지정한 Amazon Polly 지원 리전을 사용하고 있다면 AWS CLI 코드 예제에서 다음 줄을 생략하십시오.

```
--region aws-region
```

3. 명령 프롬프트에서 다음 help 명령을 입력하여 설정을 확인합니다.

```
aws help
```

유효한 AWS 명령 목록이 AWS CLI 창에 나타납니다.

## AWS CLI에서 Amazon Polly를 활성화하려면(선택 사항)

전에 AWS CLI를 다운로드하여 구성한 적이 있으면 Amazon Polly를 사용하기 위해 AWS CLI를 다시 구성해야 할 수 있습니다. 이 절차는 해당 사항이 필요한지 여부를 확인하고 Amazon Polly가 자동으로 사용할 수 없는 경우 지침을 제공합니다.

1. AWS CLI 명령 프롬프트에서 다음 help 명령을 입력하여 Amazon Polly의 사용 가능 여부를 확인하십시오.

```
aws polly help
```

Amazon Polly에 대한 설명과 유효한 명령 목록이 AWS CLI 창에 나타나면 AWS CLI에서 Amazon Polly가 지원되며 바로 사용할 수 있습니다. 이 경우 이 절차의 나머지 부분은 건너뛸 수 있습니다. 이 메시지가 표시되지 않으면 2단계로 진행합니다.

2. Amazon Polly를 사용하려면 다음 두 옵션 중 하나를 사용하십시오.
  - a. AWS CLI를 제거하고 다시 설치하십시오.

자세한 내용은 AWS Command Line Interface 사용 설명서의 [AWS Command Line Interface 설치](#)를 참조하십시오.

또는

- b. Download the file [service-2.json](#) 파일 다운로드

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.

## 다음 단계

### 3.2 단계: AWS CLI 사용 연습 시작하기 (p. 8)

## 3.2 단계: AWS CLI 사용 연습 시작하기

이제 Amazon Polly에서 제공하는 스피치 합성을 테스트할 수 있습니다. 이 연습에서는 샘플 텍스트를 전달하여 SynthesizeSpeech 작업을 호출합니다. 결과 오디오를 파일로 저장하고 콘텐츠를 확인할 수 있습니다.

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

다음은 Unix, Linux, macOS용 형식으로 지정된 AWS CLI 예제입니다. 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
```

synthesize-speech 호출에서 합성을 위한 샘플 텍스트, 사용할 음성(다음 3단계에서 설명하는 음성 ID 제공)과 출력 형식을 제공합니다. 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"
    },
    {
      "Gender": "Female",
      "Name": "Joanna",
      "LanguageName": "US English",
      "Id": "Joanna",
      "LanguageCode": "en-US"
    }
  ]
}
```

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 코드의 목록은 [Amazon Polly에서 지원하는 언어 \(p. 17\)](#)를 참조하십시오. 이러한 언어 코드는 W3C 언어 식별 태그(## ### ## ISO 639 ##-ISO 3166 ## ##)입니다. 예로, en-US(미국 영어), en-GB(영국 영어) 및 es-ES(스페인어) 등입니다.

AWS CLI에서 help 옵션을 사용하여 언어 코드 목록을 얻을 수도 있습니다.

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

## Python 예제

이 가이드는 추가 예제를 제공하며 일부는 AWS SDK for Python (Boto)을 사용하여 Amazon Polly에 API를 호출하는 Python 코드 예제입니다. Python을 설정하고 다음 단원에 나오는 예제 코드를 테스트하는 것이 좋습니다. 추가 예제는 다음(예제 애플리케이션 (p. 139))을 참조하십시오.

### Python 설정 및 예제 테스트(SDK)

Python 예제 코드를 테스트하려면 AWS SDK for Python (Boto)이 필요합니다. 자세한 방법은 [Python용 AWS SDK\(Boto3\)](#)를 참조하십시오.

Python 코드 예제를 테스트하려면

다음 Python 코드 예제는 다음과 같은 작업을 수행합니다.

- AWS SDK for Python (Boto)를 사용하여 간단한 텍스트를 입력으로 제공하여 Amazon Polly에 SynthesizeSpeech 요청을 보냅니다.
- 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 예제에 대해서는 [예제 애플리케이션 \(p. 139\)](#)을 참고하십시오.

# Amazon Polly의 음성

## 사용 가능한 음성

Amazon Polly는 텍스트에서 스피치를 합성하기 위한 다양한 음성을 여러 언어로 제공합니다.

언어	이름/ID	성별	표준 음성	신경 음성
아랍어(arb)	Zeina	여성	예	아니요
표준 중국어(cmn-CN)	Zhiyu	여성	예	아니요
덴마크어(da-DK)	Naja	여성	예	아니요
	Mads	남성	예	아니요
네덜란드어(nl-NL)	Lotte	여성	예	아니요
	Ruben	남성	예	아니요
영어(호주)(en-AU)	Nicole	여성	예	아니요
	Russell	남성	예	아니요
영어(영국)(en-GB)	Amy	여성	예	예
	Emma	여성	예	예
	Brian	남성	예	예
영어(인도)(en-IN)	Aditi*	여성	예	아니요
	Raveena	여성	예	아니요
영어(미국)(en-US)	Ivy	여성(어린이)	예	예
	Joanna**	여성	예	예
	Kendra	여성	예	예
	Kimberly	여성	예	예
	Salli	여성	예	예
	Joey	남성	예	예
	Justin	남성(어린이)	예	예
	Matthew**	남성	예	예
영어(웨일스)(en-GB-WLS)	Geraint	남성	예	아니요
프랑스어(fr-FR)	Céline/Celine	여성	예	아니요

Amazon Polly 개발자 안내서  
사용 가능한 음성

언어	이름/ID	성별	표준 음성	신경 음성
	Léa	여성	예	아니요
	Mathieu	남성	예	아니요
프랑스어(캐나다) (fr-CA)	Chantal	여성	예	아니요
독일어(de-DE)	Marlene	여성	예	아니요
	Vicki	여성	예	아니요
	Hans	남성	예	아니요
힌디어(hi-IN)	Aditi*	여성	예	아니요
아이슬란드어(is-IS)	Dóra/Dora	여성	예	아니요
	Karl	남성	예	아니요
이탈리아어(it-IT)	Carla	여성	예	아니요
	Bianca	여성	예	아니요
	Giorgio	남성	예	아니요
일본어(ja-JP)	Mizuki	여성	예	아니요
	Takumi	남성	예	아니요
한국어(ko-KR)	Seoyeon	여성	예	아니요
노르웨이어(nb-NO)	Liv	여성	예	아니요
폴란드어(pl-PL)	Ewa	여성	예	아니요
	Maja	여성	예	아니요
	Jacek	남성	예	아니요
	1월	남성	예	아니요
포르투갈어(브라질) (pt-BR)	Camila	여성	예	예
	Vitória/Vitoria	여성	예	아니요
	Ricardo	남성	예	아니요
포르투갈어(유럽) (pt-PT)	Inês/Ines	여성	예	아니요
	Cristiano	남성	예	아니요
루마니아어(ro-RO)	Carmen	여성	예	아니요
러시아어(ru-RU)	Tatyana	여성	예	아니요
	Maxim	남성	예	아니요

언어	이름/ID	성별	표준 음성	신경 음성
스페인어(유럽)(es-ES)	Conchita	여성	예	아니요
	Lucia	여성	예	아니요
	Enrique	남성	예	아니요
스페인어(멕시코)(es-MX)	Mia	여성	예	아니요
스페인어(미국)(es-US)	Lupe	여성	예	예
	Penélope/ Penelope	여성	예	아니요
	Miguel	남성	예	아니요
스웨덴어(sv-SE)	Astrid	여성	예	아니요
터키어(tr-TR)	Filiz	여성	예	아니요
웨일스어(cy-GB)	Gwyneth	여성	예	아니요

\* 이 음성은 이중 언어이며 영어와 힌디어를 둘 다 할 수 있습니다. 자세한 내용은 [이중 언어 음성 \(p. 14\)](#) 단원을 참조하십시오.

\*\* 이러한 음성은 신경 형식과 함께 사용할 때 대화 및 뉴스 진행자 말투 모두에서 사용할 수 있습니다. 자세한 내용은 [NTTS 말투 \(p. 76\)](#) 단원을 참조하십시오.

위의 음성 외에도 Amazon Polly는 브랜드 페르소나를 반영하는 맞춤형 Brand Voice를 작성하여 고객에게 독특하고 독점적인 NTTS 음성을 제공할 수 있는 기회를 제공합니다. Amazon Polly Brand Voice에 대한 자세한 내용을 알아보려면 [https://aws.amazon.com/polly/features/#Brand\\_Voice](https://aws.amazon.com/polly/features/#Brand_Voice)를 참조하십시오.

## 이중 언어 음성

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

현재는 Aditi가 유일한 이중 언어 음성입니다.

이중 언어 음성(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 शुरू 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.

## 음성 듣기

Amazon Polly 콘솔을 사용하여 Amazon Polly에서 사용 가능한 음성 중에 샘플을 들을 수 있습니다.

Amazon Polly에서 음성을 들으려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/polly/>에서 Amazon Polly 콘솔을 엽니다.
2. [Text-to-Speech] 탭을 선택합니다.
3. 엔진에서 표준 또는 Neural(신경)을 선택합니다.
4. 언어 및 리전을 선택한 후 음성을 선택합니다
5. 말하려는 음성의 텍스트를 입력하거나 기본 문구를 사용한 후 음성 듣기를 선택합니다.

Amazon Polly에서 제공하는 언어를 선택할 수 있으며, 콘솔에 해당 언어에 대해 사용 가능한 음성이 표시됩니다. 대부분의 경우 남성과 여성 음성이 하나 이상 있으며, 각각 하나 이상 있는 경우가 많습니다. 음성이 하나만 있는 경우는 적습니다. 전체 목록은 [Amazon Polly의 음성 \(p. 12\)](#) 단원을 참조하십시오.

### Note

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

각 음성은 모국어 사용자를 통해 만들어지므로, 동일한 언어 내에서도 음성마다 차이가 있습니다. 프로젝트의 음성을 선택할 때 텍스트의 한 구절로 가능한 각 음성을 테스트하여 요건에 가장 적합한 음성을 확인해야 합니다.

## 음성 속도

음성 간의 자연적인 차이로 인해 사용 가능한 각 음성은 텍스트를 약간씩 다른 속도로 말합니다. 예를 들어, 미국 영어 음성에서 "Mary had a little lamb"을 말할 때 Ivy와 Joanna는 Matthew보다 조금 더 빠르며, Joey보다 상당히 빠릅니다.

음성 간에 많은 차이가 있고 그 차이의 정도는 말하는 텍스트에 따라 다를 수 있기 때문에, Amazon Polly 음성에서 표준 속도(분당 단어 수)는 사용할 수 없습니다. 그러나 스피치 마크를 사용하여 내 음성이 선택한 텍스트를 말하는 데 걸리는 시간을 찾을 수 있습니다. Amazon Polly에서 스피치 마크 사용에 대한 자세한 내용은 [스피치 마크 사용 \(p. 80\)](#)을 참조하십시오.

텍스트 구절을 말하는 데 걸리는 대략적인 시간을 보려면

1. AWS CLI를 엽니다.
2. 필요에 따라 채워 넣고 다음 코드를 실행합니다.

```
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.txt를 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 높이거나 할 수 있습니다. 음성 속도를 고려해야 할 경우를 위해 Amazon Polly는 SSML 태그를 사용하여 이를 수정하는 기능을 제공합니다.

예:

여러분의 조직이 이민자 청중에게 책을 읽어 주는 애플리케이션을 만들려고 합니다. 이 청중은 영어를 말하지만, 능숙도는 제한적입니다. 이 경우 애플리케이션이 말하는 동안 스피치 속도를 낮춰 청중이 이해할 수 있도록 조금 더 많은 시간을 제공하는 방법을 고려할 수 있습니다.

Amazon Polly에서 다음과 같이 SSML <prosody> 태그를 사용하여 스피치 속도를 낮출 수 있습니다.

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

또는

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

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

- 속도 사전 설정: x-slow, slow, medium, fast, x-fast. 이러한 경우 각 옵션의 속도는 근사값으로, 선호하는 음성에 따라 다릅니다. medium 옵션은 일반 음성 속도입니다.
- 스피치 속도의 n%: 스피치 속도의 백분율로, 20%~200%를 사용할 수 있습니다. 이러한 경우 원하는 정확한 속도를 선택할 수 있습니다. 그러나 실제 음성 속도는 근사값으로, 선택한 음성에 따라 다릅니다. 100%는 일반 음성 속도로 간주됩니다.

각 옵션의 속도는 근사값이며 선택하는 음성에 따라 다를 수 있으므로, 선택한 음성을 다양한 속도로 테스트하여 사용자의 요구에 정확히 맞는 속도를 확인하는 것이 좋습니다.

[prosody](#) 태그를 사용하여 최상의 효과를 내는 방법에 대한 자세한 내용은 [볼륨, 말하기 속도 및 피치 제어 \(p. 96\)](#) 단원을 참조하십시오.

## Amazon Polly에서 지원하는 언어

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

각 언어와 관련된 음소 및 viseme을 보여 주는 자세한 표를 보려면 아래 표에서 각 언어의 링크를 선택합니다.

언어	언어 코드
<a href="#">아랍어(arb) (p. 19)</a>	arb
<a href="#">표준 중국어(cmn-CN) (p. 21)</a>	cmn-CN
<a href="#">덴마크어(da-DK) (p. 23)</a>	da-DK
<a href="#">네덜란드어(nl-NL) (p. 25)</a>	nl-NL
<a href="#">영어, 오스트레일리아(en-AU) (p. 27)</a>	en-AU
<a href="#">영어, 영국(en-GB) (p. 31)</a>	en-GB
<a href="#">영어, 인도(en-IN) (p. 29)</a>	en-IN
<a href="#">영어, 인도(en-IN) (p. 33)</a>	en-US
<a href="#">영어, 웨일스(en-GB-WLS) (p. 34)</a>	en-GB-WLS
<a href="#">프랑스어(fr-FR) (p. 36)</a>	fr-FR
<a href="#">프랑스어, 캐나다(fr-CA) (p. 38)</a>	fr-CA
<a href="#">힌디어(hi-IN) (p. 42)</a>	hi-IN
<a href="#">독일어(de-DE) (p. 40)</a>	de-DE
<a href="#">아이슬란드어(is-IS) (p. 43)</a>	is-IS
<a href="#">이탈리아어(it-IT) (p. 45)</a>	it-IT
<a href="#">일본어(ja-JP) (p. 47)</a>	ja-JP
<a href="#">한국어(ko-KR) (p. 48)</a>	ko-KR
<a href="#">노르웨이어(nb-NO) (p. 50)</a>	nb-NO
<a href="#">폴란드어(pl-PL) (p. 51)</a>	pl-PL
<a href="#">포르투갈어, 브라질(pt-BR) (p. 55)</a>	pt-BR
<a href="#">포르투갈어(pt-PT) (p. 53)</a>	pt-PT
<a href="#">루마니아어(ro-RO) (p. 56)</a>	ro-RO
<a href="#">러시아어(ru-RU) (p. 58)</a>	ru-RU

언어	언어 코드
스페인어(es-ES) (p. 60)	es-ES
스페인어, 멕시코(es-MX) (p. 62)	es-MX
스페인어, 미국(es-US) (p. 63)	es-US
스웨덴어(sv-SE) (p. 65)	sv-SE
터키어(tr-TR) (p. 67)	tr-TR
웨일스어(cy-GB) (p. 69)	cy-GB

자세한 내용은 [지원되는 언어의 음소 및 Viseme 표 \(p. 18\)](#) 단원을 참조하십시오.

## 지원되는 언어의 음소 및 Viseme 표

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

### 주제

- 아랍어(arb) (p. 19)
- 표준 중국어(cmn-CN) (p. 21)
- 덴마크어(da-DK) (p. 23)
- 네덜란드어(nl-NL) (p. 25)
- 영어, 오스트레일리아(en-AU) (p. 27)
- 영어, 인도(en-IN) (p. 29)
- 영어, 영국(en-GB) (p. 31)
- 영어, 미국(en-US) (p. 33)
- 영어, 웨일스(en-GB-WSL) (p. 34)
- 프랑스어(fr-FR) (p. 36)
- 프랑스어, 캐나다(fr-CA) (p. 38)
- 독일어(de-DE) (p. 40)
- 힌디어(hi-IN) (p. 42)
- 아이슬란드어(is-IS) (p. 43)
- 이탈리아어(it-IT) (p. 45)
- 일본어(ja-JP) (p. 47)
- 한국어(ko-KR) (p. 48)
- 노르웨이어(nb-NO) (p. 50)
- 폴란드어(pl-PL) (p. 51)
- 포르투갈어(pt-PT) (p. 53)
- 포르투갈어, 브라질(pt-BR) (p. 55)
- 루마니아어(ro-RO) (p. 56)
- 러시아어(ru-RU) (p. 58)
- 스페인어(es-ES) (p. 60)
- 스페인어, 멕시코(es-MX) (p. 62)
- 스페인어, 미국(es-US) (p. 63)
- 스웨덴어(sv-SE) (p. 65)
- 터키어(tr-TR) (p. 67)



- 웨일스어(cy-GB) (p. 69)

## 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	ʔ	성문 폐쇄음	أنا	
ʕ	ʔ\	유성 인두 마찰음	عَمَر	k
b	b	유성 양순 파열음	بَلَد	p
d	d	유성 치경 파열음	دَارِي	t
dˤ	d_ʔ\	강조 유성 치경 파열음	ضَوء	t
ɗʒ	dZ	유성 후치경 파찰음	جَمِيل	S
ð	D	유성 치 마찰음	ذَلِكَ	T
ðˤ	D_ʔ\	강조 유성 치 마찰음	ظَلَام	T
f	f	무성 순치 마찰음	فَصَل	f
g	g	유성 연구개 파열음	انْجَلْتِرا	k
ɣ	G	유성 연구개 마찰음	عَرَب	k
h	h	무성 성문 마찰음	هَذَا	k
j	j	경구개 접근음	يَمَشِي	i
k	k	무성 연구개 파열음	كَلَب	k
l	l	치경 설측 접근음	لَاقِي	t
lˤ	l_G	강조 치경 설측 접근음	اللّه	t
m	m	양순 비음	مَإِذَا	p
n	n	치경 비음	نور	t
p	p	무성 양순 파열음	حَبَس	p
q	q	무성 구개수 파열음	قَرِيْب	k
r	r	치경 전동음	رَمَل	r
s	s	무성 치경 마찰음	سُؤَال	s
sˤ	s_ʔ\	강조 무성 치경 마찰음	صَاحِب	s

IPA	X-SAMPA	설명	예	Viseme
ʃ	S	무성 후치경 마찰음	شُكْر	S
t	t	무성 치경 파열음	تَمَر	t
tʃ	t_ʔ\	강조 무성 치경 파열음	طالِب	t
θ	T	무성 치 마찰음	ثَلَاث	T
v	v	유성 순치 마찰음	فِيْتَامِيْن	f
w	w	양순 연구개 접근음	وَلَد	u
x	x	무성 연구개 마찰음	خَوْف	k
ħ	X\	무성 인두 마찰음	حَوْل	k
z	z	유성 치경 마찰음	زُهْر	s
모음				
a	a	전설 비원순 저모음	بَرْد	a
a:	a:	전설 비원순 저장모음	دَار	a
aʃ	A_ʔ\	강조 후설 비원순 저모음	طَبَل	a
aʃ:	A_ʔ\:	강조 후설 비원순 저장모음	طالِم	a
u	u	후설 원순 고모음	شُرْب	u
u:	u:	후설 원순 고장모음	سُور	u
uʃ	u_ʔ\	강조 후설 원순 고모음	بُذ	u
uʃ:	u_ʔ\:	강조 후설 원순 고장모음	طُول	u
i	i	전설 비원순 고모음	بِنْت	i
i:	i:	전설 비원순 고장모음	حَزِيْن	i
iʃ	i_ʔ\	강조 전설 비원순 고모음	ضَدّ	i
iʃ:	i_ʔ\:	강조 전설 비원순 고장모음	ماضِي	i
e	e	전설 비원순 중고모음	مارِكْت	e
e:	e:	전설 비원순 중고장모음	موَدِل	e
ɔ	O	후설 원순 중저모음	تَكْنُولُوجِي	O
ɔ:	O:	후설 원순 중저장모음	تَلْفِزِيُون	O

## 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 사용하십시오.

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

병음	IPA	X-SAMPA	설명	병음 예	Viseme
b	p	p	무성 양순 파열음	爸, ba4	p
p	p <sup>h</sup>	p <sub>h</sub>	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	ʃ	s <sup>˘</sup>	무성 권설 마찰음	是, shi4	S
d	t	t	무성 치경 파열음	打, da3	t
t	t <sup>h</sup>	t <sub>h</sub>	기식 무성 치경 파열음	他, ta1	t
zh	ʈʂ	t's <sup>˘</sup>	무성 권설 파찰음	之, zhi1	S
ch	ʈʂ <sup>h</sup>	t's <sup>˘</sup> <sub>h</sub>	기식 무성 권설 파찰음	吃, chi1	S
s	ʈs	ts	무성 치경 파찰음	字, zi4	s
j	ʈɕ	ts\	무성 치경구개 파찰음	鸡, ji1	J
q	ʈɕ <sup>h</sup>	ts\ <sub>h</sub>	기식 무성 치경구개 경파찰음	七, qi1	J
c	ʈs <sup>h</sup>	ts <sub>h</sub>	기식 무성 치경 파찰음	次, ci4	s
w	w	w	양순 연구개 접근음	我, wo3	u
r	ʐ	z <sup>˘</sup>	유성 권설 마찰음	日, ri4	S
"er" 및 "r" 음색을 띤 음절					
er	ʐ	@ <sup>˘</sup>	r 음색을 띤 종설 종모음	二, er4	@
-r			r 음색을 띤 음절	馅儿, xianr4	@
모음					
e	ɤ	7	후설 비원순 중고모음	恶, e4	e
e	ə	@	중설 종모음	恩, en1	@
a	a	a	전설 비원순 저모음	安, an1	a
ai	aɪ	al	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
u	u	u	후설 원순 고모음	主, zhu3	u

병음	IPA	X-SAMPA	설명	병음 예	Viseme
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
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

IPA	X-SAMPA	설명	예	Viseme
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
e	6	중설 근저모음	mor	a
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
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

IPA	X-SAMPA	설명	예	Viseme
ʌ	V	후설 평순 중저	kører	E
y	y	전설 원순 고모음	yt	u
y:	y:	전설 원순 고장모음	hyle	u
추가 기호				
'	"	제1강세	Alabama	
,	%	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
dʒ	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

IPA	X-SAMPA	설명	예	Viseme
t	t	무성 치경 파열음	tak	t
v	v	유성 순치 마찰음	vel	f
ʊ	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				
ø:	2:	전설 원순 중고장모음	neus	o
œy	9y	이중 모음	buit	O
ə	@	중설 중모음	de	@
a:	a:	전설 비원순 저장모음	baad	a
ɑ:	A	후설 비원순 저모음	bad	a
e:	e:	전설 비원순 중고장모음	beet	e
ɜ:	3:	중설 비원순 중저장모음	barrière	E
ɛ	E	전설 비원순 중저모음	bed	E
ɛi	Ei	이중 모음	beet	E
i	i	전설 비원순 고모음	vier	i
ɪ	l	근전설 비원순 근고모음	pit	i
o:	o:	후설 원순 중고장모음	boot	o
ɔ	O	후설 원순 중저모음	pot	O
u	u	후설 원순 고모음	hoed	u
ʌu	Vu	이중 모음	fout	E
y:	y:	전설 원순 고장모음	fuut	u
ʏ	Y	근전설 원순 근고모음	hut	u
추가 기호				
'	"	제1강세	Alabama	
,	%	제2강세	Alabama	
.	.	음절 경계	A.la.ba.ma	



## 영어, 오스트레일리아(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
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
l	l=	성질 치경 설측 접근음	battle	t
m	m	양순 비음	mouse	p
m̥	m=	성질 양순 비음	anthem	p
n	n	치경 비음	nap	t
n̥	n=	성질 치경 비음	nap	t
ŋ	N	연구개 비음	thing	k
p	p	무성 양순 파열음	pin	p
r	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

Amazon Polly 개발자 안내서  
지원되는 언어의 음소 및 Viseme 표

IPA	X-SAMPA	설명	예	Viseme
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	al	이중 모음	price	a
aʊ	aU	이중 모음	mouth	a
ɑ:	A:	후설 비원순 저장모음	father	a
eɪ	el	이중 모음	face	e
ɜ:	3:	중설 비원순 중저장모음	nurse	E
ɛ	E	전설 비원순 중저모음	dress	E
ɛə	E@	이중 모음	square	E
i:	i	전설 비원순 고장모음	fleece	i
ɪ	l	근전설 비원순 근고모음	kit	i
ɪə	l@	이중 모음	near	i
ɔ:	Oɪ	후설 원순 중저장모음	thought	O
ɔɪ	Oɪ	이중 모음	choice	O
ɒ	Q	후설 원순 저모음	lot	O
u:	u:	후설 원순 고장모음	goose	u
ʊ	U	근후설 원순 근고모음	foot	u
ʊə	U@	이중 모음	cure	u
ʌ	V	후설 평순 중저모음	strut	E
추가 기호				
'	"	제1강세	Alabama	
,	%	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\(hi-IN\) \(p. 42\)](#)를 참조하십시오.

음소/Viseme 표

IPA	X-SAMPA	설명	예	Viseme
자음				
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
l̪	l=	성질 치경 설측 접근음	battle	t
m	m	양순 비음	mouse	p
m̪	m=	성질 양순 비음	anthem	p
n	n	치경 비음	nap	t
n̪	n=	성질 치경 비음	nap	t
ŋ	N	연구개 비음	thing	k
p	p	무성 양순 파열음	pin	p
r	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

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	al	이중 모음	price	a
aʊ	aU	이중 모음	mouth	a
ɑ:	A:	후설 비원순 저장모음	father	a
eɪ	el	이중 모음	face	e
ɜ:	3:	중설 비원순 중저장모음	nurse	E
ɛ	E	전설 비원순 중저모음	dress	E
ɛə	E@	이중 모음	square	E
i:	i	전설 비원순 고장모음	fleece	i
ɪ	l	근전설 비원순 근고모음	kit	i
ɪə	l@	이중 모음	near	i
ɔ:	Oɪ	후설 원순 중저장모음	thought	O
ɔɪ	Oɪ	이중 모음	choice	O
ɒ	Q	후설 원순 저모음	lot	O
u:	u:	후설 원순 고장모음	goose	u
ʊ	U	근후설 원순 근고모음	foot	u
ʊə	U@	이중 모음	cure	u
ʌ	V	후설 평순 중저모음	strut	E
추가 기호				
'	"	제1강세	Alabama	
,	%	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
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
l	l=	성질 치경 설측 접근음	battle	t
m	m	양순 비음	mouse	p
m̥	m=	성질 양순 비음	anthem	p
n	n	치경 비음	nap	t
n̥	n=	성질 치경 비음	button	t
ŋ	N	연구개 비음	thing	k
p	p	무성 양순 파열음	pin	p
r	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

IPA	X-SAMPA	설명	예	Viseme
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	al	이중 모음	price	a
aʊ	aU	이중 모음	mouth	a
ɑ:	A:	후설 비원순 저장모음	father	a
eɪ	el	이중 모음	face	e
ɜ:	3:	중설 비원순 중저장모음	nurse	E
ɛ	E	전설 비원순 중저모음	dress	E
ɛə	E@	이중 모음	square	E
i:	i	전설 비원순 고장모음	fleece	i
ɪ	l	근전설 비원순 근고모음	kit	i
ɪə	l@	이중 모음	near	i
ɔ:	O:	후설 원순 중저장모음	thought	O
ɔɪ	Oɪ	이중 모음	choice	O
ɒ	Q	후설 원순 저모음	lot	O
u:	u:	후설 원순 고장모음	goose	u
ʊ	U	근후설 원순 근고모음	foot	u
ʊə	U@	이중 모음	cure	u
ʌ	V	후설 평순 중저모음	strut	E
추가 기호				
'	"	제1강세	Alabama	
,	%	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
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
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
r	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
ə	@`	중설 치경 접근 수반음	reader	@
æ	{	전설 비원순 근저모음	trap	a
aɪ	al	이중 모음	price	a
aʊ	aU	이중 모음	mouth	a
ɑ	A	후설 비원순 저장모음	father	a
eɪ	el	이중 모음	face	e
ɜ	3`	중설 비원순 중저 치경 접근 수반음	nurse	E
ɛ	E	전설 비원순 중저모음	dress	E
i	i	전설 비원순 고장모음	fleece	i
ɪ	l	근전설 비원순 근고모음	kit	i
oʊ	oU	이중 모음	goat	o
ɔ	O	후설 원순 중저장모음	thought	O
ɔɪ	Oɪ	이중 모음	choice	O
u	u	후설 원순 고장모음	goose	u
ʊ	U	근후설 원순 근고모음	foot	u
ʌ	V	후설 평순 중저모음	strut	E
추가 기호				
'	"	제1강세	Alabama	
,	%	제2강세	Alabama	
.	.	음절 경계	A.la.ba.ma	

## 영어, 웨일스(en-GB-WSL)

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



IPA	X-SAMPA	설명	예	Viseme
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
ɹ	l=	성절 치경 설측 접근음	battle	t
m	m	양순 비음	mouse	p
ɱ	m=	성절 양순 비음	anthem	p
n	n	치경 비음	nap	t
ɳ	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	al	이중 모음	price	a

IPA	X-SAMPA	설명	예	Viseme
au	aU	이중 모음	mouth	a
ɑ:	A:	후설 비원순 저장모음	father	a
eɪ	el	이중 모음	face	e
ɜ:	3:	중설 비원순 중저장모음	nurse	E
ɛ	E	전설 비원순 중저모음	dress	E
ɛə	E@	이중 모음	square	E
i:	i	전설 비원순 고장모음	fleece	i
ɪ	ɪ	근전설 비원순 근고모음	kit	i
ɪə	ɪ@	이중 모음	near	i
ɔ:	Oɪ	후설 원순 중저장모음	thought	O
ɔɪ	Oɪ	이중 모음	choice	O
ɒ	Q	후설 원순 저모음	lot	O
u:	u:	후설 원순 고장모음	goose	u
ʊ	U	근후설 원순 근고모음	foot	u
ʊə	U@	이중 모음	cure	u
ʌ	V	후설 평순 중저모음	strut	E
추가 기호				
'	"	제1강세	Alabama	
,	%	제2강세	Alabama	
.	.	음절 경계	A.la.ba.ma	

## 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

IPA	X-SAMPA	설명	예	Viseme
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
ɸ	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
ã	A~	후설 비원순 비(nasal) 저모음	camembert	a
e	e	전설 비원순 중고모음	marché	e
ɛ	E	전설 비원순 중저모음	neige	E
ẽ	E~	전설 비원순 비(nasal) 중저모음	sapin	E
i	i	전설 비원순 고모음	mille	i

IPA	X-SAMPA	설명	예	Viseme
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				
'	"	제1강세	Alabama	
,	%	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
ŋ	N	연구개 비음	parking	k
p	p	무성 양순 파열음	pomme	p
ʁ	R	유성 구개수 마찰음	amoureux	k

IPA	X-SAMPA	설명	예	Viseme
s	s	무성 치경 마찰음	sant�	s
ʃ	S	무성 후치경 마찰음	chat	S
t	t	무성 치경 파열음	t�l�phone	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				
�	2	전설 원순 중고모음	deux	o
�	9	전설 원순 중저모음	neuf	O
�	9~	전설 원순 비(nasal) 중저모음	brun	O
�	@	중설 중모음	je	@
a	a	전설 비원순 저모음	table	a
�	A~	후설 비원순 비(nasal) 저모음	camembert	a
e	e	전설 비원순 중고모음	march�	e
�	E	전설 비원순 중저모음	neige	E
�	E~	전설 비원순 비(nasal) 중저모음	sapin	E
i	i	전설 비원순 고모음	mille	i
o	o	후설 원순 중고모음	h�pital	o
ɔ	O	후설 원순 중저모음	homme	O
�	O~	후설 원순 비(nasal) 중저모음	bon	O
u	u	후설 원순 고모음	sous	u
y	y	전설 원순 고모음	dur	u
추가 기호				
'	"	제1강세	Alabama	
,	%	제2강세	Alabama	
.	.	음절 경계	A.la.ba.ma	

## 독일어(de-DE)

다음 표에는 Amazon Polly에서 지원하는 독일어 음소에 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	ʔ	성문 폐쇄음		
b	b	유성 양순 파열음	Bier	p
d	d	유성 치경 파열음	Dach	t
ç	C	무성 경구개 마찰음	ich	k
ɔ̯	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	
r	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

IPA	X-SAMPA	설명	예	Viseme
모음				
ø:	2:	전설 원순 중고장모음	böse	o
e	6	중설 근저모음	besser	a
ɐ	6_^	비성절 중설 근저모음	Klar	a
œ	9	전설 원순 중저모음	können	O
ə	@	중설 중모음	Rede	@
a	a	전설 비원순 저모음	Salz	a
a:	a:	전설 비원순 저장모음	Sahne	a
aɪ	aɪ	이중 모음	nein	a
au	aU	이중 모음	Augen	a
ã	A~	후설 비원순 비(nasal) 저모음	Restaurant	a
e:	e:	전설 비원순 중고장모음	Rede	e
ɛ	E	전설 비원순 중저모음	Keller	E
ẽ	E~	전설 비원순 비(nasal) 중저모음	Terrain	E
i:	i:	전설 비원순 고장모음	Lied	i
ɪ	ɪ	근전설 비원순 근고모음	bitte	i
o:	o:	후설 원순 중고장모음	Kohl	o
ɔ	O	후설 원순 중저모음	Koffer	O
õ	O~	후설 원순 비(nasal) 중저모음	Annonce	O
ɔʏ	OY	이중 모음	neu	O
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				
'	"	제1강세	Alabama	
,	%	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\(en-IN\) \(p. 29\)](#)를 참조하십시오.

음소/Viseme 표

IPA	X-SAMPA	설명	예
자음			
p <sup>h</sup>	p_h	무성 기식 양순 파열음	फूल(phool)
b <sup>h</sup>	b_h	유성 기식 양순 파열음	भारी(bhaari)
t̪	t_d	무성 치 파열음	तापमान(taapmaan)
t̪ <sup>h</sup>	t_d_h	무성 기식 치 파열음	थोड़ा(thoda)
d̪	d_d	유성 치 파열음	दिल्ली(dilli)
d̪ <sup>h</sup>	d_d_h	유성 기식 치 파열음	धोबी(dhobi)
t̪	t̊	무성 권설 파열음	कटोरा(katora)
t̪ <sup>h</sup>	t̊_h	무성 기식 권설 파열음	ठंड(thand)
d̪	d̊	유성 권설 파열음	डर(darr)
d̪ <sup>h</sup>	d̊_h	유성 기식 권설 파열음	ढाल(dhal)
tʃ <sup>h</sup>	tS_h	무성 기식 구개 파찰음	छाल(chaal)
dʒ <sup>h</sup>	dZ_h	유성 기식 구개 파찰음	झाल(jhaal)
k <sup>h</sup>	k_h	무성 기식 연구개 파열음	खान(khan)
g <sup>h</sup>	g_h	유성 기식 연구개 파열음	घान(ghaan)
ɳ	n̊	권설 비음	क्षण(kshan)
r	ɽ	치경 탄음	राम(ram)
ɽ	r̊	평권설 단전음	बड़ा(bada)
ɽ <sup>h</sup>	r̊_h	유성 기식 권설 단전음	बढ़ी(barhi)
ʋ	v\	양순 접근음	वसूल(wasool)
모음			
ə	@_o	중설 중모음	अच्छा(achhaa)
ẽ	@~	중설 비음화 중모음	हंसना(hansnaa)
a	A_o	전설 비원순 저모음	आग(aag)
ã	A~	전설 비원순 비음화 저모음	घड़ियाँ(ghariyaan)
ɪ	I_o	근전설 비원순 근고모음	इक्कीस(ikkees)



IPA	X-SAMPA	설명	예
ĩ	I~	근전설 비원순 비음화 근고모음	संचिई(sinchai)
i	i_o	전설 비원순 고모음	बिल्ली(billee)
ĩ	i~	전설 비원순 비음화 고모음	नही(nahin)
ʊ	U_o	근후설 원순 근고모음	उलूल(ullu)
ũ	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)
õ	O~	후설 원순 비음화 중저모음	भौ(bhaun)
o	o	후설 원순 중고모음	सोना(sona)
õ	o~	후설 원순 비음화 중고모음	क्यो(kyon)
ɛ	E_o	전설 비원순 중저모음	पैसा(paisa)
ẽ	E~	전설 비원순 비음화 중저모음	मैं(main)
e	e	전설 비원순 중고모음	एक(ek)
ẽ	e~	전설 비원순 비음화 중고모음	कतिबें (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

IPA	X-SAMPA	설명	예	Viseme
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
k <sup>h</sup>	k_h	기식 무성 연구개 파열음	ósköpunum	k
l	l	치경 설측 접근음	gólf	t
ɭ	l_0	무성 치경 설측 접근음	fólk	t
m	m	양순 비음	september	p
m̥	m_0	무성 양순 비음	kompá	p
n	n	치경 비음	númer	t
ɳ	n_0	무성 치경 비음	pöntun	t
ɲ	J	경구개 비음	pælingar	J
ŋ	N	연구개 비음	söngvarann	k
ɳ̥	N_0	무성 연구개 비음	frænka	k
p <sup>h</sup>	p_h	기식 무성 양순 파열음	afplánun	p
r	r	치경 전동음	afskrifta	r
ɽ	r_0	무성 치경 전동음	andvörpum	r
s	s	무성 치경 마찰음	baðhús	s
t <sup>h</sup>	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
œ:	9:	전설 원순 중저장모음	tvö	O
a	a	전설 비원순 저모음	nefna	a
a:	a:	전설 비원순 저장모음	fara	a

IPA	X-SAMPA	설명	예	Viseme
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	Gúliver	i
i:	i:	전설 비원순 고장모음	þír	i
ɪ	ɪ	근전설 비원순 근고모음	samspil	i
ɪ:	ɪ:	근전설 비원순 근고장모음	stig	i
ɔ	O	후설 원순 중저모음	regndropar	O
ɔ:	O:	후설 원순 중저장모음	ullarbolur	O
ou	Ou	이중 모음	tólf	O
ou:	Ou:	이중 모음	fjórir	O
u	u	후설 원순 고모음	stúlkan	u
u:	u:	후설 원순 고장모음	frú	u
ʏ	Y	근전설 원순 근고모음	tíu	u
ʏ:	Y	근전설 원순 근고장모음	gruninn	u
추가 기호				
'	"	제1강세	Alabama	
,	%	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

IPA	X-SAMPA	설명	예	Viseme
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
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				
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

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

## 일본어(ja-JP)

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				
r	4	치경 탄음	練習, renshuu	t
ʔ	ʔ	성문 폐쇄음	あつつ, atsu'	
b	b	유성 양순 파열음	舞踊, buyou	p
β	B	유성 양순 마찰음	ヴインテージ, vinteeji	B
c	c	무성 경구개 파열음	ききょう, kikyuu	k
ç	C	무성 경구개 마찰음	人, hito	k
d	d	유성 치경 파열음	濁点, dakuten	t
dz	dz\	유성 치경구개 파찰음	純, jun	J
g	g	유성 연구개 파열음	ご飯, gohan	k
h	h	무성 성문 마찰음	本, hon	k
j	j	경구개 접근음	屋根, yane	i
ʃ	J\	유성 경구개 파열음	行儀, gyougi	J
k	k	무성 연구개 파열음	漢字, kanji	k
ʃ	ʃ\	치경 설측 탄음	釣り, tsuri	r
ʃj	ʃj	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
ɲ	N\	구개수 비음	缶, kan	k

IPA	X-SAMPA	설명	예	Viseme
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
ts	ts	무성 치경 파찰음	釣り, tsuri	s
tɕ	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
모음				
ä:	a:_"	중설 비원순 저장모음	羽蟻, haari	a
ä	a_"	중설 비원순 저모음	仮名, kana	a
e:	e:_o	전설 비원순 중장모음	学生, gakusei	@
e	e_o	전설 비원순 중모음	歴, reki	@
i	i	전설 비원순 고모음	気, ki	i
i:	i:	전설 비원순 고장모음	詩歌, shiika	i
u	M USD	후설 비원순 고모음	運, un	i
u:	M USD:	후설 비원순 고장모음	宗教, shuukyoku	i
o:	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
kʰ	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

IPA	X-SAMPA	설명	예	Viseme
tʰ	t_t	무성 치경 경파열음	때, [tt]e	t
r	4	치경 탄음	사랑, sa[r]ang	t
l	l	치경 설측 접근음	둘, do[l]	t
m	m	양순 비음	무, [m]u	p
p	p	무성 양순 파열음	봄, [b]om	p
pʰ	p_t	무성 양순 경파열음	빨, [pp]eol	p
s	s	무성 치경 마찰음	새, [s]e	s
sʰ	s_t	무성 치경 경마찰음	씨, [ss]i	s
ŋ	N	연구개 비음	방, ba[ŋ]	k
t͡ɕ	ts\	무성 치경구개 파찰음	조, [j]o	J
t͡ɕʰ	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
pʰ	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
ɯ	M\	연구개 접근음>	의, [w]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 노르웨이어 음성에 대해 Amazon Polly에서 지원하는 IPA(International Phonetic Alphabet) 음소와 X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet) 기호 세트 및 해당하는 viseme 이 나와 있습니다.

IPA	X-SAMPA	설명	예	Viseme
자음				
r	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
t	t	무성 치경 파열음	lat	t
ʈ	t`	무성 권설 파열음	hardt	t
u	v\	순치 접근음	vin	f
w	w	양순 연구개 접근음	will	x
모음				
ø:	2:	전설 원순 중고장모음	søt	o



IPA	X-SAMPA	설명	예	Viseme
œ	9	전설 원순 중저모음	søtt	O
ə	@	중설 중모음	ape	@
æ:	{:	전설 비원순 근저장모음	vær	a
ʊ	}	중설 원순 고모음	lund	u
ʊ:	}::	중설 원순 고장모음	lun	u
æ	{	전설 비원순 근저모음	vært	a
ɑ	A	후설 비원순 저모음	hatt	a
ɑ:	A:	후설 비원순 저장모음	hat	a
e:	e:	전설 비원순 중고장모음	sen	e
ɛ	E	전설 비원순 중저모음	send	E
i:	i:	전설 비원순 고장모음	vin	i
ɪ	ɪ	근전설 비원순 근고모음	vind	i
o:	o:	후설 원순 중고장모음	våt	o
ɔ	O	후설 원순 중저모음	vått	O
u:	u:	후설 원순 고장모음	bok	u
ʊ	U	근후설 원순 근고모음	bukk	u
y:	y:	전설 원순 고장모음	lyn	u
ʏ	Y	근전설 원순 근고모음	lynne	u
추가 기호				
'	"	제1강세	Alabama	
,	%	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				

Amazon Polly 개발자 안내서  
지원되는 언어의 음소 및 Viseme 표

IPA	X-SAMPA	설명	예	Viseme
b	b	유성 양순 파열음	bobas, belka	p
d	d	유성 치경 파열음	dar, do	t
dz	dz	유성 치경 파찰음	dzwon, widzowie	s
ɖ	ɖ	유성 치경구개 파찰음	dźwięk	J
ɗ	ɗ	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
ʃ	s\	무성 치경구개 마찰음	śruba, śnieg	J
ʂ	s`	무성 권설 마찰음	szum, masz	S
t	t	무성 치경 파열음	tok, stół	t
ts	ts	무성 치경 파찰음	car, co	s
ɖ͡ʂ	ts\	무성 치경구개 파찰음	ćma, mieć	J
ɖ͡ʂ̺͡	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
ʐ	z\	유성 치경구개 마찰음	żrebię, bieliźnie	J
ʑ	z`	유성 권설 마찰음	żar, żona	S
모음				
a	a	전설 비원순 저모음	ja	a
ɛ	E	전설 비원순 중저모음	echo	E

IPA	X-SAMPA	설명	예	Viseme
ɛ	E~	전설 비원순 비(nasal) 중저모음	węże	E
i	i	전설 비원순 고모음	ile	i
ɔ	O	후설 원순 중저모음	oczy	O
ɔ̃	O~	후설 원순 비(nasal) 중저모음	wąż	O
u	u	후설 원순 고모음	uczta	u
ɪ	1	중설 비원순 고모음	byk	i
추가 기호				
'	"	제1강세	Alabama	
,	%	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				
r	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
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

IPA	X-SAMPA	설명	예	Viseme
<b>R</b>	R\	구개수 전동음	barroso	k
<b>s</b>	s	무성 치경 마찰음	saca	s
<b>ʃ</b>	S	무성 후치경 마찰음	chato	S
<b>t</b>	t	무성 치경 파열음	tacto	t
<b>v</b>	v	유성 순치 마찰음	vaca	f
<b>w</b>	w	양순 연구개 접근음	mau	u
<b>z</b>	z	유성 치경 마찰음	zaca	s
<b>ʒ</b>	Z	유성 후치경 마찰음	jacto	S
모음				
<b>a</b>	a	전설 비원순 저모음	parto	a
<b>ã</b>	a~	전설 비원순 비(nasal) 저모음	pega	a
<b>e</b>	e	전설 비원순 중고모음	pega	e
<b>ẽ</b>	e~	전설 비원순 비(nasal) 중고모음	movem	e
<b>ɛ</b>	E	전설 비원순 중저모음	café	E
<b>i</b>	i	전설 비원순 고모음	lingueta	i
<b>ĩ</b>	i~	전설 비원순 비(nasal) 고모음	cinto	i
<b>o</b>	o	후설 원순 중고모음	poder	o
<b>õ</b>	o~	후설 원순 비(nasal) 중고모음	compra	o
<b>ɔ</b>	O	후설 원순 중저모음	cotó	O
<b>u</b>	u	후설 원순 고모음	fui	u
<b>ũ</b>	u~	후설 원순 비(nasal) 고모음	sunto	u
추가 기호				
'	"	제1강세	Alabama	
,	%	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				
r	4	치경 탄음	pira	t
b	b	유성 양순 파열음	bato	p
d	d	유성 치경 파열음	dato	t
ɗ	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
ʈ	tS	무성 후치경 파찰음	noite	S
v	v	유성 순치 마찰음	vaca	f
w	w	양순 연구개 접근음	mau	u
x	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

IPA	X-SAMPA	설명	예	Viseme
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
ũ	u~	후설 원순 비(nasal) 고모음	sunto	u
추가 기호				
'	"	제1강세	Alabama	
,	%	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
dʒ	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

IPA	X-SAMPA	설명	예	Viseme
j	j	경구개 접근음	baie	i
k	k	무성 연구개 파열음	co#	k
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
ɛ	e_^	비성질 전설 비원순 중고모음	beau	e
i	i	전설 비원순 고모음	mie	i
o	o	후설 원순 중고모음	oră	o
oa	o_^a	이중 모음	oare	o
u	u	후설 원순 고모음	unde	u
ɨ	1	중설 비원순 고모음	România	i
추가 기호				
'	"	제1강세	Alabama	
,	%	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	bʲ	경구개음화 유성 양순 파열음	бюро	p
d	d	유성 치경 파열음	дом	t
dʲ	dʲ	경구개음화 유성 치경 파열음	дядя	t
f	f	무성 순치 마찰음	флаг	f
fʲ	fʲ	경구개음화 무성 순치 마찰음	февраль	f
g	g	유성 연구개 파열음	нога	k
gʲ	gʲ	경구개음화 유성 연구개 파열음	герой	k
j	j	경구개 접근음	дизайн, ящик	i
k	k	무성 연구개 파열음	кот	k
kʲ	kʲ	경구개음화 무성 연구개 파열음	кино	k
l	l	치경 설측 접근음	лампа	t
lʲ	lʲ	경구개음화 치경 설측 접근음	лес	t
m	m	양순 비음	мама	p
mʲ	mʲ	경구개음화 양순 비음	мяч	p
n	n	치경 비음	нос	t
nʲ	nʲ	경구개음화 치경 비음	няня	t
p	p	무성 양순 파열음	папа	p
pʲ	pʲ	경구개음화 무성 양순 파열음	перо	p
r	r	치경 전동음	роза	r
rʲ	rʲ	경구개음화 치경 전동음	рюмка	r



IPA	X-SAMPA	설명	예	Viseme
s	s	무성 치경 마찰음	сыр	s
sʲ	sʲ	경구개음화 무성 치경 마찰음	сердце, русь	s
ɕ:	s\:	무성 치경구개 마찰음	щека	J
ɬ	sʰ	무성 권설 마찰음	шум	S
t	t	무성 치경 파열음	точка	t
tʲ	tʲ	경구개음화 무성 치경 파열음	тётя	t
ts	ts	무성 치경 파찰음	царь	s
tɕ	ts\	무성 치경구개 파찰음	час	J
v	v	유성 순치 마찰음	вор	f
vʲ	vʲ	경구개음화 유성 순치 마찰음	верфь	f
x	x	무성 연구개 마찰음	хор	k
xʲ	xʲ	경구개음화 무성 연구개 마찰음	химия	k
z	z	유성 치경 마찰음	зуб	s
zʲ	zʲ	경구개음화 유성 치경 마찰음	зима	s
ʐ:	z\:	유성 치경구개 마찰음	уезжать	J
ʐ	zʰ	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				
r	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
ɟ	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
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
ʈ	tS	무성 후치경 파찰음	chubasco	S
θ	T	무성 치 마찰음	cereza, zorro, lacero, paz	T

IPA	X-SAMPA	설명	예	Viseme
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
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	ɪ	근전설 비원순 근고모음	keçi	i
ʊ	M USD	후설 비원순 고모음	kıl	i
o	o	후설 원순 중고장모음	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				
'	"	제1강세	Alabama	
,	%	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				
r	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
ɟ	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

IPA	X-SAMPA	설명	예	Viseme
x	x	무성 연구개 마찰음	jamón, general, suje, reloj	k
z	z	유성 치경 마찰음	rasgo, mismo	s
h	h	무성 성문 마찰음	Harrison	k
ɹ	ɹ	후치경 접근음	Brian	r
v	v	유성 순치 마찰음	뱅크버	f
모음				
a	a	중양의 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
ɛ	E	전설 비원순 중저모음	dress	E
ɔ	O	후설 원순 중저모음	포트	O
ə	@	중설 중모음	로더데일	@
추가 기호				
'	"	제1강세	Alabama	
,	%	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.

IPA	X-SAMPA	설명	예	Viseme
자음				
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

IPA	X-SAMPA	설명	예	Viseme
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	ɹ\	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	@
ə̃	@`	중설 치경 접근 수반음	reader	@
æ	{	전설 비원순 근저모음	trap	a
aɪ	al	이중 모음	price	a
aʊ	aU	이중 모음	mouth	a
ɑ	A	후설 비원순 저장모음	father	a
eɪ	el	이중 모음	face	e
ɝ̃	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	후설 평순 중저모음	strut	E
추가 기호				
'	"	제1강세	Alabama	
,	%	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

IPA	X-SAMPA	설명	예	Viseme
ɳ	n`	권설 비음	barn	t
ŋ	N	연구개 비음	ring	k
p	p	무성 양순 파열음	pil	p
r	r	치경 전동음	ris	r
s	s	무성 치경 마찰음	sil	s
ɕ	s\	무성 치경구개 마찰음	tjock	J
ʃ	s`	무성 권설 마찰음	fors, schlager	S
t	t	무성 치경 파열음	tal	t
ʈ	t`	무성 권설 파열음	hjord	t
v	v	유성 순치 마찰음	vår	f
w	w	양순 연구개 접근음	aula, airways	u
ɸ	x\	무성 경구개-연구개 마찰음	sjuk	k
모음				
ø	2	전설 원순 중고모음	föll, förr	o
œ	2:	전설 원순 중고장모음	föl, nöt, för	o
ə	8	중설 원순 중고모음	buss, full	o
ə	@	중설 중모음	pojken	@
ʊ	}:	중설 원순 고장모음	hus, ful	u
a	a	전설 비원순 저모음	hall, matt	a
æ	{	전설 비원순 근저모음	herr	a
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, hæl, här	E:
i:	i:	전설 비원순 고장모음	vit, sil	i:
ɪ	l	근전설 비원순 근고모음	vitt, sill	i
o:	o:	후설 원순 중고장모음	håll, mål	o
ɔ	O	후설 원순 중저모음	håll, moll	O
u:	u:	후설 원순 고장모음	sol, bot	u



IPA	X-SAMPA	설명	예	Viseme
ʊ	U	근후설 원순 근고모음	bott	u
y	y	전설 원순 고모음	bytt	u
y:	y:	전설 원순 고장모음	syl, syl	u
추가 기호				
'	"	제1강세	Alabama	
,	%	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				
r	4	치경 탄음	durum	t
ɾ	4_0_r	무성 마찰 치경 탄음	bir	t
ɽ	4_r	마찰 치경 탄음	raf	t
b	b	유성 양순 파열음	raf	p
c	c	무성 경구개 파열음	kedı	k
d	d	유성 치경 파열음	dede	t
ɖʒ	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
ʝ	J\	유성 경구개 파열음	genç	J
k	k	무성 연구개 파열음	akıl	k
l	l	치경 설측 접근음	lale	t
ɭ	5	연구개음화 치경 설측 접근음	labirent	t
m	m	양순 비음	maaş	p

IPA	X-SAMPA	설명	예	Viseme
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
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	l	근전설 비원순 근고모음	keçi	i
ʊ	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
ɣ	Y	근전설 원순 근고모음	aşı	u
추가 기호				

IPA	X-SAMPA	설명	예	Viseme
'	"	제1강세	Alabama	
,	%	제2강세	Alabama	
.	.	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
ɖʒ	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
ɱ	m_0	무성 양순 비음	ymhen	p
n	n	치경 비음	naw	t
ɳ	n_0	무성 치경 비음	anhawster	t
ŋ	N	연구개 비음	argyfwng	k
ɲ	N_0	무성 연구개 비음	anghenion	k
p	p	무성 양순 파열음	pump	p
r	r	치경 전동음	rhoi	r
ɾ	r_0	무성 치경 전동음	garw	r

IPA	X-SAMPA	설명	예	Viseme
s	s	무성 치경 마찰음	saith	s
ʃ	S	무성 후치경 마찰음	siawns	S
t	t	무성 치경 파열음	tegan	t
tʃ	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
x	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	@
a	a	전설 비원순 저모음	acen	a
ai	ai	이중 모음	dau	a
au	au	이중 모음	awdur	a
ɑ:	A:	후설 비원순 저장모음	mab	a
ɑ:ɪ	A:1	이중 모음	aelod	a
e:	e:	전설 비원순 중고장모음	peth	e
ɛ	E	전설 비원순 중저모음	pedwar	E
ɛi	Ei	이중 모음	beic	E
i:	i:	전설 비원순 고장모음	tri	i
ɪ	I	근전설 비원순 근고모음	miliwn	i
iʊ	1u	이중 모음	unigryw	i
o:	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
u:	u:	후설 원순 고장모음	cwch	u
ʊ	U	근후설 원순 근고모음	acwstig	u
ʊi	Ui	이중 모음	wyth	u

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

## 신경 TTS

Amazon Polly에는 표준 음성보다 자연스러운 음성을 낼 수 있는 NTTS(신경 TTS) 시스템이 있습니다. NTTS 시스템은 최대한 자연스럽게 사람과 유사한 텍스트 투 스피치 음성을 생성합니다.

표준 TTS 음성은 연결 합성을 사용합니다. 이 방법은 녹음된 스피치의 음소를 서로 연결하여 매우 자연스럽게 들리도록 합성된 스피치를 만들어 냅니다. 하지만 말은 변형될 수밖에 없고 파형 분절에 사용되는 기술로 인해 스피치 품질이 제한됩니다.

Amazon Polly 신경 TTS 시스템은 스피치를 생성하기 위해 표준 연결 합성을 사용하지 않으며, 다음 두 부분으로 이루어져 있습니다.

- 연속된 음소(언어의 가장 기본적인 단위)를 연속된 스펙트로그램으로 변환하는 신경망. 스펙트로그램은 다양한 주파수 대역의 에너지 레벨을 나타내는 스냅샷입니다.
- 스펙트로그램을 연속된 오디오 신호로 변환하는 보코더

신경 TTS 시스템의 첫 번째 구성 요소는 Sequence-to-Sequence 모델입니다. 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 음성 파형으로 변환됩니다. 범용 연결 합성 시스템을 빌드하기 위해 사용되는 대규모 데이터 세트에서 훈련될 경우 이 Sequence-to-Sequence 접근 방식은 더 우수하고 자연스럽게 들리는 음성을 만들어 냅니다.

주제

- [기능 및 리전 호환성](#) (p. 72)
- [음성 엔진](#) (p. 73)
- [신경 음성](#) (p. 75)
- [NTTS 말투](#) (p. 76)

## 기능 및 리전 호환성

모든 AWS 리전에서 신경 음성을 사용할 수 있는 것은 아니며 신경 음성이 Amazon Polly 기능을 모두 지원하는 것도 아닙니다.

신경 음성이 지원되는 리전은 다음과 같습니다.

- 미국 동부(버지니아 북부): us-east-1
- 미국 서부(오레곤): us-west-2
- EU(아일랜드): eu-west-1
- 아시아 태평양(시드니): ap-southeast-2

이 리전의 엔드포인트와 프로토콜은 표준 음성에서 사용되는 것과 같습니다. 자세한 내용은 [리전 및 엔드포인트](#)를 참조하십시오.

신경 음성에서 지원되는 기능은 다음과 같습니다.

- 실시간 및 비동기 스피치 합성 작업
- 뉴스 진행자 및 대화 말투 (말투에 대한 자세한 내용은 [NTTS 말투 \(p. 76\)](#) 단원을 참조하십시오.)
- 모든 스피치 마크
- Amazon Polly에서 지원하는 상당수의(전부 아님) SSML 태그(NTTS 지원 SSML 태그에 대한 자세한 내용은 [지원되는 SSML 태그 \(p. 91\)](#) 단원 참조)

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.

## 음성 엔진

Amazon Polly에서는 신경 또는 표준 음성을 engine 속성과 함께 사용할 수 있습니다. 가능한 값은 표준 또는 Neural(신경)이며 표준 음성을 사용할지 신경 음성을 사용할지를 나타냅니다. 표준이 기본값입니다.

### 음성 엔진 선택(콘솔)

음성 엔진을 선택하려면(콘솔)

1. <https://console.aws.amazon.com/polly/>에서 Amazon Polly 콘솔을 엽니다.
2. 텍스트 투 스피치 페이지에서 엔진에 대해 표준 또는 Neural(신경)을 선택합니다.

The screenshot shows the AWS Management Console for the us-east-1 region. The left sidebar shows the navigation menu with 'Amazon Polly' selected, and 'Text-to-Speech' is the active sub-section. The main content area is titled 'Text-to-Speech' and includes instructions on how to use the service. Below the instructions, there are two tabs: 'Plain text' (selected) and 'SSML'. A text input field contains the sample text 'Hi! My name is Joanna.' and a character count shows '58 characters used'. At the bottom, the 'Engine' section is highlighted with a red box, showing two radio button options: 'Standard' and 'Neural'. The 'Neural' option is selected. Below the engine selection, the 'Language and Region' section is partially visible.

aws Services Resource Groups

Amazon Polly

- Text-to-Speech
- Lexicons
- S3 synthesis tasks

## Text-to-Speech

Listen, customize, and

Type or paste your text in the  
integrate it into your applicati

With up to 3000 characters y  
saved to an S3 bucket.

Plain text SSML

Hi! My name is Joanna.

58 characters used

**Engine** ⓘ

- ☐ Standard
- ☒ Neural

**Language and Region**



Neural(신경)을 선택하면 신경 음성만 사용할 수 있으며 표준 전용 음성은 비활성화됩니다.

## 음성 엔진 선택(CLI)

음성 엔진을 선택하려면(CLI)

engine 파라미터는 선택 사항이며 가능한 값은 standard 또는 Neural입니다. SynthesisTask 작업을 만들 때 이 속성을 사용하십시오.

예를 들어 다음 코드를 사용하여 US West-2(오레곤) 리전에서 start-speech-synthesis-task AWS CLI 명령을 실행할 수 있습니다.

다음은 Unix, Linux, macOS용 형식으로 지정된 AWS CLI 예제입니다. 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 \
  --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
```

이렇게 하면 다음과 비슷한 모양의 응답이 생성됩니다.

```
"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"
}
```

## 신경 음성

신경 음성은 미국 영어(en-US), 영국 영어(en-GB), 브라질 포르투갈어(pt-BR) 및 미국 스페인어(es-US)로 제공됩니다. 다음 표에 음성이 나와 있습니다.

언어	이름/ID	성별
영어(영국)(en-GB)	Amy	여성
	Emma	여성
	Brian	남성

언어	이름/ID	성별
영어(미국)(en-US)	Ivy	여성(어린이)
	Joanna	여성
	Kendra	여성
	Kimberly	여성
	Salli	여성
	Joey	남성
	Justin	남성(어린이)
	Matthew	남성
포르투갈어(브라질)(pt-BR)	Camila	여성
스페인어(미국)(es-US)	Lupe	여성

대담한 음성은 뉴스 진행자 및 대화 말투에 사용할 수 있습니다. 자세한 내용은 [NTTS 말투 \(p. 76\)](#) 단원을 참조하십시오.

## NTTS 말투

사람의 말투는 상황에 따라 달라집니다. 가령 편안한 대화는 TV나 라디오 뉴스와 전혀 다르게 들립니다. Amazon Polly에서 표준 음성을 사용하여 스피치를 합성할 때 연결 방법이 사용됩니다. 연결 방법은 오디오 데이터베이스에 저장된 짧은 스피치 조각들을 연결해 가장 자연스럽게 들리는 최적의 스피치를 만들어 냅니다. 그러나 이러한 음성이 만들어지는 방식으로 인해 다른 말투를 만들어 낼 수 없습니다.

표준 연결 합성 이외에, Amazon Polly은 스피치 생성을 위해 신경 기술을 사용할 수 있습니다. Amazon Polly은 Sequence-to-Sequence 모델을 사용하여 신경 음성을 생성합니다. 이 모델은 오디오 데이터 입력을 사용하여 음성을 형성하는 결과를 생성하고 출력 시퀀스에서의 위치도 고려합니다. 그리고 나서 매우 자연스러운 음성으로 있는 그대로 사용할 수도 있고 특정 말투를 훈련하고 변형하며 이 말투에서 비롯되는 스피치의 일부분을 강조할 수도 있습니다.

Amazon Polly은 사용할 수 있는 두 가지 말투인 뉴스 진행자 말투와 대화 말투를 제공합니다.

뉴스 진행자 스타일은 신경 시스템을 사용하여 TV 또는 라디오 뉴스 진행자 스타일로 스피치를 생성합니다. 뉴스 진행자 스타일은 미국 영어(en-US) 전용의 Matthew 및 Joanna 음성으로만 제공됩니다.

대화 스타일은 신경 시스템을 사용하여 많은 사용 사례에서 사용할 수 있는 보다 친근하고 표현적인 대화 스타일로 스피치를 생성합니다. 대화 스타일은 미국 영어(en-US) 전용의 Matthew 및 Joanna 음성으로만 제공됩니다.

주제

- [뉴스 진행자 스타일 사용 \(p. 76\)](#)
- [대화 스타일 사용 \(p. 77\)](#)

## 뉴스 진행자 스타일 사용

뉴스 진행자 스타일은 신경 엔진을 사용하여 Matthew 및 Joanna 음성으로만 제공됩니다. 이 스타일은 미국 영어(en-US)로만 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 리전 호환성 \(p. 72\)](#) 단원을 참조하십시오.

뉴스 진행자 스타일을 적용하려면(콘솔)

1. <https://console.aws.amazon.com/polly/>에서 Amazon Polly 콘솔을 엽니다.
2. 신경 음성이 지원되는 AWS 리전을 사용하고 있는지 확인합니다.
3. 텍스트 투 스피치 페이지에서 엔진에 대해 Neural(신경)을 선택합니다.
4. [SSML] 탭을 선택합니다.
5. 뉴스 진행자 스타일 SSML 구문을 사용하여 텍스트 투 스피치 요청에 입력 텍스트를 추가합니다.

뉴스 진행자 스타일을 적용하려면(CLI)

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

```
--engine neural
```

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

입력 파일에서 뉴스 진행자 스타일 SSML 구문을 사용합니다.

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

예를 들어 다음과 같이 Matthew 또는 Joanna 음성으로 뉴스 진행자 태그를 사용할 수 있습니다.

```
<speack>
<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>
</speack>
```

SSML에 대한 자세한 내용은 [지원되는 SSML 태그 \(p. 91\)](#) 단원을 참조하십시오.

## 대화 스타일 사용

대화 스타일은 신경 엔진을 사용하여 Matthew 및 Joanna 음성으로만 제공됩니다. 이 스타일은 미국 영어(en-US)로만 제공됩니다. 대화 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 리전 호환성 \(p. 72\)](#) 단원을 참조하십시오.

### 대화 스타일을 적용하려면(콘솔)

1. <https://console.aws.amazon.com/polly/>에서 Amazon Polly 콘솔을 엽니다.
2. 신경 음성이 지원되는 AWS 리전을 사용하고 있는지 확인합니다.
3. 텍스트 투 스피치 페이지에서 엔진에 대해 Neural(신경)을 선택합니다.
4. [SSML] 탭을 선택합니다.
5. 대화 스타일 SSML 구문을 사용하여 텍스트 투 스피치 요청에 입력 텍스트를 추가합니다.

### 대화 스타일을 적용하려면(CLI)

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

```
--engine neural
```

2. 대화 스타일 SSML 구문을 사용하여 API 요청에 입력 텍스트를 추가합니다.

입력 파일에서 다음 대화 스타일 SSML을 사용합니다.

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

예를 들어 H.G의 다음 인용문에서 Matthew 또는 Joanna 음성과 함께 대화 태그를 사용할 수 있습니다. Wells의 소설 우주 전쟁(War of the Worlds):

```
<speak>  
<amazon:domain name="conversational">  
No one would have believed in the last years of the nineteenth century that this world was  
being watched  
keenly and closely by intelligences greater than man's and yet as mortal as his own; that  
as men busied  
themselves about their various concerns they were scrutinised and studied perhaps almost as  
narrowly as a  
man with a microscope might scrutinise the transient creatures that swarm and multiply in a  
drop of water.  
</amazon:domain>  
</speak>
```

SSML에 대한 자세한 내용은 [지원되는 SSML 태그 \(p. 91\)](#) 단원을 참조하십시오.

# 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.

신경 또는 표준 텍스트 두 스피치 형식을 사용할 때 스피치 마크를 사용할 수 있습니다.

주제

- [스피치 마크 형식](#) (p. 79)
- [스피치 마크 사용](#) (p. 80)
- [스피치 마크 요청\(콘솔\)](#) (p. 83)

## 스피치 마크 형식

`SynthesizeSpeech` 또는 `StartSpeechSynthesisTask` 명령에 `SpeechMarkTypes` 옵션을 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 들어, AWS CLI에서는 다음과 같습니다.

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

Amazon Polly에서는 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 [Viseme 및 Amazon Polly](#) (p. 79)를 참조하십시오.
- `ssml` – SSML 입력 텍스트의 `<mark>` 요소를 설명합니다. 자세한 내용은 [SSML 문서로부터 스피치 생성](#) (p. 84)를 참조하십시오.

## Viseme 및 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 viseme 집합이 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 표 \(p. 18\)](#)을 참조하십시오.

IPA	X-SAMPA	설명	예	Viseme
자음				
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
...	...	...	...	...

## 스피치 마크 사용

### 스피치 마크 요청

입력 텍스트의 스피치 마크를 요청하려면 `synthesize-speech` 명령을 사용하십시오. 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 지정하십시오. 사용 가능한 음성은 동일한 스피치 속도를 아키텍처. 스피치를 생성하는 데 사용된 것과 다른 음성을 사용하는 경우 메타데이터가 오디오 스트림과 일치하지 않습니다.

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

- `speech-mark-types`

원하는 스피치 마크 유형을 지정하십시오. 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 ssml`을 사용해야 합니다.

- outfile

메타데이터가 기록되는 출력 파일을 지정하십시오.

```
MaryLamb.txt
```

다음은 Unix, Linux, macOS용 형식으로 지정된 AWS CLI 예제입니다. 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 Polly에서 "Mary have little lamb" 텍스트에서 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 메타데이터는 Joanna 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 had a little lamb."이라는 간단한 문장에 대해 화면에서 요청된 메타데이터가 어떻게 표시 되는지 보여줍니다. 이 예제에서는 간소화를 위해 SSML 스피치 마크를 포함하지 않습니다.

다음은 Unix, Linux, macOS용 형식으로 지정된 AWS CLI 예제입니다. 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 [Viseme 및 Amazon Polly \(p. 79\)](#) 항목을 참조하십시오.

## 예제 2 : SSML이 있는 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 지정하십시오. 예제를 더 쉽게 읽도록 viseme 스피치 마크는 포함하지 않았지만 포함할 수도 있습니다.

다음은 Unix, Linux, macOS용 형식으로 지정된 AWS CLI 예제입니다. 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" }
```

## 스피치 마크 요청(콘솔)

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

스피치 마크를 생성하려면(콘솔)

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/polly/>에서 Amazon Polly 콘솔을 엽니다.
2. [Text-to-Speech] 탭을 선택합니다.
3. [Plain Text] 탭을 계속 사용하거나 [SSML] 탭을 선택합니다.
4. 텍스트를 입력란에 입력하거나 붙여 넣습니다.
5. [Language and region]에서 텍스트 언어를 선택합니다.
6. [Voice]에서 텍스트에 사용하려는 음성을 선택합니다.
7. 텍스트 발음을 변경하려면 [Customize Pronunciation]를 선택하고 [Apply Lexicon]에서 원하는 어휘를 선택합니다.
8. 스피치가 최종인지 확인하려면 [Listen to speech]를 선택하십시오.
9. [Change File Format]을 선택합니다.

Note

MP3, OGG, PCM 형식을 다운로드하면 스피치 마크가 생성되지 않습니다.

10. [File Format]에서 [Speech Marks]를 선택합니다.
11. [Speech Mark Types]에서 생성할 스피치 마크 유형을 선택합니다. [SSML] 메타데이터를 선택하는 옵션은 SSML 탭에서만 사용할 수 있습니다. Amazon Polly에 SSML을 사용하는 방법에 대한 자세한 내용은 [SSML 문서로부터 스피치 생성 \(p. 84\)](#)을 참조하십시오.
12. [Change]를 선택합니다.
13. [Download Speech Marks]를 선택합니다.



이 문  
자  
는  
이스  
케이  
프  
코드  
(큰  
따  
옴  
표)  
&ap;  
퍼  
샌  
드  
&abs;  
포  
스  
트  
로  
피  
또  
는  
작  
은  
따  
옴  
표  
&lt;  
다  
작  
은  
기  
호  
&gt;  
다  
큰  
기  
호

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 코드와 같습니다.

```
--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 코드와 같습니다.

```
--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 코드와 같습니다.

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

## SSML(콘솔) 사용

SSML 태그를 사용하면 음성을 사용자 지정하고 발음, 볼륨 및 스피치 속도와 같은 음성 측면을 제어할 수 있습니다. AWS 콘솔에서 오디오로 변환하려는 SSML로 확장된 텍스트가 텍스트 투 스피치 페이지의 SSML 탭에 입력됩니다. 일반 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 내용은 [연습 2: 일반 텍스트 입력을 사용하여 스피치 합성\(콘솔\)](#) (p. 6)를 참조하십시오.

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

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

그런 다음 < speak > 태그 내부의 텍스트에서 특정 SSML 태그를 사용하여 텍스트를 소리 낼 방법을 사용자 지정할 수 있습니다. 일시 중지를 추가하거나, 음성 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 태그 \(p. 91\)](#)를 참조하십시오.

다음 예에서는 SSML 태그를 사용하여 Amazon Polly에서 짧은 단락을 말할 때 "W3C"를 "World Wide Web Consortium"으로 대체하도록 지시합니다. 또한 태그를 사용하여 일시 중지를 도입하고 단어를 속삭입니다. 이 연습의 결과를 [콘솔을 사용하여 어휘 적용\(스피치 합성\) \(p. 114\)](#)의 결과와 비교합니다.

예에 사용된 SSML에 대한 자세한 내용은 [지원되는 SSML 태그 \(p. 91\)](#)을 참조하십시오.

SSML로 확장된 텍스트(콘솔)로부터 스피치를 합성하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/polly/>에서 Amazon Polly 콘솔을 엽니다.
2. 아직 표시되지 않은 경우 Text-to-Speech(텍스트 투 스피치) 탭을 선택합니다.
3. [SSML] 탭을 선택합니다.
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 World Wide Web Consortium으로 대체하도록 Amazon Polly에 지시합니다.
- < amazon: effect name="whispered">Nice job</ amazon: effect >은 속삭이는 음성으로 "Nice job"이라는 두 번째 인스턴스를 말하도록 Amazon Polly에 지시합니다.

#### Note

AWS CLI를 사용하는 경우 입력 텍스트를 따옴표로 묶어 주변 코드와 구별합니다. Amazon Polly 콘솔은 사용자에게 코드를 보여주지 않으므로 이 콘솔을 사용할 때에는 입력 텍스트를 따옴표로 묶지 않습니다.

5. [Choose a language and region]에서 [English US]를 선택한 다음 음성을 선택합니다.
6. 스피치를 들으려면 음성 듣기를 선택합니다.
7. 스피치 파일을 저장하려면 Download [format]([포맷] 다운로드)을 선택합니다. 다른 형식으로 저장하려면 파일 형식 변경을 선택하고 원하는 형식을 선택합니다. 그런 다음 변경 및 Download [format]([형식] 다운로드)를 선택합니다.

## SSML(AWS CLI) 사용

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

#### 주제

- [SynthesizeSpeech 명령을 통한 SSML 사용 \(p. 88\)](#)
- [SSML로 확장된 문서 합성 \(p. 88\)](#)

- 일반 Amazon Polly 작업에 대한 SSML 사용 (p. 89)

## Synthesize-Speech 명령을 통한 SSML 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 `<say></say>` 태그와 함께 따옴표로 묶여 있는 간단한 문자열을 지정합니다.

### Important

Amazon Polly 콘솔에서 입력 텍스트 주위에 따옴표를 사용하지 않더라도 AWS CLI를 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 "<say>Hello <break time='300ms' /> World</say>"
```

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

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

다음은 Unix, Linux, macOS용 형식으로 지정된 AWS CLI 예제입니다. 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 ssm1 \  
--text '<say>Hello world</say>' \  
--output-format mp3 \  
--voice-id Joanna \  
speech.mp3
```

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

## SSML로 확장된 문서 합성

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.xml`인 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 단어의 발음 개](#) [선 \(p. 91\)](#)을 참조하십시오.

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

다음은 Unix, Linux, macOS용 형식으로 지정된 AWS CLI의 예입니다. 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.

## 일반 Amazon Polly 작업에 대한 SSML 사용

다음 예에서는 SSML 태그를 사용하여 일반적인 Amazon Polly 작업을 완료하는 방법을 보여줍니다. SSML 태그에 대한 자세한 내용은 [지원되는 SSML 태그 \(p. 91\)](#)을 참조하십시오.

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

다음은 Unix, Linux, macOS용 형식으로 지정된 AWS CLI 예제입니다. 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>
```

## 볼륨, 피치 및 속도 제어

피치, 말하기 속도 및 음성 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 방식 지정

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.

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

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 \(p. 100\)](#) 단원을 참조하십시오.

## 외국어 단어의 발음 개선

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 Polly에서 지원하는 언어 \(p. 17\)](#)를 참조하십시오.

다음 예에서 선택된 음성(Joanna)은 미국 영어 음성입니다. 기본적으로 Amazon Polly는 입력 텍스트가 해당 음성과 동일한 언어라고 가정합니다(이 경우 미국 영어). 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.

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

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 Polly에서는 다음 SSML 태그를 지원합니다.

작업	SSML 태그	신경 음성 사용 여부
<a href="#">일시 중지 추가 (p. 92)</a>	<break>	완전히 사용 가능
<a href="#">단어 강조 (p. 93)</a>	<emphasis>	사용할 수 없음
<a href="#">특정 단어에 대하여 다른 언어 지정 (p. 94)</a>	<lang>	완전히 사용 가능

작업	SSML 태그	신경 음성 사용 여부
텍스트에 사용자 지정 태그 배치 (p. 95)	<mark>	완전히 사용 가능
단락 사이에 일시 중지 추가 (p. 95)	<p>	완전히 사용 가능
철자대로의 발음 사용 (p. 95)	<phoneme>	완전히 사용 가능
볼륨, 말하기 속도 및 피치 제어 (p. 96)	<prosody>	부분 사용 가능
합성된 스피치의 최대 기간 설정 (p. 98)	<prosody amazon:max-duration>	사용할 수 없음
문장 사이에 일시 중지 추가 (p. 100)	<s>	완전히 사용 가능
특별한 유형의 단어를 말하는 방식 제어 (p. 100)	<say-as>	부분 사용 가능
SSML로 확장된 텍스트 식별 (p. 92)	<speack>	완전히 사용 가능
두문자어 및 약어 발음 (p. 103)	<sub>	완전히 사용 가능
스피치의 일부를 구체적으로 지정함으로써 발음 개선 (p. 103)	<w>	완전히 사용 가능
숨소리 추가 (p. 104)	<amazon:auto-breaths>	사용할 수 없음
대화 말투 (p. 106)	<amazon:domain name="conversational">	신경 음성만 선택
뉴스 진행자 말투 (p. 107)	<amazon:domain name="news">	신경 음성만 선택
동적 범위 압축 추가 (p. 107)	<amazon:effect name="drc">	완전히 사용 가능
부드러운 어투 (p. 109)	<amazon:effect phonation="soft">	사용할 수 없음
음색 제어 (p. 109)	<amazon:effect vocal-tract-length>	사용할 수 없음
속삭임 기능 (p. 110)	<amazon: effect name="whispered">	사용할 수 없음

지원되지 않는 SSML 태그를 신경 또는 표준 형식으로 사용할 경우 오류가 발생합니다.

## SSML로 확장된 텍스트 식별

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

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

신경 및 표준 TTS 형식 둘 다 이 태그를 지원합니다.

## 일시 중지 추가

```
<break>
```

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 중지 기간(초)입니다. 최대 지속 시간은 10s입니다.
- **[number]**ms: 일시 중지 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.

신경 및 표준 TTS 형식 둘 다 이 태그를 지원합니다.

## 단어 강조

<emphasis>

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 strong일 때보다는 덜합니다. 기본값은 Moderate입니다.
- 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>
```

현재 TTS 형식만 이 태그를 지원합니다.

## 특정 단어에 대하여 다른 언어 지정

<lang>

<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 Polly에서 지원하는 언어 \(p. 17\)](#)를 참조하십시오.

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

예를 들어 voice-id가 (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-id를 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 스피치를 합성할 때 선택적인 DefaultLangCode 옵션의 대체로 사용할 수도 있습니다. 그러나, 이렇게 하려면 SSML을 사용하여 텍스트 서식을 지정해야 합니다.

신경 및 표준 TTS 형식 둘 다 이 태그를 지원합니다.

## 텍스트에 사용자 지정 태그 배치

<mark>

텍스트 내에 사용자 지정 태그를 넣으려면 <mark> 태그를 사용합니다. Amazon Polly에서 태그에 수행하는 작업은 없지만, SSML 메타데이터에 있는 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" }
```

신경 및 표준 TTS 형식 둘 다 이 태그를 지원합니다.

## 단락 사이에 일시 중지 추가

<p>

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

```
<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.

신경 및 표준 TTS 형식 둘 다 이 태그를 지원합니다.

## 철자대로의 발음 사용

<phoneme>

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

두 속성에는 <phoneme> 태그가 필요합니다. 이러한 태그는 Amazon Polly에서 사용되는 음성 기호와 정확 한 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 내용은 [지원되는 언어의 음소 및 Viseme 표 \(p. 18\)](#) 단원 을 참조하십시오.

<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 예에서는 단어 "pecan"에 대해 각 행마다 서로 다른 발음이 지정됩니다. Amazon Polly는 pecan에 대해 기본 발음을 사용하는 대신 ph 속성에 지정된 발음을 사용합니다.

IPA(International Phonetic Alphabet)

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

X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet)

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

또한 표준 중국어는 음운 발음에 병음을 사용합니다.

병음

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

신경 및 표준 TTS 형식 둘 다 이 태그를 지원합니다.

## 볼륨, 말하기 속도 및 피치 제어

<prosody>

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 레벨로부터 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.

```
<say>
  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 기존 음색 변경이 없음을 의미하고, +5%는 약간 더 높은 기존 음색을 제공하며, -5%는 약간 더 낮은 기존 음색을 가져옵니다.

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

```
<speak>
  Do you like synthesized 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.

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

이 태그는 표준 TTS 형식에서 완전히 지원됩니다. <prosody volume> 및 <prosody rate> 태그는 NTTS에서 완전히 지원되지만 <prosody pitch> 태그는 지원되지 않습니다.

## 합성된 스피치의 최대 기간 설정

<prosody amazon:max-duration>

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.

- **ns**: 최대 기간(초)
- **nms**: 최대 기간(밀리초)

예를 들어, 다음 말한 텍스트의 최대 기간은 2초입니다.

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

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 올리지는 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:



```
<say>
  <prosody amazon:mas-duration="2400ms">
    Human speech is a powerful way to communicate.
  </prosody>
  <break strength="strong"/>
  <prosody amazon:mas-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:mas-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>
</say>
```

<prosody amazon:mas-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

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

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

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

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

    <prosody amazon:mas-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>
</say>
```

- <prosody> 태그를 rate 태그 내의 <prosody amazon:mas-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.

```
<say>
  <prosody amazon:mas-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>
```

현재 TTS 형식만 이 태그를 지원합니다.

## 문장 사이에 일시 중지 추가

<s>

텍스트의 행 또는 문장 사이에 일시 중지를 추가하려면 <s>p</s> 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>
```

신경 및 표준 TTS 형식 둘 다 이 태그를 지원합니다.

## 특별한 유형의 단어를 말하는 방식 제어

<say-as>

<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-as>를 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.

- characters 또는 spell-out: 텍스트의 각 문자를 a-b-c와 같이 철자로 나열합니다.

#### Note

신경 음성을 사용할 경우 현재 이 옵션이 지원되지 않습니다. Amazon Polly에 이 SSML 코드가 런타임에 사용되면 영향을 받는 문장은 관련된 표준 음성을 사용하여 합성되지만, 신경 음성을 사용하는 것처럼 이 문장에 요금이 부과됩니다.

- 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 1/2과 같은 혼합 분수 모두에 적용됩니다. 자세한 내용은 아래를 참조하십시오.
- unit: 숫자로 된 텍스트를 측정값으로 해석합니다. 값은 1/2inch와 같이 숫자 또는 분수 다음에, 사이의 공백이 없이, 단위 1가지가 뒤따르거나. 1meter와 같이 오직 1단위만 있어야 합니다.
- date: 텍스트를 날짜로 해석합니다. 날짜의 형식은 형식 속성과 함께 지정되어야 합니다(아래 참조). 자세한 내용은 아래를 참조하십시오.
- time: 1'21"과 같이 숫자로 된 텍스트를 기간(분과 초)으로 해석합니다.
- address: 텍스트를 거리 주소의 일부분으로 해석합니다.
- expletive: 태그 내에 포함된 콘텐츠에 "뽀" 소리를 냅니다.
- telephone: 2025551212과 같이 숫자로 된 텍스트를 7자리 또는 10자리의 전화 번호로 해석합니다. 또한 2025551212x345와 같이 이 값을 전화 번호에 구내 번호를 추가하는 데 사용할 수도 있습니다. 자세한 내용은 아래를 참조하십시오.

#### 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 언어는 전화번호로 설정된 번호를 자동으로 처리하므로 실제로 telephone SSML 태그를 구현하지 않습니다.

#### 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" 사이에 +가 있어야 합니다. Amazon Polly는 "3 1/2" 같이 +가 없는 혼수를 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:

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

다음 형식은 date 속성과 함께 사용할 수 있습니다.

- mdy: 년-월-일
- dmy: 년-월-일
- ymd: 년-월-일
- md: 월-일
- dm: 월-일
- ym: 년-월
- my: 년-월
- d: 일
- m: Month.
- y: Year.
- yyymmdd: 년-월-일 이 형식을 사용하는 경우 Amazon Polly가 물음표를 사용하여 날짜의 일부분을 건너 뛰도록 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 테스트해봅니다.

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

"문자" 또는 "철자" 기능을 제외하고 신경 및 표준 TTS 형식에서 이 태그가 지원됩니다. Amazon Polly에 이 SSML 코드가 런타임에 사용되면 영향을 받는 문장은 관련된 표준 음성을 사용하여 합성되지만, 신경 음성을 사용하는 것처럼 이 문장에 요금이 부과됩니다.

## 두문자어 및 약어 발음

<sub>

<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.

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

신경 및 표준 TTS 형식 둘 다 이 태그를 지원합니다.

## 스피치의 일부를 구체적으로 지정함으로써 발음 개선

<w>

<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 과거 시제 또는 과거 분사로 해석합니다.

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

```
<speack>
  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>.
</speack>
```

대체 의미를 지정하려면:

- 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 종류로, 역시 "bass"라고 하지만 다르게 발음됩니다. <w role="amazon:SENSE\_1">bass</w>를 사용하면 해당 오디오 텍스트를 기본이 아닌 발음(민물고기)으로 렌더링합니다.

다음을 합성하면 이 차이를 들을 수 있습니다.

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

#### Note

일부 언어의 경우 지원되는 다른 스피치 부분을 선택할 수 있습니다.

신경 및 표준 TTS 형식 둘 다 이 태그를 지원합니다.

## 숨소리 추가

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

자연스럽게 들리는 음성에는 올바르게 말하는 단어와 숨소리가 모두 포함됩니다. 음성 합성에 숨소리를 추가하면 더 자연스럽게 들리는 음성을 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 음성 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. 기본값은 medium입니다.

#### Note

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

기본값을 사용하여 숨소리를 설정하려면 속성 없이 <amazon:breath/>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 breathes
  <amazon:breath/> so that the
  text sounds correct.
</speak>
```

다음과 같이 구절 내에 개별 숨소리를 추가할 수 있습니다.

```
<say>
  <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>
</say>
```

## 자동화 모드

자동화 모드에서는 Amazon Polly 태그를 사용하여 <amazon:auto-breaths>에서 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 선택적 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.

- 선택적 파라미터 없이 자동화 모드를 사용합니다.

```
<say>
  <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>
</say>
```

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

```
<say>
  <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 Polly에서 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>
```

현재 TTS 형식만 이 태그를 지원합니다.

## 대화 말투

```
<amazon:domain name="conversational">
```

대화 스타일은 Neural 형식의 미국 영어(en-US) 전용 Matthew 및 Joanna 음성으로만 제공됩니다.

대화 말투를 사용하려면 SSML 태그와 다음 구문을 사용합니다.

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

예를 들어 다음과 같이 Matthew 또는 Joanna 음성으로 대화 스타일을 사용할 수 있습니다.



```
<say>
<amazon:domain name="conversational">
I really didn't know how this morning was going to start. And if I had known, I think
I might have just stayed in bed. Even with a cat sleeping on my head and my dog deciding
I really didn't need to move my legs and falling asleep across both of them. As it was,
I stayed there as long as my bladder would let me.
</amazon:domain>
</say>
```

## 뉴스 진행자 말투

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

뉴스 진행자 스타일은 Neural 형식의 미국 영어(en-US) 전용 Matthew 및 Joanna 음성으로만 제공됩니다.

뉴스 진행자 스타일을 사용하려면 SSML 태그와 다음 구문을 사용합니다.

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

예를 들어 다음과 같이 Matthew 또는 Joanna 음성으로 뉴스 진행자 스타일을 사용할 수 있습니다.

```
<say>
<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.

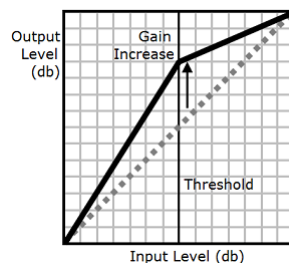
The Titanic started her trip from Southampton for New York on Wednesday. Late on Sunday
night she struck
an iceberg off the Grand Banks of Newfoundland. By wireless telegraphy she sent out signals
of distress,
and several liners were near enough to catch and respond to the call.
</amazon:domain>
</say>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

## 동적 범위 압축 추가

```
<amazon:effect name="drc">
```

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 사용하십시오.

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 사용하여 `</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:

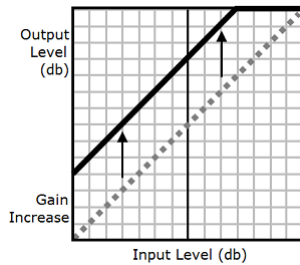
```
<speaking>
  Some audio is difficult to hear in a moving vehicle, but <amazon:effect name="drc">
  this audio
  is less difficult to hear in a moving vehicle.</amazon:effect>
</speaking>
```

#### Note

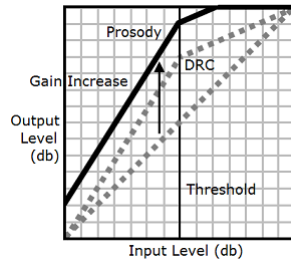
구문에 "drc"를 사용하면 대소문자를 구별합니다.amazon:effect

#### 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 사용하십시오. 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 넣으십시오. 예:

```
<speaking>
  <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>
</speaking>
```

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 사용하십시오.

신경 및 표준 TTS 형식 둘 다 이 태그를 지원합니다.

## 부드러운 어투

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

입력 데이터가 일반적인 음성보다 부드럽게 읽히도록 지정하려면 `<amazon:effect phonation="soft">` 태그를 사용합니다.

이때 다음 구문을 사용합니다.

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

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

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

현재 TTS 형식만 이 태그를 지원합니다.

## 음색 제어

`<amazon:effect vocal-tract-length>`

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 음성의 음색을 조절하려면 `vocal-tract-length` 태그를 사용합니다. 이 태그는 말하는 사람의 성도 길이를 변경하여 다르게 들리게 합니다. `vocal-tract-length`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 벗어난 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.

```
<speaking>
  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. 다양한 조합을 사용해 보면서 원하는 결과를 얻으십시오.

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>
```

현재 TTS 형식만 이 태그를 지원합니다.

## 속삭임 기능

<amazon:effect name="whispered">

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.

현재 TTS 형식만 이 태그를 지원합니다.

# 어휘 관리

발음 어휘를 사용하여 단어의 발음을 사용자 지정할 수 있습니다. Amazon Polly에서는 AWS 리전에 어휘를 저장하기 위해 사용할 수 있는 API 작업을 제공합니다. 따라서 그러한 어휘는 특정 리전과 관련이 있습니다. `SynthesizeSpeech` 작업을 사용하여 텍스트를 합성할 때 해당 리전에서 어휘를 한 개 이상 사용할 수 있습니다. 그래서 합성이 시작되기 전에 지정된 어휘가 입력 텍스트에 적용됩니다. 자세한 내용은 [SynthesizeSpeech \(p. 211\)](#) 단원을 참조하십시오.

## Note

이 어휘는 PLS(Pronunciation Lexicon Specification) W3C 권장 사항을 준수해야 합니다. 자세한 내용은 W3C 웹 사이트에서 [Pronunciation Lexicon Specification \(PLS\) Version 1.0](#)을 참조하십시오.

다음은 스피치 합성 엔진과 어휘를 사용하는 방법의 예입니다.

- 공통되는 단어는 "g3t sm4rt"(get smart)와 같이 문자를 대신하는 숫자 스타일이 사용되기도 합니다. 사람은 이 단어를 제대로 읽을 수 있지만 TTS(Text-to-Speech) 엔진은 철자에 따라 정확히 이름을 발음하여 문자 그대로 텍스트를 읽기 때문에 Amazon Polly를 사용하여 합성된 스피치를 사용자 지정하기 위해 어휘를 이용할 수 있습니다. 이 예에서는 어휘에서 "g3t sm4rt" 단어의 별칭(get smart)을 지정할 수 있습니다.
- W3C와 같이 두문자어를 텍스트에 포함할 수도 있습니다. 어휘를 사용하여 W3C라는 단어의 별칭을 정의할 수 있습니다. 그러면 확장된 전체 형태(World Wide Web Consortium)로 단어가 판독됩니다.

어휘를 사용하면 선택한 언어에 드물게 사용되는 단어를 Amazon Polly에서 발음하는 방법을 더 자세히 제어할 수 있습니다. 예를 들어, 음성 기호를 사용하여 발음을 지정할 수 있습니다. 자세한 내용은 W3C 웹 사이트에서 [Pronunciation Lexicon Specification \(PLS\) Version 1.0](#)을 참조하십시오.

## 주제

- [여러 어휘 적용 \(p. 112\)](#)
- [Amazon Polly 콘솔을 사용하여 어휘 관리 \(p. 113\)](#)
- [AWS CLI를 사용하여 어휘 관리 \(p. 116\)](#)

# 여러 어휘 적용

텍스트에 어휘를 5개까지 적용할 수 있습니다. 텍스트에 적용하는 2개 이상의 어휘에 같은 문자소가 나타나면 적용 순서에 따라 결과 스피치가 달라질 수 있습니다. 예를 들어, "Hello, my name is Bob."이라는 텍스트와 어휘소 2개가 다른 어휘에 있으면 둘 다 문자소 Bob을 사용합니다.

## LexA

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

## LexB

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

LexA 다음에 LexB가 오는 순서로 어휘가 나열되면 합성된 스피치는 "Hello, my name is Robert."입니다.  
LexB 다음에 LexA가 오는 순서로 나열되면 합성된 스피치는 "Hello, my name is Bobby."입니다.

Example -- LexB 앞에 LexA 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: "Hello, my name is Robert."

Example -- LexA 앞에 LexB 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: "Hello, my name is Bobby."

Amazon Polly 콘솔을 사용한 어휘 적용에 대한 자세한 내용은 [콘솔을 사용하여 어휘 적용\(스피치 합성\)](#) (p. 114) 항목을 참조하십시오.

## Amazon Polly 콘솔을 사용하여 어휘 관리

Amazon Polly 콘솔을 사용하여 어휘를 업로드, 다운로드, 적용, 필터링 및 삭제할 수 있습니다. 다음 절차에서는 이 프로세스를 각각 보여줍니다.

### 콘솔을 사용하여 어휘 업로드

발음 어휘를 사용하려면 먼저 업로드해야 합니다. 콘솔에서 어휘를 업로드할 수 있는 위치는 [Text-to-Speech] 탭과 [Lexicons] 탭 두 곳입니다.

다음 프로세스에서는 선택한 언어에 드물게 사용되는 단어와 문구의 발음 방식을 사용자 지정하는 데 사용할 어휘를 추가하는 방법을 설명합니다.

[Lexicons] 탭에서 어휘를 추가하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/polly/>에서 Amazon Polly 콘솔을 엽니다.
2. [Lexicons] 탭을 선택합니다.
3. Upload를 선택합니다.
4. 업로드할 어휘를 찾아봅니다. .pls 및 .xml 확장명을 사용하는 PLS 파일만 사용할 수 있습니다.
5. [Open]을 선택합니다. .pls 파일이건 .xml 파일이건 같은 이름의 어휘가 이미 있으면 업로드한 어휘가 기존 어휘를 덮어씁니다.

[Text-to-Speech] 탭에서 어휘를 추가하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/polly/>에서 Amazon Polly 콘솔을 엽니다.

2. [Text-to-Speech] 탭을 선택합니다.
3. [Customize pronunciation of words or phrases using lexicons]를 선택하고 [Upload lexicon]을 선택합니다.
4. 업로드할 어휘를 찾아봅니다. .pls 및 .xml 확장명을 사용하는 PLS 파일만 사용할 수 있습니다.
5. [Open]을 선택합니다. .pls 파일이건 .xml 파일이건 같은 이름의 어휘가 이미 있으면 업로드한 어휘가 기존 어휘를 덮어씁니다.

## 콘솔을 사용하여 어휘 적용(스피치 합성)

다음 절차에서는 w3c.pls 어휘를 적용하여 "W3C" 대신 "World Wide Web Consortium"을 사용하도록 입력 텍스트에 어휘를 적용하는 방법을 보여줍니다. 텍스트에 여러 어휘를 적용하면 위에서 아래로 적용되며 첫 번째 일치 항목이 나중 일치 항목보다 우선합니다. 어휘에 지정된 언어가 선택한 언어와 같아야 어휘가 텍스트에 적용됩니다.

일반 텍스트 또는 SSML 입력에 어휘를 적용할 수 있습니다.

### Example -- W3C.pls 어휘 적용

이 연습에 필요한 어휘를 생성하려면 [PutLexicon 작업 사용 \(p. 116\)](#) 항목을 참조하십시오. 일반 텍스트 편집기를 사용하여 항목 맨 위에 표시된 W3C.pls 어휘를 생성하십시오. 이 파일을 저장한 위치를 기억해 두십시오.

### 입력에 W3C.pls 어휘를 적용하려면

이 예에서는 "W3C"를 "World Wide Web Consortium"으로 대체할 어휘를 소개합니다. 이 연습의 결과를 미국 영어 및 다른 언어의 [SSML\(콘솔\) 사용 \(p. 86\)](#) 결과와 모두 비교하십시오.

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/polly/>에서 Amazon Polly 콘솔을 엽니다.
2. 다음 중 하나를 수행합니다.

- [Plain text] 탭을 선택한 후 이 텍스트를 입력하거나 텍스트 입력란에 붙여 넣습니다.

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

- [SSML] 탭을 선택한 후 이 텍스트를 입력하거나 텍스트 입력란에 붙여 넣습니다.

```
<speaking>He wasn't paying attention.<break time="1s"/>  
In the middle of the 10/3/2014 W3C meeting  
he shouted, "Score!" quite loudly.</speaking>
```

3. [Choose a language and region] 목록에서 [English US]를 선택한 후 이 텍스트에 사용할 음성을 선택합니다.
4. [Customize pronunciation of words or phrases using lexicons]를 선택합니다.
5. 어휘 목록에서 [w3c (English, US)]를 선택합니다.

[w3c (English, US)] 어휘가 목록에 없으면 [Upload lexicon]을 선택하여 업로드한 후 목록에서 선택합니다. 이 어휘를 생성하려면 [PutLexicon 작업 사용 \(p. 116\)](#) 단원을 참조하십시오.

6. 스피치를 즉시 들으려면 [Listen to speech]를 선택합니다.
7. 스피치를 파일에 저장하려면
  - a. [Save speech to MP3]를 선택합니다.



- b. 다른 파일 형식으로 변경하려면 [Change file format]을 선택하여 원하는 파일 형식을 선택한 다음 [Change]를 선택합니다.

이전 단계를 반복하되 다른 언어를 선택하고 출력의 차이를 확인하십시오.

## 콘솔을 사용하여 어휘 목록 필터링

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

언어별로 나열되도록 어휘를 필터링하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/polly/>에서 Amazon Polly 콘솔을 엽니다.
2. [Lexicons] 탭을 선택합니다.
3. [Filter]를 선택합니다.
4. 언어 목록에서 필터링할 언어를 선택합니다.

선택한 언어의 어휘만 목록에 표시됩니다.

## 콘솔을 사용하여 어휘 다운로드

다음 프로세스에서는 어휘를 1개 이상 다운로드하는 방법을 설명합니다. 파일의 어휘 항목을 추가, 제거 또는 수정하고 다시 업로드하여 어휘를 최신 상태로 유지할 수 있습니다.

어휘를 1개 이상 다운로드하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/polly/>에서 Amazon Polly 콘솔을 엽니다.
2. [Lexicons] 탭을 선택합니다.
3. 다운로드할 어휘를 선택합니다.
  - a. 어휘 1개를 다운로드하려면 목록에서 어휘 이름을 선택합니다.
  - b. 여러 어휘를 압축된 아카이브 파일 하나로 다운로드하려면 목록에서 다운로드할 각 항목 옆의 확인란을 선택합니다.
4. [Download]를 선택합니다.
5. 어휘를 다운로드할 폴더를 엽니다.
6. Save를 선택합니다.

## 콘솔을 사용하여 어휘 삭제

어휘를 삭제하려면

다음 프로세스에서는 어휘를 삭제하는 방법을 설명합니다. 어휘를 삭제한 후에는 다시 추가해야 다시 사용할 수 있습니다. 각 어휘 옆의 확인란을 선택하여 어휘 1개 이상을 동시에 삭제할 수 있습니다.

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/polly/>에서 Amazon Polly 콘솔을 엽니다.
2. [Lexicons] 탭을 선택합니다.
3. 목록에서 삭제할 어휘를 1개 이상 선택합니다.
4. [Delete]를 선택합니다.
5. [Delete]를 선택하여 리전에서 어휘를 제거하거나 [Cancel]을 선택하여 그대로 유지합니다.

## AWS CLI를 사용하여 어휘 관리

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

주제

- [PutLexicon 작업 사용 \(p. 116\)](#)
- [GetLexicon 작업 사용 \(p. 120\)](#)
- [ListLexicons 작업 사용 \(p. 121\)](#)
- [DeleteLexicon 작업 사용 \(p. 121\)](#)

### PutLexicon 작업 사용

Amazon Polly를 통해 [PutLexicon \(p. 203\)](#)을 사용하여 특정 AWS 리전의 계정에 발음 어휘를 저장할 수 있습니다. 그런 다음 서비스에서 텍스트 합성을 시작하기 전에 적용할 [SynthesizeSpeech \(p. 211\)](#) 요청에 저장된 어휘를 1개 이상 지정할 수 있습니다. 자세한 내용은 [어휘 관리 \(p. 112\)](#) 단원을 참조하십시오.

이 단원에서는 예제 어휘와 어휘 저장 및 테스트를 위한 단계별 지침을 제공합니다.

#### Note

이 어휘는 PLS(Pronunciation Lexicon Specification) W3C 권장 사항을 준수해야 합니다. 자세한 내용은 W3C 웹 사이트에서 [Pronunciation Lexicon Specification \(PLS\) Version 1.0](#)을 참조하십시오.

### 예제 1: 어휘소가 1개인 어휘

다음의 W3C PLS 준수 어휘를 고려하십시오.

```
<?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>
```

다음을 참조하십시오.

- <lexicon> 요소에 지정된 속성 2개:
  - `xml:lang` 속성은 어휘가 적용되는 언어 코드 `en-US`를 지정합니다. `SynthesizeSpeech` 호출에 지정하는 음성의 언어 코드가 동일할 경우(`en-US`) Amazon Polly에서 이 예제 어휘를 사용할 수 있습니다.

#### Note

`DescribeVoices` 작업을 사용하여 음성과 연결된 언어 코드를 찾을 수 있습니다.

- `alphabet` 속성은 IPA를 지정하므로 IPA(International Phonetic Alphabet) 알파벳이 발음에 사용됩니다. IPA는 표음 문자 중 하나입니다. 또한 Amazon Polly는 X-SAMPA(Extended Speech Assessment Methods Phonetic Alphabet)를 지원합니다.

- <lexeme> 요소는 <grapheme>(단어의 텍스트 표현)과 <alias> 사이의 매핑을 설명합니다.

이 어휘를 테스트하려면 다음을 수행하십시오.

1. 어휘를 example.pls로 저장합니다.
2. put-lexicon AWS CLI 명령을 실행하여 us-east-2 리전에 w3c라는 이름으로 어휘를 저장합니다.

```
aws polly put-lexicon \  
--name w3c \  
--content file://example.pls
```

3. synthesize-speech 명령을 실행하여 샘플 텍스트를 오디오 스트림으로 합성하고(speech.mp3) 선택적 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. 결과 speech.mp3를 재생하고 텍스트의 W3C라는 단어가 World Wide Web Consortium으로 바뀌는지 확인합니다.

앞에 나온 예제 어휘에 별칭이 사용되며 어휘에 사용된 IPA 알파벳은 사용되지 않습니다. 다음 어휘는 APA 알파벳과 <phoneme> 요소를 사용하여 음성 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>
```

이 어휘를 테스트하려면 같은 단계를 따르십시오. "pecan"이라는 단어가 있는 입력 텍스트(예: "Pecan pie is delicious")를 지정합니다.

## 예제 2: 어휘소가 여러 개인 어휘

이 예제에서는 어휘에 지정된 어휘소가 합성할 입력 텍스트에만 적용됩니다. 다음 어휘를 고려하십시오.

```
<?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>
```

어휘가 어휘소 3개를 지정하고, 그 중 2개가 다음과 같이 문자소 W3C의 별칭을 정의합니다.

- 첫 번째 <lexeme> 요소가 별칭(World Wide Web Consortium)을 정의합니다.
- 두 번째 <lexeme>이 대체 별칭(WWW Consortium)을 정의합니다.

Amazon Polly가 첫 번째 대체 항목을 어휘의 지정된 문자소에 사용합니다.

세 번째 <lexeme>이 Consortium이라는 단어의 대체 항목(Community)을 정의합니다.

먼저 이 어휘를 테스트하겠습니다. 다음 샘플 텍스트를 오디오 파일(speech.mp3)로 합성하고 SynthesizeSpeech 호출에 어휘를 지정합니다.

```
The W3C is a Consortium
```

SynthesizeSpeech가 먼저 다음과 같이 어휘를 적용합니다.

- 첫 번째 어휘소에 따라 단어 W3C가 World Wide Web Consortium으로 바뀝니다. 바뀐 텍스트가 다음과 같이 표시됩니다.

```
The World Wide Web Consortium is a Consortium
```

- 세 번째 어휘소에 정의된 별칭은 원래 텍스트에 포함된 단어 Consortium에만 적용되므로 다음과 같은 텍스트가 생성됩니다.

```
The World Wide Web Consortium is a Community.
```

다음과 같이 AWS CLI를 사용하여 이 항목을 테스트할 수 있습니다.

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 파일로 합성하고(speech.mp3) 선택적 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. 결과 speech.mp3 파일을 재생하여 합성된 스피치에 테스트 변경 사항이 반영되는지 확인합니다.

## 예제 3: 여러 어휘 지정

SynthesizeSpeech 호출에서 여러 어휘를 지정할 수 있습니다. 이 경우에는 지정된 첫 번째 어휘(왼쪽에서 오른쪽)가 이전의 어휘를 재정의합니다.

다음 어휘 2개를 고려하십시오. 각 어휘는 동일한 문자소 W3C의 여러 가지 별칭을 설명합니다.

- 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>
```

이 어휘를 w3c 및 w3cAlternate로 각각 저장하겠습니다. w3c 호출에서 w3cAlternate 다음에 SynthesizeSpeech가 오는 순서로 어휘를 지정할 경우 첫 번째 어휘에 정의된 W3C의 별칭이 두 번째 항목보다 우선 적용됩니다. 어휘를 테스트하려면 다음을 수행하십시오.

1. 로컬에 있는 w3c.pls 및 w3cAlternate.pls 파일에 어휘를 저장합니다.
2. put-lexicon AWS CLI 명령을 사용하여 이 어휘를 업로드합니다.
  - w3c.pls 어휘를 업로드하고 w3c로 저장합니다.

```
aws polly put-lexicon \  
--name w3c \  
--content file://w3c.pls
```

- w3cAlternate.pls 어휘를 서비스에 w3cAlternate로 업로드합니다.

```
aws polly put-lexicon \  
--name w3cAlternate \  
--content file://w3cAlternate.pls
```

3. synthesize-speech 명령을 실행하여 샘플 텍스트를 오디오 스트림으로 합성하고(speech.mp3) 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. 결과 speech.mp3를 테스트합니다. 다음과 같이 읽힙니다.

```
PLS is a World Wide Web Consortium recommendation
```

## PutLexicon API용 추가 코드 예제

- Java 샘플: [PutLexicon \(p. 131\)](#)
- Python(Boto3) 예제: [PutLexicon \(p. 137\)](#)

## GetLexicon 작업 사용

Amazon Polly는 [GetLexicon \(p. 195\)](#) API 작업을 제공하여 특정 리전의 계정에 저장한 발음 어휘 내용을 검색합니다.

다음 get-lexicon AWS CLI 명령은 example 어휘 내용을 검색합니다.

```
aws polly get-lexicon \  
--name example
```

계정에 저장된 어휘가 아직 없을 경우 PutLexicon 작업을 사용하여 어휘를 저장할 수 있습니다. 자세한 내용은 [PutLexicon 작업 사용 \(p. 116\)](#) 단원을 참조하십시오.

다음은 응답 예입니다. 응답은 어휘 내용 외에도 어휘가 적용되는 언어, 어휘에 정의된 어휘소 개수, 리소스의 ARN(Amazon 리소스 이름) 및 어휘 크기(바이트 단위) 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  
  }  
}
```

```
}
```

## GetLexicon API용 추가 코드 예제

- Java 샘플: [GetLexicon](#) (p. 130)
- Python(Boto3) 예제: [GetLexicon](#) (p. 135)

## ListLexicons 작업 사용

Amazon Polly에서는 특정 AWS 리전의 계정에서 발음 어휘 목록을 가져오는 데 사용할 수 있는 [ListLexicons](#) (p. 199) API 작업을 제공합니다. 다음 AWS CLI 호출은 us-east-2 리전의 계정에 있는 어휘를 나열합니다.

```
aws polly list-lexicons
```

다음은 w3c 및 tomato라는 어휘 2개를 표시하는 예제 응답입니다. 각 어휘마다 응답은 어휘가 적용되는 언어, 어휘에 정의된 어휘소 개수, 바이트 단위의 크기 등의 메타데이터를 반환합니다. 언어 코드는 언어와 어휘에 정의된 어휘소가 적용되는 위치를 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"
    }
  ]
}
```

## ListLexicon API용 추가 코드 예제

- Java 샘플: [ListLexicons](#) (p. 130)
- Python(Boto3) 예제: [ListLexicon](#) (p. 136)

## DeleteLexicon 작업 사용

Amazon Polly에서는 [DeleteLexicon](#) (p. 190) API 작업을 제공하여 계정의 특정 AWS 리전에서 발음 어휘를 삭제합니다. 다음 AWS CLI는 지정된 어휘를 삭제합니다.

다음은 Unix, Linux, macOS용 형식으로 지정된 AWS CLI 예제입니다. 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
```

## DeleteLexicon API용 추가 코드 예제

- Java 샘플: [DeleteLexicon \(p. 128\)](#)
- Python(Boto3) 예제: [DeleteLexicon \(p. 135\)](#)



# 긴 오디오 파일 생성

긴 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. `StartSpeechSynthesisTask`를 사용하여 입력 텍스트를 제출함으로써 합성 요청이 생성되면 Amazon Polly에서 요청을 대기열에 추가한 다음 시스템 리소스가 제공되면 바로 해당 요청을 백그라운드에서 비동기식으로 처리합니다. 그런 다음 Amazon Polly에서 결과 음성 또는 스피치 마크 스트림을 사용자의 필수 Amazon Simple Storage Service(Amazon S3) 버킷에 직접 업로드하고 사용자의 선택적 SNS 주제에서의 완료된 파일의 가용성을 알립니다.

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. 선택적 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 설명서](#)를 참조하십시오.

## 암호화

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 암호화](#)를 활성화합니다.

## 주제

- [비동기식 합성에 대한 IAM 정책 설정](#) (p. 123)
- [긴 오디오 파일 생성\(콘솔\)](#) (p. 124)
- [긴 오디오 파일 생성\(CLI\)](#) (p. 125)

# 비동기식 합성에 대한 IAM 정책 설정

비동기식 합성 기능을 사용하려면 다음을 허용하는 IAM 정책이 필요합니다.

- 새 Amazon Polly 작업 사용
- 출력 S3 버킷에 쓰기
- 상태 SNS 주제에 게시 [선택]

다음 정책은 비동기식 합성에 필요한 권한만 부여하며 IAM 사용자에게 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"
    }
  ]
}
```

## 긴 오디오 파일 생성(콘솔)

Amazon Polly 콘솔을 사용하면 AWS CLI와 함께 사용할 수 있는 동일한 기능으로 비동기식 합성을 사용하여 긴 음성을 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 Console에 로그인한 후 <https://console.aws.amazon.com/polly/>에서 Amazon Polly 콘솔을 엽니다.
2. [Text-to-Speech] 탭을 선택합니다.
3. 일반 텍스트 탭 또는 SSML 탭에서 텍스트를 입력란에 입력하거나 붙여 넣습니다.
4. 해당 텍스트에 대한 언어, 리전 및 음성을 선택합니다.
5. S3에 합성을 선택합니다.

### Note

텍스트 길이가 실시간 SynthesizeSpeech 작업에 대한 제한을 초과하면 다운로드 옵션과 음성 듣기 옵션 모두 회색 음영으로 표시됩니다.

6. 이전에 비동기식 합성을 사용한 적이 없으면 출력 파일을 저장할 위치를 선택할 수 있도록 S3 합성 작업 설정 변경 상자가 표시됩니다.
  - 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 선택적 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 참조하십시오.

- d. 합성을 선택합니다.

**S3 합성 작업 설정을 변경하려면**

1. 콘솔의 텍스트 투 스피치 탭에서 S3 작업 설정 변경을 선택합니다.
2. 대상 Amazon S3 버킷의 이름, 해당 접두사 키 또는 SNS 주제 식별자를 원하는 대로 변경합니다.
3. 완료되면 합성을 선택합니다.

**스피치 합성 작업에 대한 정보를 검색하려면**

1. 콘솔에서 S3 합성 작업 탭을 선택합니다.
2. 작업은 날짜 순서대로 표시됩니다. 작업을 필터링하려면 필터를 선택한 다음 사용할 필터를 선택합니다.
3. 특정 작업의 세부 정보를 보려면 연결된 작업 ID를 선택합니다.

## 긴 오디오 파일 생성(CLI)

Amazon Polly 비동기식 합성 기능은 다음 세 가지 `SpeechSynthesisTask`를 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 이외에도 Amazon S3 버킷 이름도 필요합니다. 작업에 대한 상태 알림을 받으려는 경우 다른 선택적 두 개의 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 예를 사용하여 미국 동부(오하이오) 리전에서 `start-speech-synthesis-task` AWS CLI 명령을 실행할 수 있습니다.

다음은 Unix, Linux, macOS용 형식으로 지정된 AWS CLI 예제입니다. 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 scheduled를 표시합니다. Amazon Polly에서 작업 처리를 시작하면 상태가 inProgress로 변경되고 나중에 completed 또는 failed로 변경됩니다. 작업이 실패하면 GetSpeechSynthesisTask 또는 ListSpeechSynthesisTasks 작업을 호출할 때 오류 메시지가 반환됩니다.

작업이 완료되면 OutputUri에서 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-id에 의해 반환된 StartSpeechSynthesisTask가 필요합니다.

예를 들어 다음 예를 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 예를 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 단원에서는 Amazon Polly 탐색에 사용할 수 있는 코드 샘플과 예제 애플리케이션을 제공합니다.

주제

- [샘플 코드 \(p. 128\)](#)
- [예제 애플리케이션 \(p. 139\)](#)

샘플 코드 항목에서는 프로그래밍 언어별로 분류되고 여러 Amazon Polly 기능에 대한 예제로 구분된 코드 조각을 제공합니다. 예제 애플리케이션 항목에는 Amazon Polly를 탐색하는 데 독립적으로 사용할 수 있는 프로그래밍 언어별로 분류된 애플리케이션이 나와 있습니다.

이 예제를 사용하려면 먼저 [Amazon Polly 작동 방식 \(p. 3\)](#) 항목을 읽고 [Amazon Polly 시작하기 \(p. 4\)](#)에 설명된 단계를 따르는 것이 좋습니다.

## 샘플 코드

이 항목에서는 Amazon Polly를 탐색하는 데 사용할 수 있는 다양한 기능을 위한 코드 샘플을 제공합니다.

프로그래밍 언어별 샘플 코드

- [Java 샘플 \(p. 128\)](#)
- [Python 샘플 \(p. 134\)](#)

## Java 샘플

다음 코드 샘플은 Java 기반 애플리케이션을 사용하여 Amazon Polly의 다양한 작업을 수행하는 방법을 보여줍니다. 이러한 샘플은 전체 예제 중 일부이며 [AWS SDK for Java](#)를 사용하는 Java 애플리케이션에 포함될 수 있습니다.

코드 조각

- [DeleteLexicon \(p. 128\)](#)
- [DescribeVoices \(p. 129\)](#)
- [GetLexicon \(p. 130\)](#)
- [ListLexicons \(p. 130\)](#)
- [PutLexicon \(p. 131\)](#)
- [StartSpeechSynthesisTask \(p. 132\)](#)
- [스피치 마크 \(p. 133\)](#)
- [SynthesizeSpeech \(p. 134\)](#)

## DeleteLexicon

다음 Java 코드 샘플은 Java 기반 애플리케이션을 사용하여 AWS 리전에 저장된 특정 어휘를 삭제하는 방법을 보여줍니다. 삭제된 어휘는 음성 합성에서 사용할 수 없으며, [GetLexicon](#) [ListLexicon](#) API를 사용하여 검색할 수도 없습니다.

이 작업에 대한 자세한 내용은 [DeleteLexicon](#) API에 대한 참조 자료를 참조하십시오.

```
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 샘플은 Java 기반 애플리케이션을 사용하여 음성 합성 요청 시 사용할 수 있는 음성 목록을 생성하는 방법을 보여줍니다. 필요한 경우, 언어 코드를 지정하여 사용 가능한 음성을 필터링할 수도 있습니다. 예를 들어 en-US를 지정하면 모든 미국 영어 음성 목록이 반환됩니다.

이 작업에 대한 자세한 내용은 [DescribeVoices](#) API에 대한 참조 자료를 참조하십시오.

```
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);
            } while (nextToken != null);
        } catch (Exception e) {
            System.err.println("Exception caught: " + e);
        }
    }
}
```

```
        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 AWS 리전에 저장된 특정 발음 어휘의 내용을 생성하는 방법을 보여줍니다.

이 작업에 대한 자세한 내용은 [GetLexicon](#) API에 대한 참조 자료를 참조하십시오.

```
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 샘플에서는 Java 기반 애플리케이션을 사용하여 AWS 리전에 저장된 발음 어휘 목록을 생성하는 방법을 보여줍니다.

이 작업에 대한 자세한 내용은 [ListLexicons](#) API에 대한 참조 자료를 참조하십시오.

```
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](#)에 대한 참조 자료를 참조하십시오.

```

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

다음은 Java 기반 애플리케이션을 사용하여 긴 스피치를 합성(최대 100,000자)하고 이를 Amazon S3 버킷에 직접 저장하는 방법을 보여주는 Java 코드 샘플입니다.

자세한 내용은 [StartSpeechSynthesisTask API](#)에 대한 참조 자료를 참조하십시오.

```
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-2:561828872312: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);

        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 입력 텍스트의 스피치 마크를 합성하는 방법을 보여줍니다. 이 기능은 SynthesizeSpeech API를 사용합니다.

이 기능에 대한 자세한 내용은 [스피치 마크](#) (p. 79) 단원을 참조하십시오.

API에 대한 자세한 내용은 [SynthesizeSpeech](#) API에 대한 참조 자료를 참조하십시오.

```
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

다음은 거의 실시간으로 처리하기 위해 Java 기반 애플리케이션을 사용하여 스피치를 더 짧은 텍스트로 합성하는 방법을 보여주는 Java 코드 샘플입니다.

자세한 내용은 [SynthesizeSpeech API](#)에 대한 참조 자료를 참조하십시오.

```
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.");

        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 Polly의 다양한 작업을 수행하는 방법을 보여줍니다. 이러한 샘플은 전체 예제 중 일부이며 [AWS SDK for Python \(Boto\)](#)을 사용하는 Python 애플리케이션에 포함될 수 있습니다.

코드 조각

- [DeleteLexicon](#) (p. 135)
- [GetLexicon](#) (p. 135)
- [ListLexicon](#) (p. 136)
- [PutLexicon](#) (p. 137)
- [StartSpeechSynthesisTask](#) (p. 138)
- [SynthesizeSpeech](#) (p. 138)

## DeleteLexicon

다음 Python 코드 예제에서는 AWS SDK for Python (Boto)을 사용하여 로컬 AWS 구성에 지정된 리전에서 어휘를 삭제합니다. 예제에서는 지정된 어휘만 삭제합니다. 어휘를 삭제하기 전에 계속할지 확인하는 메시지가 나타납니다.

다음 코드 예제에서는 AWS SDK 구성 파일에 저장된 기본 자격 증명을 사용합니다. 구성 파일 생성에 대한 자세한 내용은 [3.1단계: AWS Command Line Interface\(AWS CLI\) 설정](#) (p. 7) 항목을 참조하십시오.

이 작업에 대한 자세한 내용은 [DeleteLexicon](#) API에 대한 참조 자료를 참조하십시오.

```
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 코드 예제에서는 AWS SDK for Python (Boto)을 사용하여 AWS 리전에 저장된 모든 어휘를 검색합니다. 예제에서 어휘 이름을 명령줄 파라미터로 받아들이고 해당 어휘만 가져와 어휘가 로컬로 저장된 tmp 경로를 출력합니다.

다음 코드 예제에서는 AWS SDK 구성 파일에 저장된 기본 자격 증명을 사용합니다. 구성 파일 생성에 대한 자세한 내용은 [3.1단계: AWS Command Line Interface\(AWS CLI\) 설정 \(p. 7\)](#) 항목을 참조하십시오.

이 작업에 대한 자세한 내용은 [GetLexicon](#) API에 대한 참조 자료를 참조하십시오.

```
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)을 사용하여 로컬 AWS 구성에 지정된 리전의 계정에 있는 어휘를 나열합니다. 구성 파일 생성에 대한 자세한 내용은 [3.1단계: AWS Command Line Interface\(AWS CLI\) 설정 \(p. 7\)](#) 항목을 참조하십시오.

이 작업에 대한 자세한 내용은 [ListLexicons](#) API에 대한 참조 자료를 참조하십시오.

```
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 애플리케이션을 사용하여 AWS 리전에 발음 어휘를 저장하는 방법을 보여줍니다.

이 작업에 대한 자세한 내용은 [PutLexicon API](#)에 대한 참조 자료를 참조하십시오.

다음은 참조하십시오.

- 로컬 어휘 파일 이름 및 저장된 어휘 이름을 제공하여 코드를 업데이트해야 합니다.
- 예제에서는 하위 디렉터리 pls에 어휘 파일이 생성되어 있다고 가정합니다. 경로를 적절히 업데이트해야 합니다.

다음 코드 예제에서는 AWS SDK 구성 파일에 저장된 기본 자격 증명을 사용합니다. 구성 파일 생성에 대한 자세한 내용은 [3.1단계: AWS Command Line Interface\(AWS CLI\) 설정 \(p. 7\)](#) 항목을 참조하십시오.

이 작업에 대한 자세한 내용은 [PutLexicon API](#)에 대한 참조 자료를 참조하십시오.

```
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)을 사용하여 로컬 AWS 구성에 지정된 리전의 계정에 있는 어휘를 나열합니다. 구성 파일 생성에 대한 자세한 내용은 [3.1단계: AWS Command Line Interface\(AWS CLI\) 설정 \(p. 7\)](#) 항목을 참조하십시오.

자세한 내용은 [StartSpeechSynthesisTask](#) API에 대한 참조 자료를 참조하십시오.

```
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.')

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)을 사용하여 로컬 AWS 구성에 지정된 리전의 계정에 있는 어휘를 나열합니다. 구성 파일 생성에 대한 자세한 내용은 [3.1단계: AWS Command Line Interface\(AWS CLI\) 설정 \(p. 7\)](#) 항목을 참조하십시오.

API에 대한 자세한 내용은 [SynthesizeSpeech](#) API에 대한 참조 자료를 참조하십시오.

```
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.')

file = open('speech.mp3', 'wb')
file.write(response['AudioStream'].read())
file.close()
```

## 예제 애플리케이션

이 단원에서는 Amazon Polly를 탐색하는 데 사용할 수 있는 예제 애플리케이션을 제공합니다.

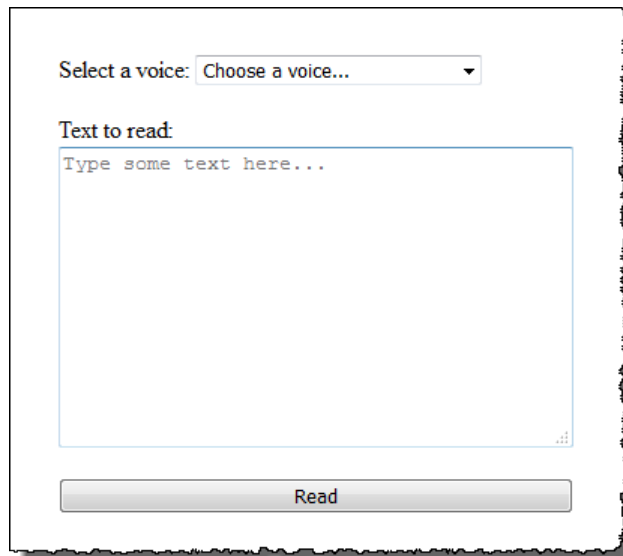
프로그래밍 언어별 예제 애플리케이션

- [Python 예제\(HTML5 클라이언트 및 Python 서버\)](#) (p. 139)
- [Java 예제](#) (p. 148)
- [iOS 예제](#) (p. 151)
- [Android 예제](#) (p. 152)

## Python 예제(HTML5 클라이언트 및 Python 서버)

이 예제 애플리케이션은 다음과 같이 구성되어 있습니다.

- HTTP 조각난 전송 코딩을 사용하는 HTTP 1.1 서버([조각난 전송 코딩 참조](#))
- HTTP 1.1 서버와 상호 작용하는 간단한 HTML5 사용자 인터페이스(아래에 표시):



이 예제에서는 Amazon Polly를 사용하여 브라우저 기반 HTML5 애플리케이션에서 스피치를 스트리밍하는 방법을 보여줍니다. 대화 시스템, 화면 판독기 등 응답이 중요한 요인으로 간주되는 경우에는 텍스트가 합성될 때 Amazon Polly에서 생성된 오디오 스트림을 소비하는 것이 좋습니다.

이 예제 애플리케이션을 실행하려면 다음 사항이 필요합니다.

- HTML5 및 EcmaScript5 표준에 맞는 웹 브라우저(예: Chrome 23.0 이상, Firefox 21.0 이상, Internet Explorer 9.0 이상)
- Python 버전 3.0 이상

애플리케이션을 테스트하려면

1. 서버 코드를 `server.py`로 저장합니다. 코드는 [Python 예제: Python 서버 코드\(server.py\)](#) (p. 143) 항목을 참조하십시오.
2. HTML5 클라이언트 코드를 `index.html`로 저장합니다. 코드는 [Python 예제: HTML5 사용자 인터페이스\(index.html\)](#) (p. 140) 항목을 참조하십시오.
3. `server.py`를 저장한 경로에서 다음 명령을 실행하여 애플리케이션을 시작합니다(명령을 실행할 때 `python` 대신 `python3`을 사용해야 하는 일부 시스템에서).

```
$ python server.py
```

애플리케이션이 시작되면 터미널에 URL이 나타납니다.

4. 터미널에 표시된 URL을 웹 브라우저에서 엽니다.

파라미터로 사용할 애플리케이션 서버의 주소와 포트를 `server.py`에 전달할 수 있습니다. 자세한 내용을 보려면 `python server.py -h`를 실행하십시오.

5. 스피치를 들으려면 목록에서 음성을 선택하고 텍스트를 입력한 후 [Read]를 선택합니다. Amazon Polly에서 오디오 데이터의 첫 번째 사용 가능한 청크를 전송하자마자 스피치가 재생되기 시작합니다.
6. 애플리케이션 테스트가 끝나 Python 서버를 중지하려면 서버가 실행되는 터미널에서 Ctrl+C를 누릅니다.

#### Note

서버에서 AWS SDK for Python (Boto)을 사용하여 Boto3 클라이언트를 생성합니다. 클라이언트는 컴퓨터의 AWS 구성 파일에 저장된 자격 증명을 사용하여 Amazon Polly에 대한 요청에 서명하고 인증합니다. AWS 구성 파일을 생성하고 자격 증명을 저장하는 방법에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서에서 [AWS 명령줄 인터페이스 구성](#) 단원을 참조하십시오.

## Python 예제: HTML5 사용자 인터페이스(index.html)

이 단원에서는 [Python 예제\(HTML5 클라이언트 및 Python 서버\)](#) (p. 139)에 설명된 HTML5 클라이언트용 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 ECMA Script 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+
    */
  </script>
</head>
```

```
* - 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>
```

## Python 예제: Python 서버 코드(server.py)

이 단원에서는 [Python 예제\(HTML5 클라이언트 및 Python 서버\)](#) (p. 139)에 설명된 Python 서버용 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)
        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 보여줍니다. 예제에서는 [Java용 AWS SDK](#)를 사용하여 목록에서 선택한 음성으로 지정된 텍스트를 읽습니다.

표시된 코드는 주요 작업에 해당하지만 사소한 오류 확인도 수행합니다. Amazon Polly에 오류가 생기면 애플리케이션이 종료됩니다.

이 예제 애플리케이션을 실행하려면 다음 사항이 필요합니다.

- Java 8 JDK(Java Development Kit)
- [Java용 AWS SDK](#)
- [Apache Maven](#)

애플리케이션을 테스트하려면

1. JAVA\_HOME 환경 변수가 JDK에 맞게 설정되어 있어야 합니다.

예를 들어 JDK 1.8.0\_121을 Windows의 C:\Program Files\Java\jdk1.8.0\_121에 설치한 경우 명령 프롬프트에 다음을 입력합니다.

```
set JAVA_HOME="C:\Program Files\Java\jdk1.8.0_121"
```

JDK 1.8.0\_121을 Linux의 /usr/lib/jvm/java8-openjdk-amd64에 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, Maven 3.3.9를 Windows의 C:\Program Files\apache-maven-3.3.9에 설치한 경우 다음을 입력합니다.

```
set M2_HOME="C:\Program Files\apache-maven-3.3.9"
set M2=%M2_HOME%\bin
set PATH=%M2%;%PATH%
```

Maven 3.3.9를 Linux의 /home/ec2-user/opt/apache-maven-3.3.9에 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-demo 인 디렉터리를 새로 생성합니다.
4. polly-java-demo 디렉터리에 이름이 pom.xml 인 새 파일을 만들고 다음 코드를 파일에 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. 이름이 polly인 새 디렉터리를 src/main/java/com/amazonaws/demos에 생성합니다.
6. polly 디렉터리에 이름이 PollyDemo.java인 새 Java 원본 파일을 만들고 다음 코드를 파일에 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 javax.swing.plaf.basic.AdvancedPlayer;
import javax.swing.plaf.basic.PlaybackEvent;
```

```
import javax.speech.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);
        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,
            javax.speech.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 돌아가 데모를 지우고, 컴파일하며, 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 작업에 해당하지만 오류는 처리하지 않습니다. 전체 코드는 [AWS SDK for iOS Amazon Polly demo](#) 항목을 참조하십시오.

### 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
})
```

## Android 예제

다음 예제에서는 Amazon Polly용 Android SDK를 사용하여 음성 목록에서 선택한 음성으로 지정된 텍스트를 읽습니다.

여기에 표시된 코드는 주요 작업에 해당하지만 오류는 처리하지 않습니다. 전체 코드는 [AWS SDK for Android Amazon Polly demo](#) 항목을 참조하십시오.

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 스트림 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);
```

#### 합성된 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();
    }
});
```

# Windows SAPI 규정 준수 애플리케이션용 Amazon Polly

Amazon Polly는 Windows용 Amazon Polly 플러그인을 사용하여 SAPI(Speech Application Programming Interface) 규정 준수 Windows 애플리케이션을 위해 대폭 확장된 스피치 기능을 제공합니다. SAPI는 데스크톱 애플리케이션에서 스피치를 생성할 수 있는 Microsoft Windows API입니다.

Windows는 SAPI API에 함께 사용할 수 있는 남성 음성 1개와 여성 음성 1개를 제공합니다. Windows용 Amazon Polly 플러그인을 사용하면 Windows 애플리케이션이 AWS 클라이언트가 되므로 표준 음성과 신경 음성을 포함한 모든 Amazon Polly 음성과 뉴스 진행자 말투를 사용할 수 있습니다. Windows용 Amazon Polly 플러그인은 64비트 SAPI 애플리케이션만 지원합니다.

Windows용 Amazon Polly는 SSML(Speech Synthesis Markup Language)과 같은 모든 Amazon Polly 기능을 지원합니다. SSML을 플러그인과 함께 사용하면 표준 기능을 사용할 수 있고 표준 제한이 적용됩니다. SAML을 Amazon Polly와 함께 사용하는 방법에 대한 자세한 내용은 [SSML 문서로부터 스피치 생성 \(p. 84\)](#) 단원을 참조하십시오. 제한에 대한 자세한 내용은 [Amazon Polly의 할당량 \(p. 167\)](#) 단원을 참조하십시오.

## 주제

- [Windows용 Amazon Polly\(SAPI\) 설치 및 구성 \(p. 154\)](#)
- [애플리케이션에서 Amazon Polly 사용 \(p. 156\)](#)

## Windows용 Amazon Polly(SAPI) 설치 및 구성

Windows용 Amazon Polly(SAPI) 플러그인을 설치하고 구성하려면 AWS 계정이 필요합니다. 계정이 없는 경우 [1.1단계: AWS에 가입 \(p. 4\)](#) 단원을 참조하십시오.

## 주제

- [AWS 클라이언트용 IAM 사용자 생성 \(p. 154\)](#)
- [Windows용 AWS CLI 설치 \(p. 155\)](#)
- [AWS 클라이언트용 프로필 생성 \(p. 155\)](#)
- [Windows용 Amazon Polly 플러그인 설치 \(p. 155\)](#)

## AWS 클라이언트용 IAM 사용자 생성

AWS 클라이언트를 AWS 계정에 연결하려면 먼저 클라이언트용 IAM(Identity and Access Management) 사용자를 생성한 다음 권한 정책을 해당 사용자에게 연결해야 합니다. IAM 사용자는 AWS 제품에 대한 API 호출을 수행할 수 있는 AWS 계정에 속한 사람 또는 애플리케이션입니다.

AWS 클라이언트는 Amazon Polly 서비스를 사용하므로 `AmazonPollyReadOnlyAccess` 권한 정책이 필요합니다.

### IAM 사용자를 생성하려면

1. AWS Management Console에 로그인하고 [IAM 콘솔](#)을 엽니다.
2. 사용자를 선택합니다.
3. 사용자 추가를 선택합니다.
4. 사용자 이름에 `polly-windows-user`를 입력합니다.
5. Access Type(액세스 유형)에서 Programmatic access(프로그래밍 방식 액세스)를 선택한 후 Next: Permissions(다음: 권한)를 선택합니다.



6. Attach existing policies directly(기존 정책 직접 연결)를 선택합니다.
7. 검색 상자에 polly를 입력합니다.
8. AmazonPollyReadOnlyAccess를 선택합니다.
9. Next: Tags(다음: 태그)를 선택합니다.
10. Next: Review(다음: 검토)를 선택합니다.
11. Create User를 선택합니다.
12. 액세스 키 ID 및 보안 액세스 키를 기록해 둡니다. 이러한 키는 AWS 클라이언트를 구성할 때 필요합니다.

#### Important

이번 한 번만 이러한 키에 액세스할 수 있으므로 기록해 두는 것이 좋습니다.

## Windows용 AWS CLI 설치

AWS 명령줄 인터페이스(AWS CLI)는 Windows 명령 프롬프트의 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다. AWS CLI를 설치하려면 AWS 명령줄 인터페이스 사용 설명서에서 [Windows에 AWS CLI 설치](#) 단원을 참조하십시오.

## AWS 클라이언트용 프로필 생성

Windows용 Amazon Polly 플러그인은 polly-windows라는 AWS 프로필이 필요합니다. 이 프로필은 Amazon Polly 엔진이 올바른 계정을 사용하도록 보장합니다.

AWS 클라이언트용 **polly-windows** 프로필을 생성하려면

1. Windows에서 명령 프롬프트를 엽니다.
2. 명령 프롬프트에서 `aws configure --profile polly-windows` 명령을 실행합니다.
3. AWS 액세스 키 ID 및 AWS 보안 액세스 키를 입력하라는 메시지가 나타나면 IAM 사용자를 생성할 때 저장된 값을 입력합니다.
4. 기본 리전에 원하는 AWS 리전 이름을 소문자로 입력합니다.

#### Note

AWS 리전에 따라 일부 Amazon Polly 음성 및 기능을 사용할 수 없습니다. 자세한 내용은 [기능 및 리전 호환성 \(p. 72\)](#) 단원을 참조하십시오.

5. Default output format(기본 출력 형식)에 Enter를 누릅니다.
6. 명령 프롬프트에서 `aws --profile polly-windows polly describe-voices` 명령을 실행하여 프로필을 확인합니다. 프로필이 성공적으로 생성되면 AWS CLI에 이용 가능한 Amazon Polly 이름 목록이 표시됩니다.

## Windows용 Amazon Polly 플러그인 설치

Amazon Simple Storage Service(Amazon S3)에서 다운로드하고 설치 프로그램을 실행하여 Windows용 Amazon Polly 플러그인을 설치합니다. 플러그인을 설치하면 SAPI API를 사용하는 모든 Windows 애플리케이션에서 Amazon Polly 음성을 사용할 수 있습니다.

#### Note

SAPI 애플리케이션에서 신경 음성을 사용하려는 경우, Windows용 플러그인을 설치하기 전에 클라이언트가 NTTS를 지원하는 AWS 리전 용으로 구성되어 있는지 확인하십시오. 자세한 내용은 [기능 및 리전 호환성 \(p. 72\)](#) 단원을 참조하십시오.

#### Windows용 Amazon Polly 플러그인을 설치하려면

1. Windows에서 [플러그인](#)을 다운로드합니다.
2. 폴더를 압축 해제합니다.
3. 실행 AmazonPollyForWindowsSetup.exe.
4. 제어판에서 speech settings를 검색합니다.
5. 텍스트 음성 변환에서 음성에 Amazon Polly 음성을 선택합니다.
6. 음성 미리 보기를 선택합니다.

## 애플리케이션에서 Amazon Polly 사용

Windows용 Amazon Polly 플러그인을 설치하면 Windows SAPI를 구현하는 모든 Windows 애플리케이션에서 Amazon Polly 음성에 액세스할 수 있습니다. 음성을 사용하려면 애플리케이션의 Amazon Polly 음성 목록에서 선택하십시오. 사용 가능한 음성에 대한 전체 목록은 [사용 가능한 음성 \(p. 12\)](#) 단원을 참조하십시오.

Windows 애플리케이션(Adobe Captivate)에서 Amazon Polly 음성을 사용하는 방법에 대한 예는 AWS 머신러닝 블로그에서 [Windows 애플리케이션에서 Amazon Polly 사용](#) 단원을 참조하십시오.

Windows 애플리케이션에서 Amazon Polly 음성을 사용하려면 PollyPlayer를 사용하면 됩니다. PollyPlayer는 Windows용 Amazon Polly 플러그인을 설치할 때 설치되는 간단한 Windows 스피치 합성 애플리케이션입니다.

#### PollyPlayer를 사용하려면

1. Windows 시작 메뉴의 Windows용 Amazon Polly에서 PollyPlayer를 선택합니다.
2. PollyPlayer 애플리케이션에서 음성을 선택한 다음 듣고 싶은 텍스트를 입력합니다.



PollyPlayer

## Pick a Voice:

Amazon Polly - Arabic - Zeina (Standard)  
Amazon Polly - Australian English - Nicole (Standard)  
Amazon Polly - Australian English - Russell (Standard)  
Amazon Polly - Brazilian Portuguese - Ricardo (Standard)  
Amazon Polly - Brazilian Portuguese - Vitoria (Standard)  
Amazon Polly - British English - Amy (Neural)  
Amazon Polly - British English - Amy (Standard)  
Amazon Polly - British English - Brian (Neural)  
Amazon Polly - British English - Brian (Standard)  
Amazon Polly - British English - Emma (Neural)  
Amazon Polly - British English - Emma (Standard)  
Amazon Polly - Canadian French - Chantal (Standard)  
Amazon Polly - Castilian Spanish - Conchita (Standard)  
Amazon Polly - Castilian Spanish - Enrique (Standard)  
Amazon Polly - Castilian Spanish - Lucia (Standard)  
Amazon Polly - Chinese Mandarin - Zhiyu (Standard)  
Amazon Polly - Danish - Mads (Standard)  
Amazon Polly - Danish - Naja (Standard)  
Amazon Polly - Dutch - Lotte (Standard)  
Amazon Polly - Dutch - Ruben (Standard)  
Amazon Polly - French - Celine (Standard)  
Amazon Polly - French - Lea (Standard)  
Amazon Polly - French - Mathieu (Standard)  
Amazon Polly - German - Hans (Standard)

3. 텍스트를 들으려면 Say It(말하세요)을 선택합니다.

# AWS for WordPress 플러그인

AWS for WordPress 플러그인을 사용하면 Amazon Polly를 비롯한 여러 AWS 서비스를 설정할 수 있습니다. Amazon Polly를 사용하면 WordPress 웹 사이트 방문자에게 사용자 콘텐츠의 오디오 레코딩을 제공할 수 있습니다. 이 플러그인을 사용하면 Amazon Polly에서 지원하는 모든 음성 및 언어로 오디오 파일을 생성할 수 있습니다. 사용자의 방문자는 인라인 오디오 플레이어 및 모바일 애플리케이션을 사용하여 편리하게 오디오를 스트리밍할 수 있습니다.

사용자는 다음을 수행하도록 플러그인을 구성할 수 있습니다.

- 게시할 때 새 콘텐츠의 오디오 레코딩 자동 생성 또는 개별 게시물에 대한 레코딩을 생성하도록 선택
- 아카이브된 콘텐츠의 자동 레코딩 생성
- Amazon Pollycast RSS 피드를 사용하여 오디오 콘텐츠 팟캐스트

## 주제

- [설치를 위한 사전 조건 \(p. 159\)](#)
- [플러그인 설치 및 구성 \(p. 161\)](#)
- [WordPress 페이지 사용자 지정 \(p. 162\)](#)
- [Amazon Pollycast로 팟캐스트 \(p. 164\)](#)
- [오디오 파일 저장 \(p. 165\)](#)

## 설치를 위한 사전 조건

AWS for WordPress 플러그인을 사용하려면 AWS 계정, AWS Identity and Access Management(IAM) 사용자 및 WordPress 웹 사이트가 필요합니다.

## 주제

- [AWS 계정 생성 \(p. 159\)](#)
- [IAM 사용자 생성 \(p. 159\)](#)
- [WordPress 웹 사이트 만들기 \(p. 161\)](#)

## AWS 계정 생성

AWS 계정이 이미 있으면 이 단원을 건너뛰어도 됩니다. 계정이 없으면 만드십시오.

AWS 계정을 만들려면 다음을 수행합니다.

1. <https://portal.aws.amazon.com/billing/signup>을 엽니다.
2. 온라인 지시 사항을 따릅니다.

등록 절차 중 전화를 받고 전화 키패드를 사용하여 확인 코드를 입력하는 과정이 있습니다.

## IAM 사용자 생성

AWS for WordPress 플러그인을 사용하려면 플러그인의 IAM 사용자를 만들어야 합니다. IAM 사용자는 AWS 서비스에 대한 API 호출을 수행할 권한이 있는 AWS 계정에 속한 사람 또는 애플리케이션입니다.

#### Note

[WordPress.com](https://wordpress.com)을 사용하지 않고 Amazon EC2에 자체 호스팅되는 WordPress 웹 사이트가 있으면 IAM 사용자 대신 IAM 역할을 사용해도 됩니다. 자세한 내용은 Amazon EC2 사용 설명서의 [Amazon EC2에 대한 IAM 역할](#)을 참조하십시오.

다음 절차에는 IAM 정책을 만들어 IAM 사용자에게 연결하는 단계가 있습니다. IAM 정책은 사용자에게 적용되는 권한을 정의하는 문서입니다.

#### IAM 사용자를 생성하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 정책을 선택합니다. 그런 다음 정책 생성을 선택합니다.
3. JSON을 선택합니다.
4. 정책 텍스트 상자의 내용을 모두 삭제한 후 다음 JSON 정책을 텍스트 상자에 붙여 넣거나 입력합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "acm:DeleteCertificate",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestedRegion": "us-east-1"
        }
      }
    }
  ]
}
```

5. 정책 검토를 선택합니다.
6. 정책 검토 페이지에서 다음을 수행합니다.
  - a. 이름에 **AWSForWordPressDeleteCert**를 입력합니다.
  - b. 정책 생성을 선택합니다.
7. 탐색 창에서 사용자를 선택합니다. 그런 다음 Add user(사용자 추가)를 선택합니다.
8. Set user details(사용자 세부 정보 설정) 페이지에서 다음을 수행합니다.
  - a. 사용자 이름에 **AWSForWordPressPlugin**을 입력합니다.
  - b. Access type(액세스 유형)에서 Programmatic access(프로그래밍 방식 액세스)를 선택합니다.
  - c. Next: Permissions(다음: 권한)를 선택합니다.
9. 권한 설정 페이지에서 다음을 수행합니다.
  - a. Attach existing policies directly(기존 정책 직접 연결)를 선택합니다.
  - b. 검색 상자에 **WordPress**를 입력한 후 AWSForWordPressPolicy와 AWSForWordPressDeleteCert 옆의 확인란을 선택합니다. 두 WordPress 정책의 확인란을 모두 선택해야 합니다..

#### Note

AWSForWordPressPolicy는 AWS 관리형 정책이며 사용자에게 AWS for WordPress 플러그인에 있는 기능을 모두 사용할 수 있는 권한을 부여합니다. 플러그인에 새 기능이 추가되면 AWS가 새 기능을 사용하는 데 필요한 권한을 포함하도록 이 정책을 업데이트합니다.

- c. 다음: 태그를 선택합니다.
10. [Next: Review]를 선택합니다.

11. Create user를 선택합니다.
12. Download .csv(.csv 다운로드)를 선택하여 사용자의 자격 증명(액세스 키 ID 및 보안 액세스 키)을 컴퓨터에 저장합니다. 이는 AWS for WordPress 플러그인을 구성할 때 필요합니다.

#### Important

여기서만 사용자의 보안 액세스 키를 저장할 수 있으므로 지금 저장해야 합니다.

## WordPress 웹 사이트 만들기

이미 WordPress 웹 사이트가 있으면 [플러그인 설치 및 구성 \(p. 161\)](#) 단원으로 건너뛰어도 좋습니다.

WordPress 웹 사이트가 없으면 [WordPress.com](#)을 사용하여 만들 수 있습니다. AWS for WordPress 플러그인을 사용하려면 WordPress.com Business 또는 eCommerce 플랜이 필요합니다.

Amazon Lightsail, Amazon EC2 또는 다른 웹 호스팅 플랫폼을 사용하여 자체 웹 서버에 WordPress 소프트웨어를 설치할 수도 있습니다. 자체 WordPress 웹 사이트를 호스팅하면 WordPress.com을 사용할 때보다 많은 단계가 포함되며 웹 서버, 로드 밸런서, DNS 레코드 및 웹 서버 인증서를 구성하고 관리할 수 있어야 합니다.

WordPress 웹 사이트 설정 방법에 관계없이 AWS for WordPress 플러그인을 사용하려면 다음 사항이 필요합니다.

- 웹 사이트에 고유한 도메인 이름이 있어야 합니다. 웹 주소나 URL(uniform resource locator)이라고도 하는 도메인 이름은 방문자가 웹 사이트에 갈 때 사용하는 주소입니다. 예를 들어 Amazon의 도메인 이름은 amazon.com입니다. 이 주제에서는 example.com을 도메인 이름의 일반적인 예로 사용하지만 특정 웹 사이트에는 그에 맞는 도메인 이름이 필요합니다.
- HTTPS를 사용하여 웹 사이트가 작동해야 합니다. 이 사항은 보안 모범 사례이며 플러그인은 웹 사이트가 HTTPS를 사용하여 작동한다고 가정합니다. 확인하려면 HTTPS를 사용하는 웹 사이트 주소(예: https://example.com)로 이동하여 웹 사이트가 제대로 표시되는지 확인하십시오.

웹 사이트에 도메인 이름이 있고 웹 사이트가 HTTPS를 사용하여 작동하는 경우 다음 단원으로 진행하십시오.

## 플러그인 설치 및 구성

플러그인을 설치하기 전에 [사전 조건 \(p. 159\)](#)을 모두 갖추어야 합니다.

플러그인을 설치하려면

1. WordPress 웹 사이트 관리 대시보드(WP Admin)에 로그인합니다.
2. Plugins(플러그인)를 선택합니다.
3. 새로 추가를 선택합니다.
4. 검색 상자에 **AWS for WordPress**를 입력합니다.
5. AWS for WordPress 플러그인을 찾습니다. Install Now(지금 설치)를 선택한 후 활성화를 선택합니다.

## 플러그인 구성

AWS for WordPress 플러그인을 설치한 후에는 팟캐스팅, 대체 스토리지 위치 및 기타 옵션을 활성화하도록 구성합니다.

#### Note

다음 절차에서 명령 및 필드 이름은 WordPress에서 사용된 이름과 조금 다를 수 있습니다.

## 플러그인을 구성하려면

1. Import WordPress(WordPress 가져오기) 페이지에서 Settings(설정)을 선택합니다.
2. 다음 옵션을 사용하여 플러그인을 구성합니다.
  - AWS access key and AWS secret key(AWS 액세스 키 및 AWS 보안 키) — 이 AWS 자격 증명을 통해 Amazon Polly 및 Amazon Simple Storage Service(Amazon S3)를 사용할 수 있습니다. 앞에서 생성한 AWS 액세스 및 보안 키를 입력합니다. Amazon EC2에서 WordPress 사이트를 호스팅하는 경우 자격 증명 대신 IAM 역할을 사용할 수 있습니다. 이 경우 이러한 두 필드를 공백 상태로 남겨둡니다.
  - 샘플 비율 — 생성할 오디오 파일의 샘플 비율(Hz)입니다. 샘플 비율이 높을수록 오디오 품질이 높아 집니다.
  - Voice name(음성 이름) — 오디오 파일에서 사용할 Amazon Polly 음성입니다.

Note

음성 선택은 사용 중인 언어에 따라 결정됩니다. 사용 가능한 전체 음성은 [사용 가능한 음성 \(p. 12\)](#) 단원을 참조하십시오.
  - 플레이어 위치—웹 사이트에서의 오디오 플레이어 위치입니다. 게시물의 앞이나 뒤에 넣거나 전혀 사용하지 않을 수 있습니다. Amazon Pollycast를 사용하여 파일을 팟캐스트로 제공하려면 오디오 플레이어 표시하지 마십시오.
  - 새 게시 기본값—Amazon Polly가 모든 새 게시물에 대한 오디오 파일을 자동으로 생성해야 할지 여부를 지정합니다. Amazon Polly에서 새 게시물마다 오디오 파일을 만들려면 이 옵션을 선택합니다.
  - 자동 재생—방문자가 방문할 때 오디오 플레이어에서 게시물에 대한 오디오 재생을 자동으로 시작할지 여부를 지정합니다.
  - Store audio in Amazon S3(Amazon S3에 오디오 저장) — 오디오 파일을 웹 서버가 아닌 S3 버킷에 저장하려면 이 옵션을 선택합니다. Amazon Polly에서 자동으로 버킷을 만듭니다. 자세한 내용 및 요금은 [Amazon S3](#)를 참조하십시오.
  - Amazon CloudFront (CDN) domain name(Amazon CloudFront(CDN) 도메인 이름) — Amazon CloudFront를 사용하여 오디오 파일을 브로드캐스트하려면 CloudFront 도메인의 이름을 제공합니다. 플러그인은 도메인을 사용하여 오디오를 스트리밍합니다. 아직 도메인이 없는 경우 Amazon CloudFront에서 도메인을 생성합니다.
  - iTunes category(ITunes 범주) — 팟캐스트 범주입니다. 범주를 선택하면 팟캐스트 사용자가 팟캐스트 범주에서 팟캐스트를 쉽게 찾을 수 있습니다.
  - iTunes 명시적 허용—iTunes 명시적 허용 설정을 사용하여 Amazon Pollycast 팟캐스트를 사용할지 여부를 지정합니다.
  - 모든 게시물 대량 업데이트—이전 WordPress 게시물을 모두 변환하여 AWS for WordPress 플러그인을 사용하려는 경우 이 옵션을 선택합니다.
3. 변경 사항 저장을 선택합니다.

## WordPress 페이지 사용자 지정

WordPress 콘텐츠를 사용자 지정하여 Amazon Polly WordPress 플러그인과 더 효과적으로 호환될 수 있도록 몇 가지 옵션을 사용할 수 있습니다.

- 오디오 파일 미세 조정을 위한 플러그인 설정의 조정 (p. 162)
- 콘텐츠에서 SSML을 사용하여 읽혀지는 방식을 수정합니다. (p. 163)
- 콘텐츠에서 오디오 전용 및 워드 전용 태그 사용 (p. 163)

## 오디오 파일 미세 조정을 위한 플러그인 설정의 조정

AWS for WordPress 플러그인 설정에는 오디오 파일을 위해 WordPress 텍스트의 사운드를 사용자 지정하는 데 유용한 세 가지 옵션이 있습니다.



- 음성 이름: 음성 이름과 언어를 선택함으로써 Amazon Polly 음성의 성별을 선택할 수 있습니다. 각 언어별로 특정 음성을 사용할 수 있으며 다수의 언어에서 각 성별 내에서도 몇 개의 옵션을 제공합니다. 자세한 내용은 [Amazon Polly의 음성 \(p. 12\)](#) 단원을 참조하십시오.
- 자동화된 숨소리: 활성화된 경우 Amazon Polly에서 콘텐츠에 숨소리 잡음을 적절한 시점을 골라 자동으로 추가합니다. 이 옵션을 설정하면 자동화된 숨소리만을 사용할 수 있고 숨소리를 수동으로 설정할 수 없습니다. 자세한 내용은 [자동화된 숨소리](#)를 참조하십시오.
- 오디오 속도: 이 옵션을 통해 콘텐츠의 오디오 버전의 전달 속도를 기본 음성 속도의 20%~200% 범위 내에서 변경할 수 있습니다.

## 콘텐츠에서 SSML을 사용하여 읽혀지는 방식을 수정합니다.

Amazon Polly는 사용자가 제공한 텍스트로부터 Amazon Polly에 의해 스피치를 생성하는 방식의 많은 측면을 제어 가능하게 해 주는 여러 SSML 태그를 자체적으로 지원합니다. SSML 및 Amazon Polly에 대한 자세한 내용은 [지원되는 SSML 태그 \(p. 91\)](#) 단원을 참조하십시오. 이러한 태그 중 일부는 사용자의 설정을 구성할 때 설정 내의 플러그인에 반영됩니다. 그러나 현재 AWS for WordPress 플러그인에 직접 사용할 수 있는 것은 <break> 태그가 유일합니다.

<break> 태그를 사용해 텍스트의 음성 버전에 일시 중지를 추가할 수 있습니다. 사용자의 특정 필요에 맞추어 이러한 일시 중지의 길이를 조정할 수 있습니다. 일시 중지의 기본 길이는 첨부 다음에 오는 일시 중지에 해당합니다. <break> 태그에 대한 자세한 내용은 [지원되는 SSML 태그 \(p. 91\)](#)을 참조하십시오.

SSML 태그를 사용하여 WordPress 텍스트를 확장하려면, WordPress Admin(WordPress 관리자) 페이지에 있는 Amazon Polly 설정에서 SSML 지원 활성화 옵션을 선택해야 합니다. SSML 태그가 사용된 오디오 파일은 S3 버킷에 보관해야 하기 때문에 Store audio in Amazon S3 option(Amazon S3에 오디오 저장 옵션) 또한 선택해야 합니다.

## 콘텐츠에서 오디오 전용 및 워드 전용 태그 사용

오디오 팟캐스트에 무언가 추가하고 싶지만 브라우저에 추가 사항이 표시되길 원치 않을 수 있습니다. 또는 브라우저를 통해 무언가를 보여주고 싶지만 오디오 파일에 포함되는 것은 원치 않을 수 있습니다. 이런 문제는 Audio Only 및 Word Only 태그를 WordPress 콘텐츠에 배치하여 해당 텍스트의 어떤 부분을 문자로 표시하거나 읽어줄지 제어함으로써 해결할 수 있습니다.

텍스트를 오디오로 전환하되 브라우저에서 보여주지 않으려면

1. 선택된 텍스트의 위아래에 비어있는 행을 추가하여 WordPress 페이지로부터 분리시킵니다.
2. 선택된 텍스트의 위쪽 행에는 다음과 같은 태그를 삽입합니다.

-AMAZONPOLLY-ONLYAUDIO-START-

3. 선택된 텍스트의 아래쪽 행에는 다음과 같은 태그를 삽입합니다.

-AMAZONPOLLY-ONLYAUDIO-END-

-AMAZONPOLLY-ONLYWORDS-START- 및 -AMAZONPOLLY-ONLYWORDS-END- 태그를 동일한 방식으로 사용하면 텍스트를 브라우저에서 보여주고 오디오 파일에는 포함시키지 않는 동일한 절차를 활용하는 샘플입니다.

예:

```
Initial text of your blog displayed in the browser and heard in the audio file.]
-AMAZONPOLLY-ONLYAUDIO-START-
[This part will not be displayed in the browser but will be heard in the audio file.]
-AMAZONPOLLY-ONLYAUDIO-END-
```

[Subsequent text of your blog displayed in the browser and heard in the audio file.]

그리고

[Initial text of your blog displayed in the browser and heard in the audio file.]  
-AMAZONPOLLY-ONLYWORD-START-  
This part will be displayed in the browser but will not be heard in the audio file.]  
-AMAZONPOLLY-ONLYWORD-END-  
Subsequent text of your blog displayed in the browser and heard in the audio file.

## 게시물에 번역된 텍스트 추가

AWS for WordPress 플러그인은 Amazon Translate를 사용하여 게시물을 1개 이상의 다른 언어로 번역한 버전을 생성합니다. 이 서비스에는 영어 외에도 스페인어, 프랑스어, 독일어, 포르투갈어 등 4개국어가 더 제공됩니다. Amazon Translate 구성에 있는 설정 페이지에서 사용할 언어와 그 언어에 해당하는 음성을 구성할 수 있습니다.

WordPress 게시물을 다른 언어로 번역하려면

1. Add New Post(새로운 게시물 추가) 페이지에서 새로운 WordPress 게시물을 생성하여 게시합니다.
2. 동일한 페이지에서 반드시 Amazon Polly 활성화 옵션을 선택해야 합니다.
3. 원래 사용된 언어 및 추가로 선택한 언어로 오디오 파일을 만드는 데 소요되는 대략적인 비용을 보려면 How much will this cost to convert?(오디오 변환 비용 견적)를 선택합니다. OK(확인)를 선택하여 Add New Post(새 게시물 추가) 페이지로 복귀합니다.
4. [Translate]를 선택합니다.

게시물을 번역할 언어를 설정하려면

1. [Amazon Translate configuration] 아래에 있는 [Settings] 페이지에서 [Source language] 드롭다운 목록으로부터 게시물 작성에 사용된 언어를 선택합니다.
2. [Target languages] 아래에서 게시물을 번역할 언어를 선택합니다.
3. [Voice] 드롭다운 목록에서 선택된 각 언어별로 사용할 음성을 선택합니다.
4. 언어 선택에 관한 레이블을 입력합니다.
5. 변경 사항 저장을 선택합니다.

## Amazon Pollycast로 팟캐스트

Amazon Pollycast 피드를 사용하면 방문자가 표준 팟캐스트 애플리케이션을 사용하여 사용자의 오디오 콘텐츠를 들을 수 있습니다. RSS 2.0 호환 Pollycast 피드는 iTunes와 같은 인기 있는 모바일 팟캐스트 애플리케이션 및 팟캐스트 디렉터리별로 팟캐스트를 집계하는 데 필요한 XML 데이터를 제공합니다.

AWS for WordPress 플러그인을 설치할 때 자동으로 Amazon Pollycast 엔드포인트를 모든 WordPress 아카이브 URL에 추가하는 iTunes 명시적 허용 옵션을 선택합니다. 이를 통해 사이트 전체 또는 대상 팟캐스트를 합성할 수 있습니다. 플러그인을 설치할 때 iTunes 명시적 허용 옵션을 선택하지 않은 경우 다음 단계를 수행합니다.

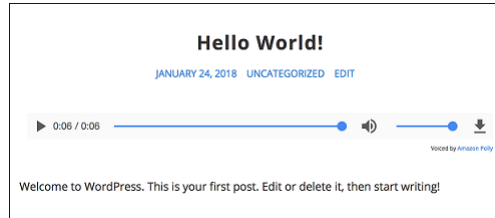
1. Import WordPress(WordPress 가져오기) 페이지에서 Settings(설정)을 선택합니다.
2. Settings(설정) 페이지에서 iTunes explicit(ITunes 명시적 허용)을 선택합니다.

팟캐스트 애플리케이션에서 페이지의 URL에 /amazon-pollycast/를 추가하여 Amazon Pollycast 엔드포인트를 수동으로 추가할 수 있습니다. 예:

```
example.com/amazon-pollycast/  
example.com/category/news/amazon-pollycast/  
example.com/author/john/amazon-pollycast/
```

## 플레이어 위치 지정

AWS for WordPress 플러그인을 설치할 때 오디오 플레이어를 사이트 텍스트 아래에 표시하거나 표시하지 않도록 선택하지 않은 경우 오디오 플레이어는 기본적으로 WordPress 웹 사이트 상단에 표시됩니다.



언제든지 이 플레이어를 재배치하거나, 제거하거나, 추가(표시하지 않도록 선택한 경우)할 수 있습니다.

이 플레이어를 재배치하거나, 제거하거나, WordPress 웹 사이트에 추가하려면

1. Import WordPress(WordPress 가져오기) 페이지에서 Settings(설정)을 선택합니다.
2. Amazon Polly 설정 페이지의 Player position(플레이어 위치)에 대해 적합한 옵션을 선택합니다.

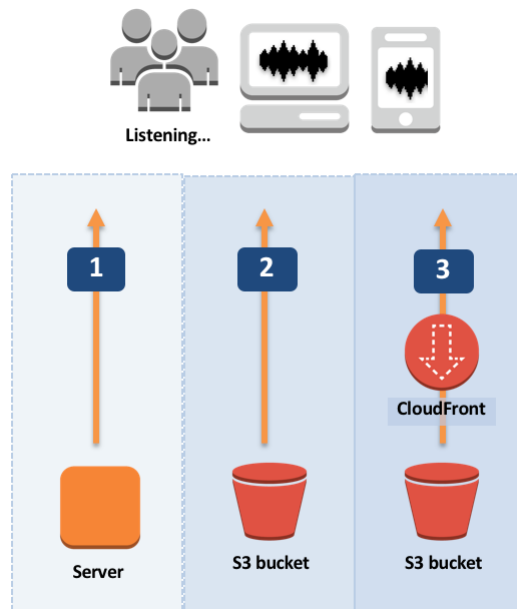
설정 구성 옵션에 대한 자세한 내용은 [플러그인 구성 \(p. 161\)](#) 단원을 참조하십시오.

## 오디오 파일 저장

사이트에 콘텐츠를 게시하면 합성을 위해 Amazon Polly로 전송됩니다. 기본적으로 Amazon Polly에서 새 오디오 파일은 웹 서버에 저장합니다. 또한 파일을 Amazon Simple Storage Service(Amazon S3) 또는 Amazon CloudFront(글로벌 CDN(콘텐츠 전송 네트워크))에 저장할 수 있습니다.

사용자는 오디오 파일을 저장하는 위치와 상관없이 동일한 청취 경험을 갖습니다. 브로드캐스트 위치만 변경됩니다.

1. WordPress 서버에 저장된 오디오 파일의 경우 파일은 서버에서 직접 브로드캐스트됩니다.
2. S3 버킷에 저장된 파일의 경우 해당 버킷에서 브로드캐스트됩니다.
3. CloudFront를 사용하는 경우 파일은 Amazon S3에 저장되고 CloudFront와 함께 브로드캐스트됩니다.



Amazon Polly 플러그인을 설치할 때 파일을 저장할 위치를 선택할 수 있습니다.

# Amazon Polly의 할당량

다음은 Amazon Polly를 사용할 때 유의할 제한 사항입니다.

## 지원하는 리전

Amazon Polly가 지원되는 AWS 리전 목록은 Amazon Web Services 일반 참조의 [Amazon Polly 엔드포인트 및 할당량](#)을 참조하십시오.

현재 신경 음성이 지원되는 리전은 다음과 같습니다.

- 미국 동부(버지니아 북부): us-east-1
- 미국 서부(오레곤): us-west-2
- EU(아일랜드): eu-west-1
- 아시아 태평양(시드니): ap-southeast-2

이 리전의 엔드포인트와 프로토콜은 표준 음성 사용에 사용되는 것과 같습니다.

## Throttling

- 계정당 조절 속도: 100건의 tps(초당 트랜잭션, 요청 및 작업) 및 버스트 한도 120tps입니다.

계정당 동시 연결 수: 90

- 작업당 조절 속도:

작업	한도
어휘	
DeleteLexicon	이 작업이 완료된 후 2건의 초당 트랜잭션(tps)입니다. 허용되는 최대 버스트는 4tps입니다.
PutLexicon	
GetLexicon	
ListLexicons	
음성	
DescribeVoices	80tps 및 버스트 한도 100tps
SynthesizeSpeech	표준 음성: 80tps 및 버스트 한도 100tps 신경 음성: 8tps 및 버스트 한도 10tps
StartSpeechSynthesisTask	표준 음성: 10tps 및 버스트 한도 12tps 신경 음성: 1tps

작업	한도
GetSynthesizeSpeechTask 및 ListSynthesizeSpeechTask	최대 10tps 결합 허용

## 발음 어휘

- 계정당 어휘를 최대 100개까지 저장할 수 있습니다.
- 어휘 이름은 최대 20자의 영숫자 문자열로 지정할 수 있습니다.
- 각 어휘의 최대 크기는 4,000자입니다. 어휘의 크기는 SynthesizeSpeech 작업의 지연 시간에 영향을 줍니다.
- 어휘의 <phoneme> 또는 <alias> 대체어마다 100자까지 지정할 수 있습니다.

어휘를 사용하는 방법에 대한 자세한 내용은 [어휘 관리 \(p. 112\)](#) 단원을 참조하십시오.

## SynthesizeSpeech API 작업

SynthesizeSpeech API 작업 사용과 관련하여 다음 제한 사항에 유의하십시오.

- 청구되는 입력 텍스트의 최대 크기는 3000자(총 6000자)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 [SynthesizeSpeech \(p. 211\)](#) 단원을 참조하십시오.

### Note

SynthesizeSpeech API 작업의 몇 가지 제한을 StartSynthesizeSpeechTask API 작업을 사용하여 우회할 수 있습니다. 자세한 내용은 [긴 오디오 파일 생성 \(p. 123\)](#) 단원을 참조하십시오.

## SpeechSynthesisTask API 작업

StartSpeechSynthesisTask, GetSpeechSynthesisTask 및 ListSpeechSynthesisTasks API 작업 사용과 관련하여 다음 제한 사항에 유의하십시오.

- 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 유의하십시오.

- <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 문서로부터 스피치 생성 \(p. 84\)](#) 단원을 참조하십시오.

# Amazon Polly의 보안

AWS에서는 클라우드 보안을 가장 중요하게 생각합니다. AWS 고객은 보안에 매우 보안에 민감한 조직의 요구 사항에 부합하도록 구축된 데이터 센터 및 네트워크 아키텍처의 혜택을 누릴 수 있습니다.

보안은 AWS와 귀하의 공동 책임입니다. [공동 책임 모델](#)은 이 사항을 클라우드의 보안 및 클라우드 내 보안으로 설명합니다.

- 클라우드의 보안 – AWS는 AWS 클라우드에서 AWS 서비스를 실행하는 인프라를 보호합니다. AWS는 또한 안전하게 사용할 수 있는 서비스를 제공합니다. 타사 감사자는 [AWS 규정 준수 프로그램](#)의 일환으로 보안 효과를 정기적으로 테스트하고 검증합니다. Amazon Polly에 적용되는 규정 준수 프로그램에 대한 자세한 내용은 [규정 준수 프로그램 제공 AWS 범위 내 서비스](#)를 참조하십시오.
- 클라우드 내 보안 – 귀하의 책임은 귀하가 사용하는 AWS 서비스에 의해 결정됩니다. 또한 귀하는 데이터의 민감도, 회사 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

이 설명서는 Amazon Polly 사용 시 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 방법을 배우게 됩니다.

## 주제

- [Amazon Polly의 데이터 보호 \(p. 170\)](#)
- [Amazon Polly의 자격 증명 및 액세스 관리 \(p. 171\)](#)
- [Amazon Polly의 로깅 및 모니터링 \(p. 181\)](#)
- [Amazon Polly 규정 준수 확인 \(p. 182\)](#)
- [Amazon Polly의 복원성 \(p. 182\)](#)
- [Amazon Polly의 인프라 보안 \(p. 182\)](#)
- [Amazon Polly의 보안 모범 사례 \(p. 183\)](#)

## Amazon Polly의 데이터 보호

Amazon Polly은 AWS [공동 책임 모델](#)을 준수하며, 여기에는 데이터 보호 관련 규정 및 지침이 포함됩니다. AWS는 모든 AWS 제품을 실행하는 글로벌 인프라를 보호할 책임을 갖습니다. AWS는 고객 콘텐츠 및 개인 데이터의 처리를 위한 보안 구성 제어 등 이 인프라에서 호스팅되는 데이터에 대한 제어권을 유지합니다. 데이터 제어자 또는 데이터 처리자의 역할을 담당하는 AWS 고객과 APN 파트너는 AWS 클라우드에 올린 모든 개인 데이터에 대한 책임을 갖습니다.

데이터 보호를 위해 AWS 계정 자격 증명을 보호하고 AWS Identity and Access Management (IAM)을 사용하여 개별 사용자 계정을 설정하여 각 사용자에게 직무를 수행하는 데 필요한 권한만 부여하는 것이 좋습니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정마다 멀티 팩터 인증(MFA)을 사용합니다.
- SSL/TLS를 사용하여 AWS 리소스와 통신합니다.
- AWS CloudTrail로 API 및 사용자 활동 로깅을 설정합니다.
- AWS 암호화 솔루션을 AWS 서비스 내의 모든 기본 보안 컨트롤과 함께 사용합니다.

이름 필드와 같은 자유 형식 필드에 고객 계정 번호와 같은 중요 식별 정보를 절대 입력하지 마십시오. 여기에는 Amazon Polly 또는 기타 AWS 서비스에서 콘솔, API, AWS CLI 또는 AWS SDK를 사용하여 작업하는 경우가 포함됩니다. Amazon Polly 또는 기타 서비스에 입력하는 모든 데이터는 진단 로그에 포함되도록 선



택할 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 블로그에서 [AWS 공동 책임 모델 및 GDPR](#) 블로그 게시물을 참조하십시오.

## 저장 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" \(p. 211\)](#) 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 Polly은 텍스트 제출의 콘텐츠를 보관하지 않습니다.

## 인터넷워크 트래픽 개인 정보

Amazon Polly에 대한 액세스는 AWS 콘솔, CLI 또는 SDK를 통해 이루어집니다. 통신은 기밀성을 위해 TLS(전송 계층 보안) 세션 암호화를 활용하고 인증 및 무결성을 위해 [디지털 서명](#)을 사용합니다.

# Amazon Polly의 자격 증명 및 액세스 관리

AWS Identity and Access Management(IAM)는 관리자가 AWS 리소스에 대한 액세스를 안전하게 제어할 수 있도록 지원하는 AWS 서비스입니다. IAM 관리자는 Amazon Polly 리소스를 사용하기 위해 인증(로그인) 및 권한 부여(권한 있음)할 수 있는 사람을 제어합니다. IAM은 추가 비용 없이 사용할 수 있는 AWS 서비스입니다.

주제

- [대상 \(p. 171\)](#)
- [자격 증명을 통한 인증 \(p. 172\)](#)
- [정책을 사용하여 액세스 관리 \(p. 173\)](#)
- [Amazon Polly에서 IAM을 사용하는 방법 \(p. 174\)](#)
- [Amazon Polly 자격 증명 기반 정책 예제 \(p. 176\)](#)
- [Amazon Polly API 권한: 작업, 권한 및 리소스 참조 \(p. 180\)](#)

## 대상

AWS Identity and Access Management(IAM)를 사용하는 방법은 Amazon Polly에서 수행하는 작업에 따라 다릅니다.

서비스 사용자 – Amazon Polly 서비스를 사용하여 작업을 수행하는 경우 필요한 자격 증명과 권한을 관리자가 제공합니다. 더 많은 Amazon Polly 기능을 사용하여 작업을 수행하게 되면 추가 권한이 필요할 수 있습니다. 액세스 권한 관리 방식을 이해하면 적절한 권한을 관리자에게 요청할 수 있습니다.

서비스 관리자 – 회사에서 Amazon Polly 리소스를 책임지고 있는 경우 Amazon Polly에 대한 전체 액세스 권한을 가지고 있을 것입니다. 서비스 관리자는 직원이 액세스해야 하는 Amazon Polly 기능과 리소스를 결정합니다. 그런 다음 IAM 관리자에게 요청을 제출하여 서비스 사용자의 권한을 변경합니다. 이 페이지의 정보

를 검토하여 IAM의 기본 개념을 이해하십시오. 회사가 Amazon Polly에서 IAM을 사용하는 방법에 대해 자세히 알아보려면 [Amazon Polly에서 IAM을 사용하는 방법](#) (p. 174) 단원을 참조하십시오.

IAM 관리자 – IAM 관리자는 Amazon Polly에 대한 액세스 권한을 관리하는 정책을 작성하는 방법에 대해 자세히 알아볼 필요가 있습니다. IAM에서 사용할 수 있는 Amazon Polly 자격 증명 기반 정책 예제를 보려면 [Amazon Polly 자격 증명 기반 정책 예제](#) (p. 176) 단원을 참조하십시오.

## 자격 증명을 통한 인증

인증은 ID 자격 증명을 사용하여 AWS에 로그인하는 방식입니다. AWS Management 콘솔을 사용하여 로그인하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 [IAM 콘솔 및 로그인 페이지](#)를 참조하십시오.

AWS 계정 루트 사용자, class="non-printable-char non-printable-space"> IAM class="non-printable-char non-printable-space"> 사용자로서 class="non-printable-char non-printable-space"> 또는 class="non-printable-char non-printable-space"> IAM class="non-printable-char non-printable-space"> 역할을 class="non-printable-char non-printable-space"> 수임하여 class="non-printable-char non-printable-space"> 인증(AWS에 class="non-printable-char non-printable-space"> 로그인)되어야 class="non-printable-char non-printable-space"> 합니다. 회사의 싱글 사인온(SSO) 인증을 사용하거나 Google 또는 Facebook을 사용하여 로그인할 수도 있습니다. 이러한 경우 관리자는 이전에 IAM 역할을 사용하여 자격 증명 연동을 설정한 것입니다. 다른 회사의 자격 증명을 사용하여 AWS에 액세스하면 간접적으로 역할을 가정하는 것입니다.

[AWS Management 콘솔](#)에 직접 로그인하려면 루트 사용자 이메일이나 IAM 사용자 이름과 비밀번호를 사용하십시오. 루트 사용자 또는 IAM 사용자 액세스 키를 사용하여 프로그래밍 방식으로 AWS에 액세스할 수 있습니다. AWS는 자격 증명을 사용하여 암호화 방식으로 요청에 서명할 수 있는 SDK 및 명령줄 도구를 제공합니다. AWS 도구를 사용하지 않는 경우 요청에 직접 서명해야 합니다. 이렇게 하려면 인바운드 API 요청을 인증하기 위한 프로토콜인 서명 버전 4를 사용합니다. 요청 인증에 대한 자세한 정보는 AWS General Reference의 [서명 버전 4 서명 프로세스](#) 단원을 참조하십시오.

사용하는 인증 방법에 상관 없이 추가 보안 정보를 제공해야 할 수도 있습니다. 예를 들어, AWS는 멀티 팩터 인증(MFA)을 사용하여 계정의 보안을 강화하는 것을 권장합니다. 자세한 내용은 IAM 사용 설명서의 [AWS에서 Multi-Factor Authentication\(MFA\) 사용](#)을 참조하십시오.

## AWS 계정 루트 사용자

AWS 계정을 처음 생성할 때는 해당 계정의 모든 AWS 서비스 및 리소스에 대한 완전한 액세스 권한이 있는 SSO(Single Sign-In) ID로 시작합니다. 이 자격 증명은 AWS 계정 루트 사용자라고 하며, 계정을 생성할 때 사용한 이메일 주소와 암호로 로그인하여 액세스합니다. 일상적인 작업은 물론 관리 작업에도 루트 사용자를 사용하지 않는 것이 좋습니다. 대신 [IAM 사용자를 처음 생성할 때만 루트 사용자를 사용하는 모범 사례](#)를 준수하십시오. 그런 다음 루트 사용자 자격 증명을 안전하게 보관해 두고 몇 가지 계정 및 서비스 관리 작업을 수행할 때만 해당 자격 증명을 사용합니다.

## IAM 사용자 및 그룹

[IAM 사용자](#)는 단일 개인 또는 애플리케이션에 대한 특정 권한을 가지고 있는 AWS 계정 내 ID입니다. IAM 사용자에게는 사용자 이름과 암호 또는 액세스 키 세트와 같은 장기 자격 증명이 있을 수 있습니다. 액세스 키를 생성하는 방법을 알아보려면 IAM 사용 설명서의 [IAM 사용자의 액세스 키 관리](#)를 참조하십시오. IAM 사용자의 액세스 키를 생성할 때는 키 페어를 보고 안전하게 저장해야 합니다. 향후에 보안 액세스 키를 복구할 수 없습니다. 그 대신 새 액세스 키 페어를 생성해야 합니다.

[IAM 그룹](#)은 IAM 사용자 컬렉션을 지정하는 ID입니다. 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, IAMAdmins이라는 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 자격 증명만 제공합니다. 자세한 내용은 IAM 사용 설명서의 [IAM 사용자\(역할 대신\)를 생성하는 경우](#)를 참조하십시오.

## IAM 역할

**IAM 역할**은 특정 권한을 가지고 있는 AWS 계정 내 ID입니다. 이 역할은 IAM 사용자와 비슷하지만, 특정 개인과 연결되지 않습니다. **역할을 전환**하여 AWS Management 콘솔에서 IAM 역할을 임시로 수입할 수 있습니다. AWS CLI 또는 AWS API 작업을 호출하거나 사용자 지정 URL을 사용하여 역할을 수입할 수 있습니다. 역할 사용 방법에 대한 자세한 내용은 IAM 사용 설명서의 **IAM 역할 사용**을 참조하십시오.

임시 자격 증명이 있는 IAM 역할은 다음과 같은 상황에서 유용합니다.

- **임시 IAM 사용자 권한** – IAM 사용자는 IAM 역할을 수입하여 특정 작업에 대한 다른 권한을 임시로 받을 수 있습니다.
- **연합된 사용자 액세스** – IAM 사용자를 생성하는 대신 AWS Directory Service의 기존 ID, 엔터프라이즈 사용자 디렉터리 또는 웹 ID 공급자를 사용할 수 있습니다. 이 사용자를 연합된 사용자라고 합니다. AWS에서는 **ID 공급자**를 통해 액세스가 요청되면 연합된 사용자에게 역할을 할당합니다. 연합된 사용자에 대한 자세한 내용은 IAM 사용 설명서의 **연합된 사용자 및 역할**을 참조하십시오.
- **교차 계정 액세스** – IAM 역할을 사용하여 다른 계정의 사용자(신뢰할 수 있는 보안 주체)가 내 계정의 리소스에 액세스하도록 할 수 있습니다. 역할은 교차 계정 액세스를 부여하는 기본적인 방법입니다. 그러나 일부 AWS 서비스를 사용하면 역할을 프록시로 사용하는 대신 리소스에 정책을 직접 연결할 수 있습니다. 교차 계정 액세스를 위한 역할과 리소스 기반 정책의 차이점을 알아보려면 IAM 사용 설명서의 **IAM 역할과 리소스 기반 정책의 차이**를 참조하십시오.
- **AWS 서비스 액세스** – 서비스 역할은 서비스가 사용자를 대신하여 사용자 계정에서 작업을 수행하기 위해 수입하는 IAM 역할입니다. 일부 AWS 서비스 환경을 설정할 때 서비스에서 맡을 역할을 정의해야 합니다. 이 서비스 역할에는 서비스가 AWS 리소스에 액세스하는 데 필요한 모든 권한이 포함되어야 합니다. 서비스 역할은 서비스마다 다르지만 해당 서비스에 대한 문서화된 요구 사항을 충족하는 한 대부분의 경우 권한을 선택할 수 있습니다. 서비스 역할은 해당 계정 내 액세스 권한만 제공하며 다른 계정의 서비스에 대한 액세스 권한을 부여하는 데 사용될 수 없습니다. IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 예를 들어 Amazon Redshift에서 사용자 대신 Amazon S3 버킷에 액세스하도록 허용하는 역할을 생성한 후 해당 버킷에 있는 데이터를 Amazon Redshift 클러스터로 로드할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 **AWS 서비스에 대한 권한을 위임할 역할 생성**을 참조하십시오.
- **Amazon EC2에서 실행 중인 애플리케이션** – IAM 역할을 사용하여 EC2 인스턴스에서 실행되고 AWS CLI 또는 AWS API 요청을 수행하는 애플리케이션의 임시 자격 증명을 관리할 수 있습니다. 이는 EC2 인스턴스 내에 액세스 키를 저장할 때 권장되는 방법입니다. EC2 인스턴스에 AWS 역할을 할당하고 해당 역할을 모든 애플리케이션에서 사용할 수 있도록 하려면 인스턴스에 연결된 인스턴스 프로파일을 생성합니다. 인스턴스 프로파일에는 역할이 포함되어 있으며 EC2 인스턴스에서 실행되는 프로그램이 임시 자격 증명을 얻을 수 있습니다. 자세한 내용은 IAM 사용 설명서의 **IAM 역할을 사용하여 Amazon EC2 인스턴스에서 실행되는 애플리케이션에 권한 부여**를 참조하십시오.

IAM 역할을 사용할지 여부를 알아보려면 IAM 사용 설명서의 **사용자 대신 IAM 역할을 생성해야 하는 경우**를 참조하십시오.

## 정책을 사용하여 액세스 관리

정책을 생성하고 IAM 자격 증명 또는 AWS 리소스에 연결하여 AWS 액세스를 제어합니다. 정책은 자격 증명이나 리소스와 연결될 때 해당 권한을 정의하는 AWS의 객체입니다. AWS는 엔터티(루트 사용자, IAM 사용자 또는 IAM 역할)가 요청을 보낼 때 이러한 정책을 평가합니다. 정책에서 권한은 요청이 허용되거나 거부되는지 여부를 결정합니다. 대부분의 정책은 AWS에 JSON 문서로서 저장됩니다. JSON 정책 문서의 구조와 콘텐츠에 대한 자세한 내용은 IAM 사용 설명서의 **JSON 정책 개요**를 참조하십시오.

IAM 관리자는 AWS 리소스에 액세스할 수 있는 사람과 해당 리소스에 대해 수행할 수 있는 작업을 지정할 수 있습니다. 모든 IAM 엔터티(사용자 또는 역할)는 처음에는 권한이 없습니다. 다시 말해, 기본적으로 사용자는 아무 작업도 수행할 수 없으며, 자신의 암호를 변경할 수도 없습니다. 사용자에게 작업을 수행할 권한을 부여하기 위해 관리자는 사용자에게 권한 정책을 연결해야 합니다. 또한 관리자는 의도한 권한을 가지고 있는 그룹에 사용자를 추가할 수 있습니다. 관리자가 그룹에 권한을 부여하면 그룹의 모든 사용자가 해당 권한을 받습니다.

IAM 정책은 작업을 실행하기 위한 방법과 상관없이 작업을 정의합니다. 예를 들어, `iam:GetRole` 작업을 허용하는 정책이 있다고 가정합니다. 해당 정책이 있는 사용자는 AWS Management 콘솔, AWS CLI 또는 AWS API에서 역할 정보를 가져올 수 있습니다.

## 자격 증명 기반 정책

자격 증명 기반 정책은 IAM 사용자, 역할 또는 그룹과 같은 자격 증명에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 자격 증명이 수행할 수 있는 작업, 대상 리소스 및 이에 관한 조건을 제어합니다. 자격 증명 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 [IAM 정책 생성](#)을 참조하십시오.

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 AWS 계정에 속한 다수의 사용자, 그룹 및 역할에게 독립적으로 추가할 수 있는 정책입니다. 관리형 정책에는 AWS 관리형 정책과 고객 관리형 정책이 포함되어 있습니다. 관리형 정책 또는 인라인 정책을 선택하는 방법을 알아보려면 IAM 사용 설명서의 [관리형 정책과 인라인 정책의 선택](#)을 참조하십시오.

## 리소스 기반 정책

Amazon S3과 같은 다른 서비스도 리소스 기반 권한 정책을 지원합니다. 예를 들어, 정책을 S3 버킷에 연결하여 해당 버킷에 대한 액세스 권한을 관리할 수 있습니다. Amazon Polly의 경우 리소스 기반 정책을 지원하지 않습니다.

## 기타 정책 유형

AWS는 비교적 일반적이지 않은 추가 정책 유형을 지원합니다. 이러한 정책 유형은 더 일반적인 정책 유형에 따라 사용자에게 부여되는 최대 권한을 설정할 수 있습니다.

- 권한 경계 – 권한 경계는 자격 증명 기반 정책에 따라 IAM 엔터티(IAM 사용자 또는 역할)에 부여할 수 있는 최대 권한을 설정하는 고급 기능입니다. 엔터티에 대한 권한 경계를 설정할 수 있습니다. 그 결과로 얻는 권한은 엔터티의 자격 증명 기반 정책의 교차와 그 권한 경계입니다. `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 내용은 IAM 사용 설명서의 [IAM 엔터티에 대한 권한 경계](#)를 참조하십시오.
- 서비스 제어 정책(SCP) – SCP는 AWS Organizations에서 조직 또는 조직 단위(OU)에 최대 권한을 지정하는 JSON 정책입니다. AWS Organizations는 기업이 소유하는 여러 개의 AWS 계정을 그룹화하고 중앙에서 관리하기 위한 서비스입니다. 조직에서 모든 기능을 활성화할 경우 서비스 제어 정책(SCP)을 임의의 또는 모든 계정에 적용할 수 있습니다. SCP는 각 AWS 계정 루트 사용자를 비롯하여 멤버 계정의 엔터티에 대한 권한을 제한합니다. 조직 및 SCP에 대한 자세한 내용은 AWS Organizations 사용 설명서의 [SCP의 작동 방식](#) 단원을 참조하십시오.
- 세션 정책 – 세션 정책은 역할 또는 연합된 사용자에게 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 결과적으로 얻는 세션의 권한은 사용자 또는 역할의 자격 증명 기반 정책의 교차와 세션 정책입니다. 또한 권한을 리소스 기반 정책에서 가져올 수도 있습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 자세한 내용은 IAM 사용 설명서의 [세션 정책](#)을 참조하십시오.

## 여러 정책 유형

여러 정책 유형이 요청에 적용되는 경우 결과 권한은 이해하기가 더 복잡합니다. 여러 정책 유형이 관련될 때 AWS가 요청을 허용할지 여부를 결정하는 방법을 알아보려면 IAM 사용 설명서의 [정책 평가 로직](#)을 참조하십시오.

## Amazon Polly에서 IAM을 사용하는 방법

IAM을 사용하여 Amazon Polly에 대한 액세스를 관리하려면 먼저 어떤 IAM 기능을 Amazon Polly에 사용할 수 있는지를 이해해야 합니다. Amazon Polly 및 기타 AWS 서비스에서 IAM을 사용하는 방법을 개괄적으로 알아보려면 IAM 사용 설명서의 [IAM으로 작업하는 AWS 서비스](#)를 참조하십시오.

## 주제

- [Amazon Polly 자격 증명 기반 정책 \(p. 175\)](#)
- [Amazon Polly IAM 역할 \(p. 175\)](#)

## Amazon Polly 자격 증명 기반 정책

IAM 자격 증명 기반 정책을 사용하면 허용되거나 거부되는 작업과 리소스뿐 아니라 작업이 허용되거나 거부되는 조건을 지정할 수 있습니다. Amazon Polly은 특정 작업, 리소스 및 조건 키를 지원합니다. JSON 정책에서 사용하는 모든 요소에 대해 알아보려면 IAM 사용 설명서의 [IAM JSON 정책 요소 참조](#)를 참조하십시오.

### Actions

IAM 자격 증명 기반 정책의 Action 요소는 정책에 따라 허용되거나 거부되는 특정 작업에 대해 설명합니다. 일반적으로 정책 작업의 이름은 연결된 AWS API 작업의 이름과 동일합니다. 이 작업은 연결된 작업을 수행할 수 있는 권한을 부여하기 위한 정책에서 사용됩니다.

Amazon Polly의 정책 작업은 작업 앞에 polly: 접두사를 사용합니다. 예를 들어 Amazon Polly DeleteLexicon API 작업으로 Amazon Polly 어휘를 삭제할 수 있는 권한을 부여하려면 해당 정책에 polly:DeletedLexicon 작업을 포함합니다. 정책 문에는 Action 또는 NotAction 요소가 포함되어야 합니다. Amazon Polly은 이 서비스로 수행할 수 있는 작업을 설명하는 고유한 작업 집합을 정의합니다.

명령문 하나에 여러 작업을 지정하려면 다음과 같이 쉼표로 구분합니다.

```
"Action": [
    "polly:action1",
    "polly:action2"]
```

와일드카드(\*)를 사용하여 여러 작업을 지정할 수 있습니다. 예를 들어, Get이라는 단어로 시작하는 모든 작업을 지정하려면 다음 작업을 포함시킵니다.

```
"Action": "polly:Get*"
```

Amazon Polly 작업 목록을 보려면 IAM 사용 설명서의 [Actions Defined by Amazon Polly](#) 단원을 참조하십시오.

### 리소스

Amazon Polly은 정책에서 리소스 ARN 지정을 지원하지 않습니다.

### 조건 키

Amazon Polly은 서비스별 조건 키를 제공하지 않지만, 일부 전역 조건 키 사용을 지원합니다. 모든 AWS 전역 조건 키를 보려면 IAM 사용 설명서의 [AWS 전역 조건 컨텍스트 키](#)를 참조하십시오.

### 예제

Amazon Polly 자격 증명 기반 정책 예제를 보려면 [Amazon Polly 자격 증명 기반 정책 예제 \(p. 176\)](#) 단원을 참조하십시오.

## 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 AWS 계정(예: 계정 B) 또는 AWS 서비스에 교차 계정 권한을 부여할 역할을 생성할 수 있습니다.

1. 계정 A 관리자는 IAM 역할을 생성하고 계정 A의 리소스에 대한 권한을 부여하는 역할에 권한 정책을 연결합니다.
2. 계정 A 관리자는 계정 B를 역할을 수입할 보안 주체로 식별하는 역할에 신뢰 정책을 연결합니다.
3. 계정 B 관리자는 계정 B의 사용자에게 역할을 수입할 권한을 위임할 수 있습니다. 그러면 계정 B의 사용자가 계정 A에서 리소스를 생성하거나 액세스할 수 있습니다. AWS 서비스에 역할 수입 권한을 부여할 경우 신뢰 정책의 보안 주체가 AWS 서비스 보안 주체이기도 합니다.

IAM을 사용하여 권한을 위임하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 [액세스 관리](#)를 참조하십시오.

다음은 어휘를 입력하거나 가져오고 현재 사용 가능한 어휘를 나열할 수 있는 권한을 부여하는 정책 예제입니다.

Amazon Polly은 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. 예: 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/*"
  }]
}
```

However, not all operations use ARNs. This is the case with the DescribeVoices, ListLexicons, GetSpeechSynthesisTasks, and ListSpeechSynthesisTasks operations.

사용자, 그룹, 역할 및 권한에 대한 자세한 내용은 IAM 사용 설명서의 [자격 증명\(사용자, 그룹 및 역할\)](#)을 참조하십시오.

## Amazon Polly 자격 증명 기반 정책 예제

기본적으로 IAM 사용자 및 역할은 Amazon Polly 리소스를 생성하거나 수정할 수 있는 권한이 없습니다. 또한 AWS Management 콘솔, AWS CLI 또는 AWS API를 사용해 작업을 수행할 수 없습니다. IAM 관리자는 지정된 리소스에서 특정 API 작업을 수행할 수 있는 권한을 사용자와 역할에게 부여하는 IAM 정책을 생성해야 합니다. 그런 다음 관리자는 해당 권한이 필요한 IAM 사용자 또는 그룹에 이러한 정책을 연결해야 합니다.

이러한 예제 JSON 정책 문서를 사용하여 IAM 자격 증명 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 [JSON 탭에서 정책 생성](#)을 참조하십시오.

주제

- [정책 모범 사례 \(p. 177\)](#)
- [Amazon Polly 콘솔 사용 \(p. 177\)](#)
- [사용자가 자신이 권한을 볼 수 있도록 허용 \(p. 177\)](#)
- [Amazon Polly에 대한 AWS 관리형\(미리 정의된\) 정책 \(p. 178\)](#)
- [고객 관리형 정책 예 \(p. 178\)](#)



## 정책 모범 사례

자격 증명 기반 정책은 매우 강력합니다. 이 정책은 계정에서 사용자가 Amazon Polly 리소스를 생성, 액세스 또는 삭제할 수 있는지 여부를 결정합니다. 이 작업으로 인해 AWS 계정에 비용이 발생할 수 있습니다. 자격 증명 기반 정책을 생성하거나 편집할 때는 다음 지침과 권장 사항을 따르십시오.

- AWS 관리형 정책을 사용하여 시작하기 – Amazon Polly 사용을 빠르게 시작하려면 AWS 관리형 정책을 사용하여 필요한 권한을 직원에게 부여합니다. 이 정책은 이미 계정에서 사용할 수 있으며 AWS에 의해 유지 관리 및 업데이트됩니다. 자세한 내용은 IAM 사용 설명서의 [AWS 관리형 정책으로 권한 사용 시작하기](#)를 참조하십시오.
- 최소 권한 부여 – 사용자 지정 정책을 생성할 때는 작업을 수행하는 데 필요한 권한만 부여합니다. 최소한의 권한 조합으로 시작하여 필요에 따라 추가 권한을 부여합니다. 처음부터 권한을 많이 부여한 후 나중에 줄이는 방법보다 이 방법이 안전합니다. 자세한 내용은 IAM 사용 설명서의 [최소 권한 부여](#)를 참조하십시오.
- 중요한 작업에 대해 MFA 활성화 – 보안을 강화하기 위해 IAM 사용자가 중요한 리소스 또는 API 작업에 액세스하려면 멀티 팩터 인증(MFA)을 사용해야 합니다. 자세한 내용은 IAM 사용 설명서의 [AWS에서 Multi-Factor Authentication\(MFA\) 사용하기](#)를 참조하십시오.
- 보안 강화를 위해 정책 조건 사용 – 실제로 가능한 경우, 자격 증명 기반 정책이 리소스에 대한 액세스를 허용하는 조건을 정의합니다. 예를 들어 요청을 할 수 있는 IP 주소의 범위를 지정하도록 조건을 작성할 수 있습니다. 지정된 날짜 또는 시간 범위 내에서만 요청을 허용하거나, SSL 또는 MFA를 사용해야 하는 조건을 작성할 수도 있습니다. 자세한 내용은 IAM 사용 설명서의 [IAM JSON 정책 요소: 조건](#)을 참조하십시오.

## Amazon Polly 콘솔 사용

사용자가 Amazon Polly 콘솔로 작업하려면 AWS 계정에서 Amazon Polly 리소스를 설명하기 위한 최소 권한이 있어야 합니다.

최소 필수 권한보다 더 제한적인 IAM 정책을 만들면 콘솔에서는 해당 IAM 정책에 연결된 사용자에게 대해 의도대로 작동하지 않습니다.

AWS CLI 또는 Amazon Polly API만 호출하는 사용자에게 최소 콘솔 권한을 허용할 필요가 없습니다.

Amazon Polly 콘솔을 사용하려면 모든 Amazon Polly API에 권한을 부여하십시오. 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 자신이 권한을 볼 수 있도록 허용

이 예제는 IAM 사용자가 자신의 사용자 자격 증명에 연결된 인라인 및 관리형 정책을 볼 수 있도록 허용하는 정책을 생성하는 방법을 보여 줍니다. 이 정책에는 콘솔에서 또는 AWS CLI나 AWS API를 사용하여 프로그래밍 방식으로 이 작업을 완료할 수 있는 권한이 포함됩니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```
{
  "Sid": "ViewOwnUserInfo",
  "Effect": "Allow",
  "Action": [
    "iam:GetUserPolicy",
    "iam:ListGroupsForUser",
    "iam:ListAttachedUserPolicies",
    "iam:ListUserPolicies",
    "iam:GetUser"
  ],
  "Resource": ["arn:aws:iam::*:user/${aws:username}"]
},
{
  "Sid": "NavigateInConsole",
  "Effect": "Allow",
  "Action": [
    "iam:GetGroupPolicy",
    "iam:GetPolicyVersion",
    "iam:GetPolicy",
    "iam:ListAttachedGroupPolicies",
    "iam:ListGroupPolicies",
    "iam:ListPolicyVersions",
    "iam:ListPolicies",
    "iam:ListUsers"
  ],
  "Resource": "*"
}
]
```

## Amazon Polly에 대한 AWS 관리형(미리 정의된) 정책

AWS는 AWS에서 생성하고 관리하는 독립형 IAM 정책을 제공하여 많은 일반 사용 사례를 처리합니다. 이러한 AWS 관리형 정책은 사용자가 필요한 권한을 조사할 필요가 없도록 일반 사용 사례에 필요한 권한을 부여합니다. 자세한 내용은 IAM 사용 설명서의 [AWS 관리형 정책](#) 단원을 참조하십시오.

계정의 사용자에게 연결할 수 있는 다음 AWS 관리형 정책은 Amazon Polly에 대해 고유합니다.

- AmazonPollyReadOnlyAccess – 리소스에 대한 읽기 전용 액세스 권한을 부여하고 어휘 나열, 어휘 불러오기, 사용 가능한 음성 나열, 스피치 합성(합성된 스피치에 어휘 적용 포함) 등을 허용합니다.
- AmazonPollyFullAccess – 리소스와 지원되는 모든 작업에 대한 전체 액세스 권한을 부여합니다.

### Note

IAM 콘솔에 로그인하고 이 콘솔에서 특정 정책을 검색하여 이러한 권한 정책을 검토할 수 있습니다.

IAM 작업 및 리소스에 대한 권한을 허용하는 고유의 사용자 지정 Amazon Polly 정책을 생성할 수도 있습니다. 해당 권한이 필요한 IAM 사용자 또는 그룹에 이러한 사용자 지정 정책을 연결할 수 있습니다.

## 고객 관리형 정책 예

이 단원에서는 다양한 Amazon Polly 작업에 대한 권한을 부여하는 사용자 정책의 예를 제공합니다. 이러한 정책은 AWS SDK 또는 AWS CLI를 사용하는 경우에 유효합니다. 콘솔을 사용할 경우 모든 Amazon Polly API에 권한을 부여합니다.

### Note

모든 예제는 us-east-2 리전을 사용하며 가상의 계정 ID를 포함합니다.

예



- 예제 1: 모든 Amazon Polly 작업 허용 (p. 179)
- 예제 2: DeleteLexicon을 제외한 모든 Amazon Polly 작업 허용 (p. 179)
- 예제 3: DeleteLexicon 허용 (p. 179)
- 예제 4: 지정된 리전에서 어휘 삭제 허용 (p. 180)
- 예제 5: 지정된 어휘에 대한 DeleteLexicon 허용 (p. 180)

## 예제 1: 모든 Amazon Polly 작업 허용

가입한 후(1.1단계: AWS에 가입 (p. 4) 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 사용자를 Amazon Polly로 작업하는 서비스별 관리자라고 생각하십시오. 이 사용자에게 다음 권한 정책을 연결할 수 있습니다.

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

## 예제 2: DeleteLexicon을 제외한 모든 Amazon Polly 작업 허용

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": "*"
  }]
}
```

## 예제 3: DeleteLexicon 허용

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/*"
  }]
}
```

#### 예제 5: 지정된 어휘에 대한 DeleteLexicon 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 자격 증명(자격 증명 기반 정책)에 연결할 수 있는 권한 정책을 설정할 때 다음 목록을 참조로 사용할 수 있습니다. 예는 Amazon Polly API 작업, 부여된 권한으로 수행할 수 있는 작업, 권한 부여된 AWS 리소스가 각각 나열되어 있습니다. 정책의 Action 필드에서 작업을 지정하고, 정책의 Resource 필드에서 리소스 값을 지정합니다.

Amazon Polly 정책에서 AWS 차원 조건 키를 사용하여 조건을 표시할 수 있습니다. AWS 차원 키의 전체 목록은 IAM 사용 설명서의 [사용할 수 있는 키](#) 섹션을 참조하십시오.

#### Note

작업을 지정하려면 polly 접두사 다음에 API 작업 이름을 사용합니다(예: polly:GetLexicon).

Amazon Polly은 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 Polly은 리소스 수준의 작업에 대한 권한을 지원하지 않으므로, 대부분의 정책에서는 와일드카드 문자(\*)를 Resource 값으로 지정합니다. 하지만 권한을 특정 리전으로 제한해야 하는 경우 이 와일드카드 문자를 적절한 ARN(`arn:aws:polly:region:account-id:lexicon/*`)으로 대체합니다.

Amazon Polly API 및 작업에 대한 필수 권한

API 작업: [DeleteLexicon](#) (p. 190)

필요한 권한(API 작업): `polly:DeleteLexicon`

리소스: `arn:aws:polly:region:account-id:lexicon/LexiconName`

API 작업: [DescribeVoices](#) (p. 192)

필요한 권한(API 작업): `polly:DescribeVoices`

리소스: `arn:aws:polly:region:account-id:lexicon/voice-name`

API 작업: [GetLexicon](#) (p. 195)

필요한 권한(API 작업): `polly:GetLexicon`

리소스: `arn:aws:polly:region:account-id:lexicon/voice-name`

API 작업: [ListLexicons](#) (p. 199)

필요한 권한(API 작업): `polly:ListLexicons`

리소스: `arn:aws:polly:region:account-id:lexicon/*`

API 작업: [PutLexicon](#) (p. 203)

필요한 권한(API 작업): `polly:ListLexicons`

리소스: \*

API 작업: [SynthesizeSpeech](#) (p. 211)

필요한 권한(API 작업): `polly:SynthesizeSpeech`

리소스: \*

## Amazon Polly의 로깅 및 모니터링

모니터링은 Amazon Polly 애플리케이션의 안정성, 가용성 및 성능을 유지하는 중요한 역할을 합니다. Amazon Polly API 호출을 모니터링하려면 AWS CloudTrail를 사용합니다. 작업 상태를 모니터링하려면 Amazon CloudWatch Logs를 사용합니다.

- Amazon CloudWatch Alarms – CloudWatch 경보를 사용하면 지정한 기간 동안 단일 지표를 감시할 수 있습니다. 지표가 지정한 임계값을 초과하면 Amazon Simple Notification Service 주제 또는 AWS Auto Scaling 정책에 알림이 전송됩니다. CloudWatch 경보는 지표가 특정 상태에 있는 경우 작업을 호출하지 않습니다. 대신, 상태가 변경되어 지정한 기간 동안 유지되어야 합니다. 자세한 내용은 [CloudWatch와 Amazon Polly 통합](#) (p. 186) 단원을 참조하십시오.
- AWS CloudTrail Logs – CloudTrail은 Amazon Polly에서 사용자, 역할 또는 AWS 서비스가 수행한 작업의 기록을 제공합니다. CloudTrail에서 수집하는 정보를 사용하여 Amazon Polly에 어떤 요청이 이루어졌는지 확인할 수 있습니다. 또한 어떤 IP 주소에서 요청했는지, 누가 언제 요청했는지 등의 추가 세부 정보도 확

인할 수 있습니다. 자세한 내용은 [AWS CloudTrail을 사용하여 Amazon Polly API 호출 로깅 \(p. 184\)](#) 단원을 참조하십시오.

## Amazon Polly 규정 준수 확인

타사 감사자는 여러 AWS 규정 준수 프로그램의 일환으로 Amazon Polly의 보안 및 규정 준수 상태를 평가합니다. 여기에는 SOC, PCI, FedRAMP, HIPAA 등이 포함됩니다.

특정 규정 준수 프로그램의 범위 내에 있는 AWS 서비스 목록은 [규정 준수 프로그램 제공 범위 내 AWS 서비스](#)를 참조하십시오. 일반적인 내용은 [AWS 규정 준수 프로그램](#)을 참조하십시오.

AWS Artifact를 사용하여 타사 감사 보고서를 다운로드할 수 있습니다. 자세한 내용은 [AWS 아티팩트에서 보고서 다운로드](#)를 참조하십시오.

Amazon Polly 사용 시 규정 준수 책임은 데이터의 민감도, 회사의 규정 준수 목표 및 관련 법률과 규정에 따라 결정됩니다. AWS에서는 규정 준수를 지원할 다음과 같은 리소스를 제공합니다.

- [보안 및 규정 준수 빠른 시작 안내서](#) – 이 배포 가이드에서는 아키텍처 고려 사항에 대해 설명하고 보안 및 규정 준수에 중점을 둔 기본 AWS 환경을 배포하기 위한 단계를 제공합니다.
- [HIPAA 보안 및 규정 준수 기술 백서 설계](#) – 이 백서는 기업에서 AWS를 사용하여 HIPAA를 준수하는 애플리케이션을 만드는 방법을 설명합니다.
- [AWS 규정 준수 리소스](#) – 이 워크북 및 안내서 모음은 귀사가 속한 업계 및 위치에 적용될 수 있습니다.
- AWS Config 개발자 안내서의 [규칙을 사용하여 리소스 평가](#) – 이 AWS Config 서비스는 리소스 구성이 내부 사례, 업계 지침, 규정을 얼마나 잘 준수하는지 평가합니다.
- [AWS Security Hub](#) – 이 AWS 제품으로 보안 업계 표준 및 모범 사례 규정 준수 여부를 확인하는 데 도움이 되는 AWS 내 보안 상태에 대한 포괄적인 관점을 제공합니다.

## Amazon Polly의 복원성

AWS 글로벌 인프라는 AWS 리전 및 가용 영역을 중심으로 구축됩니다. AWS 리전에서는 물리적으로 분리되고 격리된 다수의 가용 영역을 제공하며 이러한 가용 영역은 짧은 지연 시간, 높은 처리량 및 높은 중복성을 갖춘 네트워크에 연결되어 있습니다. 가용 영역을 사용하면 중단 없이 영역 간에 자동으로 장애 조치가 이루어지는 애플리케이션 및 데이터베이스를 설계하고 운영할 수 있습니다. 가용 영역은 기존의 단일 또는 다중 데이터 센터 인프라보다 가용성, 내결함성, 확장성이 뛰어납니다.

AWS 리전 및 가용 영역에 대한 자세한 내용은 [AWS 글로벌 인프라](#)를 참조하십시오.

## Amazon Polly의 인프라 보안

관리형 서비스인 Amazon Polly은 [Amazon Web Services: 보안 프로세스 개요](#) 백서에 설명된 AWS 글로벌 네트워크 보안 절차로 보호됩니다.

네트워크를 통해 Amazon Polly에 액세스하려면 AWS에서 게시한 API 호출을 사용합니다. 클라이언트가 TLS(전송 계층 보안) 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 보안 액세스 키를 사용하여 서명해야 합니다. 또는 [AWS Security Token Service\(AWS STS\)](#)를 사용하여 임시 보안 자격 증명을 생성하여 요청에 서명할 수 있습니다.

## Amazon Polly의 보안 모범 사례

AWS에서는 고객의 신뢰, 개인 정보, 콘텐츠의 보안을 최우선으로 하고 있습니다. AWS에서는 콘텐츠에 대한 무단 액세스 또는 유출을 방지하기 위해 정교하고 책임 있는 기술적 및 물리적 제어를 구현하고 고객과의 약속을 충실히 준수합니다. 자세한 내용은 [AWS 데이터 프라이버시 FAQ](#)를 참조하십시오.

Amazon Polly은 텍스트 제출의 콘텐츠를 보관하지 않습니다.

규정 준수, 침투 테스트, 공지 및 리소스를 비롯한 AWS 보안에 대한 자세한 내용을 보려면 [AWS 클라우드 보안](#) 웹사이트를 방문하십시오.

# AWS CloudTrail을 사용하여 Amazon Polly API 호출 로깅

Amazon Polly는 Amazon Polly에서 사용자, 역할 또는 AWS 서비스가 수행한 작업에 대한 레코드를 제공하는 서비스인 AWS CloudTrail과 통합됩니다. CloudTrail은 Amazon Polly에 대한 모든 API 호출을 이벤트로 캡처합니다. 캡처되는 호출에는 Amazon Polly 콘솔로부터의 호출과 Amazon Polly API 작업에 대한 호출이 포함됩니다. 추적을 생성하면 Amazon Polly에 대한 이벤트를 비롯하여 CloudTrail 이벤트를 Amazon S3 버킷으로 지속적으로 배포할 수 있습니다. 추적을 구성하지 않은 경우 Event history(이벤트 기록)에서 CloudTrail 콘솔의 최신 이벤트를 볼 수도 있습니다. CloudTrail에서 수집하는 정보를 사용하여 Amazon Polly에 수행된 요청, 요청이 수행된 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다.

그 구성 및 활성화 방법을 포함하여 CloudTrail에 대한 자세한 내용은 [AWS CloudTrail User Guide](#)를 참조하십시오.

## CloudTrail의 Amazon Polly 정보

CloudTrail은 계정 생성 시 AWS 계정에서 활성화됩니다. 지원되는 이벤트 활동이 Amazon Polly에서 이루어지면 해당 활동이 이벤트 이력의 다른 AWS 서비스 이벤트와 함께 CloudTrail 이벤트에 기록됩니다. AWS 계정에서 최신 이벤트를 확인, 검색 및 다운로드할 수 있습니다. 자세한 내용은 [CloudTrail 이벤트 기록에서 이벤트 보기](#)를 참조하십시오.

Amazon Polly 이벤트를 비롯하여 AWS 계정에 이벤트를 지속적으로 기록하려면 추적을 생성하십시오. 추적은 CloudTrail이 Amazon S3 버킷으로 로그 파일을 전송할 수 있도록 합니다. 콘솔에서 추적을 생성하면 기본적으로 모든 AWS 리전에 추적이 적용됩니다. 추적은 AWS 파티션에 있는 모든 리전의 이벤트를 로깅하고 지정한 Amazon S3 버킷으로 로그 파일을 전송합니다. 또는 CloudTrail 로그에서 수집된 이벤트 데이터를 추가 분석 및 처리하도록 다른 AWS 서비스를 구성할 수 있습니다. 자세한 정보는 다음을 참조하십시오.

- [추적 생성 개요](#)
- [CloudTrail 지원 서비스 및 통합](#)
- [CloudTrail에 대한 Amazon SNS 알림 구성](#)
- [여러 리전에서 CloudTrail 로그 파일 받기 및 여러 계정에서 CloudTrail 로그 파일 받기](#)

Amazon Polly에서는 다음 작업을 CloudTrail 로그 파일에 이벤트로 로깅할 수 있습니다.

- [DeleteLexicon](#) (p. 190)
- [DescribeVoices](#) (p. 192)
- [GetLexicon](#) (p. 195)
- [GetSpeechSynthesisTask](#) (p. 197)
- [ListLexicons](#) (p. 199)
- [ListSpeechSynthesisTasks](#) (p. 201)
- [PutLexicon](#) (p. 203)
- [StartSpeechSynthesisTask](#) (p. 205)
- [SynthesizeSpeech](#) (p. 211)

모든 이벤트 및 로그 항목에는 요청을 생성한 사용자에 대한 정보가 들어 있습니다. 자격 증명 정보를 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청을 루트로 했는지 아니면 AWS Identity and Access Management(IAM) 사용자 자격 증명으로 했는지 여부
- 역할 또는 연합된 사용자에게 대한 임시 보안 자격 증명을 사용하여 요청이 생성되었는지 여부.
- 다른 AWS 서비스에서 요청했는지 여부.

자세한 내용은 [CloudTrail userIdentity](#) 요소를 참조하십시오.

## 예제: Amazon Polly 로그 파일 항목

추적은 지정한 Amazon S3 버킷에 이벤트를 로그 파일로 제공할 수 있도록 해 주는 구성입니다. CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함됩니다. 이벤트는 어떤 소스로부터의 단일 요청을 나타내며 요청된 작업, 작업 날짜와 시간, 요청 파라미터 등에 대한 정보가 포함되어 있습니다. CloudTrail 로그 파일은 퍼블릭 API 호출의 주문 스택 추적이 아니므로 특정 순서로 표시되지 않습니다.

다음은 SynthesizeSpeech를 설명하는 CloudTrail 로그 항목을 보여주는 예입니다.

```
{
  "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-alaf-11e6-b17f-d7cfc06cb461",
      "requestParameters": {
        "lexiconNames": [
          "SampleLexicon"
        ],
        "outputFormat": "mp3",
        "sampleRate": "22050",
        "text": "*****",
        "textType": "text",
        "voiceId": "Kendra"
      },
      "responseElements": {
        "contentType": "audio/mpeg",
        "requestCharacters": 25
      },
      "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"
      }
    }
  ]
}
```

# CloudWatch와 Amazon Polly 통합

Amazon Polly와 상호 작용할 때 매분 다음 측정치와 차원을 CloudWatch로 전송합니다. 다음 절차에 따라 Amazon Polly에 대한 측정치를 볼 수 있습니다.

Amazon Polly에서 원시 데이터를 수집하여 읽기 가능하며 실시간에 가까운 측정치로 처리하는 CloudWatch를 통해 Amazon Polly를 모니터링할 수 있습니다. 이러한 통계는 2주간 기록되므로 `historical information`을 보고 웹 애플리케이션이나 서비스가 어떻게 실행되고 있는지 전체적으로 더 잘 파악할 수 있습니다. 기본적으로 Amazon Polly 측정치 데이터는 1분 간격으로 CloudWatch에 전송됩니다. 자세한 내용은 Amazon CloudWatch 사용 설명서의 [Amazon CloudWatch란?](#) 단원을 참조하십시오.

## CloudWatch 측정치 가져오기(콘솔)

1. <https://console.aws.amazon.com/cloudwatch/>에서 CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 [Metrics]를 선택합니다.
3. 범주별 CloudWatch 측정치 창의 Amazon Polly에 대한 측정치 범주에서 측정치 범주를 선택하고, 위쪽 창에서 아래로 스크롤하여 전체 측정치 목록을 봅니다.

## CloudWatch 측정치 가져오기(CLI)

다음 코드는 Amazon Polly에 사용 가능한 측정치를 표시합니다.

```
aws cloudwatch list-metrics --namespace "AWS/Polly"
```

이전 명령은 다음과 유사한 Amazon Polly 측정치 목록을 반환합니다. `MetricName` 요소는 측정치를 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 참조의 [GetMetricStatistics](#) 단원을 참조하십시오.



## Amazon Polly 지표

Amazon Polly은(는) 각 요청에 대한 다음 지표를 만듭니다. 이러한 지표는 집계된 다음 1분 간격으로 CloudWatch(으)로 전송되며 집계된 지표는 &CW;에서 사용할 수 있습니다.

지표	설명
<code>RequestCharacters</code>	<p>요청의 문자 수. 청구 가능한 문자이며 SSML 태그가 포함되지 않습니다.</p> <p>유효한 차원: Operation</p> <p>유효한 통계: Minimum, Maximum, Average, SampleCount, Sum</p> <p>단위: 수</p>
<code>ResponseLatency</code>	<p>요청 시간과 스트리밍 응답이 시작되는 시간 사이의 지연 시간</p> <p>유효한 차원: Operation</p> <p>유효한 통계: Minimum, Maximum, Average, SampleCount</p> <p>단위: 밀리초</p>
<code>2XXCount</code>	<p>성공적 응답 시 반환되는 HTTP 200 레벨 코드</p> <p>유효한 차원: Operation</p> <p>유효한 통계: Average, SampleCount, Sum</p> <p>단위: 수</p>
<code>4XXCount</code>	<p>오류 발생 시 반환되는 HTTP 400 레벨 오류 코드. 성공적 응답이 발생할 때마다 0을 내보냅니다.</p> <p>유효한 차원: Operation</p> <p>유효한 통계: Average, SampleCount, Sum</p> <p>단위: 수</p>
<code>5XXCount</code>	<p>오류 발생 시 반환되는 HTTP 500 레벨 오류 코드. 성공적 응답이 발생할 때마다 0을 내보냅니다.</p> <p>유효한 차원: Operation</p> <p>유효한 통계: Average, SampleCount, Sum</p> <p>단위: 수</p>

## Amazon Polly 지표 차원

Amazon Polly 지표는 AWS/Polly 네임스페이스를 사용하며 다음 차원의 지표를 제공합니다.

차원	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 Amazon Web Services 일반 참조의 [AWS API 요청에 서명](#) 단원을 참조하십시오.

## 주제

- [Actions](#) (p. 189)
- [Data Types](#) (p. 215)

## Actions

The following actions are supported:

- [DeleteLexicon](#) (p. 190)
- [DescribeVoices](#) (p. 192)
- [GetLexicon](#) (p. 195)
- [GetSpeechSynthesisTask](#) (p. 197)
- [ListLexicons](#) (p. 199)
- [ListSpeechSynthesisTasks](#) (p. 201)
- [PutLexicon](#) (p. 203)
- [StartSpeechSynthesisTask](#) (p. 205)
- [SynthesizeSpeech](#) (p. 211)

## DeleteLexicon

Deletes the specified pronunciation lexicon stored in an AWS Region. A lexicon which has been deleted is not available for speech synthesis, nor is it possible to retrieve it using either the `GetLexicon` or `ListLexicon` APIs.

For more information, see [Managing Lexicons](#).

## Request Syntax

```
DELETE /v1/lexicons/LexiconName HTTP/1.1
```

## URI Request Parameters

The request requires the following URI parameters.

`LexiconName` (p. 190)

The name of the lexicon to delete. Must be an existing lexicon in the region.

Pattern: `[0-9A-Za-z]{1,20}`

## Request Body

The request does not have a request body.

## Response Syntax

```
HTTP/1.1 200
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

## Errors

`LexiconNotFoundException`

Amazon Polly can't find the specified lexicon. This could be caused by a lexicon that is missing, its name is misspelled or specifying a lexicon that is in a different region.

Verify that the lexicon exists, is in the region (see [ListLexicons](#) (p. 199)) and that you spelled its name is spelled correctly. Then try again.

HTTP Status Code: 404

`ServiceFailureException`

An unknown condition has caused a service failure.

HTTP Status Code: 500

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## DescribeVoices

Returns the list of voices that are available for use when requesting speech synthesis. Each voice speaks a specified language, is either male or female, and is identified by an ID, which is the ASCII version of the voice name.

When synthesizing speech ( `SynthesizeSpeech` ), you provide the voice ID for the voice you want from the list of voices returned by `DescribeVoices`.

For example, you want your news reader application to read news in a specific language, but giving a user the option to choose the voice. Using the `DescribeVoices` operation you can provide the user with a list of available voices to select from.

You can optionally specify a language code to filter the available voices. For example, if you specify `en-US`, the operation returns a list of all available US English voices.

This operation requires permissions to perform the `polly:DescribeVoices` action.

## Request Syntax

```
GET /v1/voices?  
Engine=Engine&IncludeAdditionalLanguageCodes=IncludeAdditionalLanguageCodes&LanguageCode=LanguageCode&  
HTTP/1.1
```

## URI Request Parameters

The request requires the following URI parameters.

### [Engine](#) (p. 192)

Specifies the engine (`standard` or `neural`) used by Amazon Polly when processing input text for speech synthesis.

Valid Values: `standard` | `neural`

### [IncludeAdditionalLanguageCodes](#) (p. 192)

Boolean value indicating whether to return any bilingual voices that use the specified language as an additional language. For instance, if you request all languages that use US English (`es-US`), and there is an Italian voice that speaks both Italian (`it-IT`) and US English, that voice will be included if you specify `yes` but not if you specify `no`.

### [LanguageCode](#) (p. 192)

The language identification tag (ISO 639 code for the language name-ISO 3166 country code) for filtering the list of voices returned. If you don't specify this optional parameter, all available voices are returned.

Valid Values: `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`

### [NextToken](#) (p. 192)

An opaque pagination token returned from the previous `DescribeVoices` operation. If present, this indicates where to continue the listing.

Length Constraints: Minimum length of 0. Maximum length of 4096.

## Request Body

The request does not have a request body.

## 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" ]
    }
  ]
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

### [NextToken \(p. 193\)](#)

The pagination token to use in the next request to continue the listing of voices. `NextToken` is returned only if the response is truncated.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 4096.

### [Voices \(p. 193\)](#)

A list of voices with their properties.

Type: Array of [Voice \(p. 223\)](#) objects

## Errors

### InvalidNextTokenException

The `NextToken` is invalid. Verify that it's spelled correctly, and then try again.

HTTP Status Code: 400

### ServiceFailureException

An unknown condition has caused a service failure.

HTTP Status Code: 500

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)



# GetLexicon

Returns the content of the specified pronunciation lexicon stored in an AWS Region. For more information, see [Managing Lexicons](#).

## Request Syntax

```
GET /v1/lexicons/LexiconName HTTP/1.1
```

## URI Request Parameters

The request requires the following URI parameters.

[LexiconName](#) (p. 195)

Name of the lexicon.

Pattern: `[0-9A-Za-z]{1,20}`

## Request Body

The request does not have a request body.

## 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
  }
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

[Lexicon](#) (p. 195)

Lexicon object that provides name and the string content of the lexicon.

Type: [Lexicon](#) (p. 216) object

### [LexiconAttributes \(p. 195\)](#)

Metadata of the lexicon, including phonetic alphabetic used, language code, lexicon ARN, number of lexemes defined in the lexicon, and size of lexicon in bytes.

Type: [LexiconAttributes \(p. 217\)](#) object

## Errors

### LexiconNotFoundException

Amazon Polly can't find the specified lexicon. This could be caused by a lexicon that is missing, its name is misspelled or specifying a lexicon that is in a different region.

Verify that the lexicon exists, is in the region (see [ListLexicons \(p. 199\)](#)) and that you spelled its name is spelled correctly. Then try again.

HTTP Status Code: 404

### ServiceFailureException

An unknown condition has caused a service failure.

HTTP Status Code: 500

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## GetSpeechSynthesisTask

Retrieves a specific SpeechSynthesisTask object based on its TaskID. This object contains information about the given speech synthesis task, including the status of the task, and a link to the S3 bucket containing the output of the task.

### Request Syntax

```
GET /v1/speechSynthesisTasks/TaskId HTTP/1.1
```

### URI Request Parameters

The request requires the following URI parameters.

[TaskId](#) (p. 197)

The Amazon Polly generated identifier for a speech synthesis task.

Pattern: `^[a-zA-Z0-9_-]{1,100}$`

### Request Body

The request does not have a request body.

### 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"
  }
}
```

### Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

### [SynthesisTask \(p. 197\)](#)

SynthesisTask object that provides information from the requested task, including output format, creation time, task status, and so on.

Type: [SynthesisTask \(p. 220\)](#) object

## Errors

### InvalidTaskIdException

The provided Task ID is not valid. Please provide a valid Task ID and try again.

HTTP Status Code: 400

### ServiceFailureException

An unknown condition has caused a service failure.

HTTP Status Code: 500

### SynthesisTaskNotFoundException

The Speech Synthesis task with requested Task ID cannot be found.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## ListLexicons

Returns a list of pronunciation lexicons stored in an AWS Region. For more information, see [Managing Lexicons](#).

## Request Syntax

```
GET /v1/lexicons?NextToken=NextToken HTTP/1.1
```

## URI Request Parameters

The request requires the following URI parameters.

[NextToken](#) (p. 199)

An opaque pagination token returned from previous `ListLexicons` operation. If present, indicates where to continue the list of lexicons.

Length Constraints: Minimum length of 0. Maximum length of 4096.

## Request Body

The request does not have a request body.

## 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"
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

[Lexicons](#) (p. 199)

A list of lexicon names and attributes.

Type: Array of [LexiconDescription](#) (p. 219) objects

[NextToken](#) (p. 199)

The pagination token to use in the next request to continue the listing of lexicons. `NextToken` is returned only if the response is truncated.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 4096.

## Errors

`InvalidNextTokenException`

The `NextToken` is invalid. Verify that it's spelled correctly, and then try again.

HTTP Status Code: 400

`ServiceFailureException`

An unknown condition has caused a service failure.

HTTP Status Code: 500

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## ListSpeechSynthesisTasks

Returns a list of SpeechSynthesisTask objects ordered by their creation date. This operation can filter the tasks by their status, for example, allowing users to list only tasks that are completed.

### Request Syntax

```
GET /v1/synthesisTasks?MaxResults=MaxResults&NextToken=NextToken&Status=Status HTTP/1.1
```

### URI Request Parameters

The request requires the following URI parameters.

#### MaxResults (p. 201)

Maximum number of speech synthesis tasks returned in a List operation.

Valid Range: Minimum value of 1. Maximum value of 100.

#### NextToken (p. 201)

The pagination token to use in the next request to continue the listing of speech synthesis tasks.

Length Constraints: Minimum length of 0. Maximum length of 4096.

#### Status (p. 201)

Status of the speech synthesis tasks returned in a List operation

Valid Values: `scheduled` | `inProgress` | `completed` | `failed`

### Request Body

The request does not have a request body.

### 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"
    }
  ]
}
```

```
}  
  ]  
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

### [NextToken](#) (p. 201)

An opaque pagination token returned from the previous List operation in this request. If present, this indicates where to continue the listing.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 4096.

### [SynthesisTasks](#) (p. 201)

List of SynthesisTask objects that provides information from the specified task in the list request, including output format, creation time, task status, and so on.

Type: Array of [SynthesisTask](#) (p. 220) objects

## Errors

### InvalidNextTokenException

The NextToken is invalid. Verify that it's spelled correctly, and then try again.

HTTP Status Code: 400

### ServiceFailureException

An unknown condition has caused a service failure.

HTTP Status Code: 500

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)



## PutLexicon

Stores a pronunciation lexicon in an AWS Region. If a lexicon with the same name already exists in the region, it is overwritten by the new lexicon. Lexicon operations have eventual consistency, therefore, it might take some time before the lexicon is available to the SynthesizeSpeech operation.

For more information, see [Managing Lexicons](#).

## Request Syntax

```
PUT /v1/lexicons/LexiconName HTTP/1.1
Content-type: application/json

{
  "Content": "string"
}
```

## URI Request Parameters

The request requires the following URI parameters.

[LexiconName \(p. 203\)](#)

Name of the lexicon. The name must follow the regular express format `[0-9A-Za-z]{1,20}`. That is, the name is a case-sensitive alphanumeric string up to 20 characters long.

Pattern: `[0-9A-Za-z]{1,20}`

## Request Body

The request accepts the following data in JSON format.

[Content \(p. 203\)](#)

Content of the PLS lexicon as string data.

Type: String

Required: Yes

## Response Syntax

```
HTTP/1.1 200
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

## Errors

`InvalidLexiconException`

Amazon Polly can't find the specified lexicon. Verify that the lexicon's name is spelled correctly, and then try again.

HTTP Status Code: 400

LexiconSizeExceededException

The maximum size of the specified lexicon would be exceeded by this operation.

HTTP Status Code: 400

MaxLexemeLengthExceededException

The maximum size of the lexeme would be exceeded by this operation.

HTTP Status Code: 400

MaxLexiconsNumberExceededException

The maximum number of lexicons would be exceeded by this operation.

HTTP Status Code: 400

ServiceFailureException

An unknown condition has caused a service failure.

HTTP Status Code: 500

UnsupportedPlsAlphabetException

The alphabet specified by the lexicon is not a supported alphabet. Valid values are `x-sampa` and `ipa`.

HTTP Status Code: 400

UnsupportedPlsLanguageException

The language specified in the lexicon is unsupported. For a list of supported languages, see [Lexicon Attributes](#).

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## StartSpeechSynthesisTask

Allows the creation of an asynchronous synthesis task, by starting a new `SpeechSynthesisTask`. This operation requires all the standard information needed for speech synthesis, plus the name of an Amazon S3 bucket for the service to store the output of the synthesis task and two optional parameters (`OutputS3KeyPrefix` and `SnsTopicArn`). Once the synthesis task is created, this operation will return a `SpeechSynthesisTask` object, which will include an identifier of this task as well as the current status.

### 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 Request Parameters

The request does not use any URI parameters.

### Request Body

The request accepts the following data in JSON format.

#### [Engine \(p. 205\)](#)

Specifies the engine (`standard` or `neural`) for Amazon Polly to use when processing input text for speech synthesis. Using a voice that is not supported for the engine selected will result in an error.

Type: String

Valid Values: `standard` | `neural`

Required: No

#### [LanguageCode \(p. 205\)](#)

Optional language code for the Speech Synthesis request. This is only necessary if using a bilingual voice, such as Aditi, which can be used for either Indian English (`en-IN`) or Hindi (`hi-IN`).

If a bilingual voice is used and no language code is specified, Amazon Polly will use the default language of the bilingual voice. The default language for any voice is the one returned by the [DescribeVoices](#) operation for the `LanguageCode` parameter. For example, if no language code is specified, Aditi will use Indian English rather than Hindi.

Type: String

Valid Values: 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

Required: No

#### [LexiconNames \(p. 205\)](#)

List of one or more pronunciation lexicon names you want the service to apply during synthesis. Lexicons are applied only if the language of the lexicon is the same as the language of the voice.

Type: Array of strings

Array Members: Maximum number of 5 items.

Pattern: [0-9A-Za-z]{1,20}

Required: No

#### [OutputFormat \(p. 205\)](#)

The format in which the returned output will be encoded. For audio stream, this will be mp3, ogg\_vorbis, or pcm. For speech marks, this will be json.

Type: String

Valid Values: json | mp3 | ogg\_vorbis | pcm

Required: Yes

#### [OutputS3BucketName \(p. 205\)](#)

Amazon S3 bucket name to which the output file will be saved.

Type: String

Pattern: ^[a-z0-9][\.\-a-z0-9]{1,61}[a-z0-9]\$

Required: Yes

#### [OutputS3KeyPrefix \(p. 205\)](#)

The Amazon S3 key prefix for the output speech file.

Type: String

Pattern: ^[0-9a-zA-Z\/\!\-\\_\.\\*\'\(\)]{0,800}\$

Required: No

#### [SampleRate \(p. 205\)](#)

The audio frequency specified in Hz.

The valid values for mp3 and ogg\_vorbis are "8000", "16000", "22050", and "24000". The default value for standard voices is "22050". The default value for neural voices is "24000".

Valid values for pcm are "8000" and "16000" The default value is "16000".

Type: String

Required: No

#### [SnsTopicArn \(p. 205\)](#)

ARN for the SNS topic optionally used for providing status notification for a speech synthesis task.

Type: String

Pattern: `^arn:aws(-(cn|iso(-b)?|us-gov))?:sns:[a-z0-9_-]{1,50}:\d{12}:[a-zA-Z0-9_-]{1,256}$`

Required: No

[SpeechMarkTypes \(p. 205\)](#)

The type of speech marks returned for the input text.

Type: Array of strings

Array Members: Maximum number of 4 items.

Valid Values: `sentence` | `ssml` | `viseme` | `word`

Required: No

[Text \(p. 205\)](#)

The input text to synthesize. If you specify `ssml` as the `TextType`, follow the SSML format for the input text.

Type: String

Required: Yes

[TextType \(p. 205\)](#)

Specifies whether the input text is plain text or SSML. The default value is plain text.

Type: String

Valid Values: `ssml` | `text`

Required: No

[Voiceld \(p. 205\)](#)

Voice ID to use for the synthesis.

Type: String

Valid Values: `Aditi` | `Amy` | `Astrid` | `Bianca` | `Brian` | `Camila` | `Carla` | `Carmen` | `Celine` | `Chantal` | `Conchita` | `Cristiano` | `Dora` | `Emma` | `Enrique` | `Ewa` | `Filiz` | `Geraint` | `Giorgio` | `Gwyneth` | `Hans` | `Ines` | `Ivy` | `Jacek` | `Jan` | `Joanna` | `Joey` | `Justin` | `Karl` | `Kendra` | `Kimberly` | `Lea` | `Liv` | `Lotte` | `Lucia` | `Lupe` | `Mads` | `Maja` | `Marlene` | `Mathieu` | `Matthew` | `Maxim` | `Mia` | `Miguel` | `Mizuki` | `Naja` | `Nicole` | `Penelope` | `Raveena` | `Ricardo` | `Ruben` | `Russell` | `Salli` | `Seoyeon` | `Takumi` | `Tatyana` | `Vicki` | `Vitoria` | `Zeina` | `Zhiyu`

Required: Yes

## 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"
}
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

[SynthesisTask](#) (p. 207)

SynthesisTask object that provides information and attributes about a newly submitted speech synthesis task.

Type: [SynthesisTask](#) (p. 220) object

## Errors

### EngineNotSupportedException

This engine is not compatible with the voice that you have designated. Choose a new voice that is compatible with the engine or change the engine and restart the operation.

HTTP Status Code: 400

### InvalidS3BucketException

The provided Amazon S3 bucket name is invalid. Please check your input with S3 bucket naming requirements and try again.

HTTP Status Code: 400

### InvalidS3KeyException

The provided Amazon S3 key prefix is invalid. Please provide a valid S3 object key name.

HTTP Status Code: 400

### InvalidSampleRateException

The specified sample rate is not valid.

HTTP Status Code: 400

### InvalidSnsTopicArnException

The provided SNS topic ARN is invalid. Please provide a valid SNS topic ARN and try again.

HTTP Status Code: 400

#### InvalidSsmlException

The SSML you provided is invalid. Verify the SSML syntax, spelling of tags and values, and then try again.

HTTP Status Code: 400

#### LanguageNotSupportedException

The language specified is not currently supported by Amazon Polly in this capacity.

HTTP Status Code: 400

#### LexiconNotFoundException

Amazon Polly can't find the specified lexicon. This could be caused by a lexicon that is missing, its name is misspelled or specifying a lexicon that is in a different region.

Verify that the lexicon exists, is in the region (see [ListLexicons \(p. 199\)](#)) and that you spelled its name is spelled correctly. Then try again.

HTTP Status Code: 404

#### MarksNotSupportedForFormatException

Speech marks are not supported for the `OutputFormat` selected. Speech marks are only available for content in `json` format.

HTTP Status Code: 400

#### ServiceFailureException

An unknown condition has caused a service failure.

HTTP Status Code: 500

#### SsmlMarksNotSupportedForTextTypeException

SSML speech marks are not supported for plain text-type input.

HTTP Status Code: 400

#### TextLengthExceededException

The value of the "Text" parameter is longer than the accepted limits. For the `SynthesizeSpeech` API, the limit for input text is a maximum of 6000 characters total, of which no more than 3000 can be billed characters. For the `StartSpeechSynthesisTask` API, the maximum is 200,000 characters, of which no more than 100,000 can be billed characters. SSML tags are not counted as billed characters.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)

- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)



## SynthesizeSpeech

Synthesizes UTF-8 input, plain text or SSML, to a stream of bytes. SSML input must be valid, well-formed SSML. Some alphabets might not be available with all the voices (for example, Cyrillic might not be read at all by English voices) unless phoneme mapping is used. For more information, see [How it Works](#).

### 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 Request Parameters

The request does not use any URI parameters.

### Request Body

The request accepts the following data in JSON format.

#### [Engine \(p. 211\)](#)

Specifies the engine (`standard` or `neural`) for Amazon Polly to use when processing input text for speech synthesis. Using a voice that is not supported for the engine selected will result in an error.

Type: String

Valid Values: `standard` | `neural`

Required: No

#### [LanguageCode \(p. 211\)](#)

Optional language code for the Synthesize Speech request. This is only necessary if using a bilingual voice, such as Aditi, which can be used for either Indian English (`en-IN`) or Hindi (`hi-IN`).

If a bilingual voice is used and no language code is specified, Amazon Polly will use the default language of the bilingual voice. The default language for any voice is the one returned by the [DescribeVoices](#) operation for the `LanguageCode` parameter. For example, if no language code is specified, Aditi will use Indian English rather than Hindi.

Type: String

Valid Values: `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`

Required: No

#### [LexiconNames \(p. 211\)](#)

List of one or more pronunciation lexicon names you want the service to apply during synthesis. Lexicons are applied only if the language of the lexicon is the same as the language of the voice. For information about storing lexicons, see [PutLexicon](#).

Type: Array of strings

Array Members: Maximum number of 5 items.

Pattern: `[0-9A-Za-z]{1,20}`

Required: No

#### [OutputFormat \(p. 211\)](#)

The format in which the returned output will be encoded. For audio stream, this will be mp3, ogg\_vorbis, or pcm. For speech marks, this will be json.

When pcm is used, the content returned is audio/pcm in a signed 16-bit, 1 channel (mono), little-endian format.

Type: String

Valid Values: `json` | `mp3` | `ogg_vorbis` | `pcm`

Required: Yes

#### [SampleRate \(p. 211\)](#)

The audio frequency specified in Hz.

The valid values for mp3 and ogg\_vorbis are "8000", "16000", "22050", and "24000". The default value for standard voices is "22050". The default value for neural voices is "24000".

Valid values for pcm are "8000" and "16000". The default value is "16000".

Type: String

Required: No

#### [SpeechMarkTypes \(p. 211\)](#)

The type of speech marks returned for the input text.

Type: Array of strings

Array Members: Maximum number of 4 items.

Valid Values: `sentence` | `ssml` | `viseme` | `word`

Required: No

#### [Text \(p. 211\)](#)

Input text to synthesize. If you specify `ssml` as the `TextType`, follow the SSML format for the input text.

Type: String

Required: Yes

#### [TextType \(p. 211\)](#)

Specifies whether the input text is plain text or SSML. The default value is plain text. For more information, see [Using SSML](#).

Type: String

Valid Values: `ssml` | `text`

Required: No

[Voiceld \(p. 211\)](#)

Voice ID to use for the synthesis. You can get a list of available voice IDs by calling the [DescribeVoices](#) operation.

Type: String

Valid Values: Aditi | Amy | Astrid | Bianca | Brian | Camila | Carla | Carmen | Celine | Chantal | Conchita | Cristiano | Dora | Emma | Enrique | Ewa | Filiz | Geraint | Giorgio | Gwyneth | Hans | Ines | Ivy | Jacek | Jan | Joanna | Joey | Justin | Karl | Kendra | Kimberly | Lea | Liv | Lotte | Lucia | Lupe | Mads | Maja | Marlene | Mathieu | Matthew | Maxim | Mia | Miguel | Mizuki | Naja | Nicole | Penelope | Raveena | Ricardo | Ruben | Russell | Salli | Seoyeon | Takumi | Tatyana | Vicki | Vitoria | Zeina | Zhiyu

Required: Yes

## Response Syntax

```
HTTP/1.1 200
Content-Type: ContentType
x-amzn-RequestCharacters: RequestCharacters

AudioStream
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The response returns the following HTTP headers.

[ContentType \(p. 213\)](#)

Specifies the type audio stream. This should reflect the `OutputFormat` parameter in your request.

- If you request `mp3` as the `OutputFormat`, the `ContentType` returned is `audio/mpeg`.
- If you request `ogg_vorbis` as the `OutputFormat`, the `ContentType` returned is `audio/ogg`.
- If you request `pcm` as the `OutputFormat`, the `ContentType` returned is `audio/pcm` in a signed 16-bit, 1 channel (mono), little-endian format.
- If you request `json` as the `OutputFormat`, the `ContentType` returned is `audio/json`.

[RequestCharacters \(p. 213\)](#)

Number of characters synthesized.

The response returns the following as the HTTP body.

[AudioStream \(p. 213\)](#)

Stream containing the synthesized speech.

## Errors

### EngineNotSupportedException

This engine is not compatible with the voice that you have designated. Choose a new voice that is compatible with the engine or change the engine and restart the operation.

HTTP Status Code: 400

### InvalidSampleRateException

The specified sample rate is not valid.

HTTP Status Code: 400

### InvalidSsmlException

The SSML you provided is invalid. Verify the SSML syntax, spelling of tags and values, and then try again.

HTTP Status Code: 400

### LanguageNotSupportedException

The language specified is not currently supported by Amazon Polly in this capacity.

HTTP Status Code: 400

### LexiconNotFoundException

Amazon Polly can't find the specified lexicon. This could be caused by a lexicon that is missing, its name is misspelled or specifying a lexicon that is in a different region.

Verify that the lexicon exists, is in the region (see [ListLexicons \(p. 199\)](#)) and that you spelled its name is spelled correctly. Then try again.

HTTP Status Code: 404

### MarksNotSupportedForFormatException

Speech marks are not supported for the `OutputFormat` selected. Speech marks are only available for content in `json` format.

HTTP Status Code: 400

### ServiceFailureException

An unknown condition has caused a service failure.

HTTP Status Code: 500

### SsmlMarksNotSupportedForTextTypeException

SSML speech marks are not supported for plain text-type input.

HTTP Status Code: 400

### TextLengthExceededException

The value of the "Text" parameter is longer than the accepted limits. For the `SynthesizeSpeech` API, the limit for input text is a maximum of 6000 characters total, of which no more than 3000 can be billed characters. For the `StartSpeechSynthesisTask` API, the maximum is 200,000 characters, of which no more than 100,000 can be billed characters. SSML tags are not counted as billed characters.

HTTP Status Code: 400

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

## Data Types

The following data types are supported:

- [Lexicon](#) (p. 216)
- [LexiconAttributes](#) (p. 217)
- [LexiconDescription](#) (p. 219)
- [SynthesisTask](#) (p. 220)
- [Voice](#) (p. 223)

## Lexicon

Provides lexicon name and lexicon content in string format. For more information, see [Pronunciation Lexicon Specification \(PLS\) Version 1.0](#).

## Contents

### Content

Lexicon content in string format. The content of a lexicon must be in PLS format.

Type: String

Required: No

### Name

Name of the lexicon.

Type: String

Pattern: `[0-9A-Za-z]{1,20}`

Required: No

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V3](#)

## LexiconAttributes

Contains metadata describing the lexicon such as the number of lexemes, language code, and so on. For more information, see [Managing Lexicons](#).

### Contents

#### Alphabet

Phonetic alphabet used in the lexicon. Valid values are `ipa` and `x-sampa`.

Type: String

Required: No

#### LanguageCode

Language code that the lexicon applies to. A lexicon with a language code such as "en" would be applied to all English languages (en-GB, en-US, en-AUS, en-WLS, and so on).

Type: String

Valid Values: `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`

Required: No

#### LastModified

Date lexicon was last modified (a timestamp value).

Type: Timestamp

Required: No

#### LexemesCount

Number of lexemes in the lexicon.

Type: Integer

Required: No

#### LexiconArn

Amazon Resource Name (ARN) of the lexicon.

Type: String

Required: No

#### Size

Total size of the lexicon, in characters.

Type: Integer

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V3](#)



## LexiconDescription

Describes the content of the lexicon.

### Contents

#### Attributes

Provides lexicon metadata.

Type: [LexiconAttributes](#) (p. 217) object

Required: No

#### Name

Name of the lexicon.

Type: String

Pattern: `[0-9A-Za-z]{1,20}`

Required: No

### See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V3](#)

# SynthesisTask

SynthesisTask object that provides information about a speech synthesis task.

## Contents

### CreationTime

Timestamp for the time the synthesis task was started.

Type: Timestamp

Required: No

### Engine

Specifies the engine (`standard` or `neural`) for Amazon Polly to use when processing input text for speech synthesis. Using a voice that is not supported for the engine selected will result in an error.

Type: String

Valid Values: `standard` | `neural`

Required: No

### LanguageCode

Optional language code for a synthesis task. This is only necessary if using a bilingual voice, such as Aditi, which can be used for either Indian English (`en-IN`) or Hindi (`hi-IN`).

If a bilingual voice is used and no language code is specified, Amazon Polly will use the default language of the bilingual voice. The default language for any voice is the one returned by the [DescribeVoices](#) operation for the `LanguageCode` parameter. For example, if no language code is specified, Aditi will use Indian English rather than Hindi.

Type: String

Valid Values: `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`

Required: No

### LexiconNames

List of one or more pronunciation lexicon names you want the service to apply during synthesis. Lexicons are applied only if the language of the lexicon is the same as the language of the voice.

Type: Array of strings

Array Members: Maximum number of 5 items.

Pattern: `[0-9A-Za-z]{1,20}`

Required: No

### OutputFormat

The format in which the returned output will be encoded. For audio stream, this will be `mp3`, `ogg_vorbis`, or `pcm`. For speech marks, this will be `json`.

Type: String

Valid Values: `json` | `mp3` | `ogg_vorbis` | `pcm`

Required: No

#### OutputUri

Pathway for the output speech file.

Type: String

Required: No

#### RequestCharacters

Number of billable characters synthesized.

Type: Integer

Required: No

#### SampleRate

The audio frequency specified in Hz.

The valid values for `mp3` and `ogg_vorbis` are "8000", "16000", "22050", and "24000". The default value for standard voices is "22050". The default value for neural voices is "24000".

Valid values for `pcm` are "8000" and "16000". The default value is "16000".

Type: String

Required: No

#### SnsTopicArn

ARN for the SNS topic optionally used for providing status notification for a speech synthesis task.

Type: String

Pattern: `^arn:aws(-(cn|iso(-b)?|us-gov))?:sns:[a-z0-9_-]{1,50}:\d{12}::[a-zA-Z0-9_-]{1,256}$`

Required: No

#### SpeechMarkTypes

The type of speech marks returned for the input text.

Type: Array of strings

Array Members: Maximum number of 4 items.

Valid Values: `sentence` | `ssml` | `viseme` | `word`

Required: No

#### TaskId

The Amazon Polly generated identifier for a speech synthesis task.

Type: String

Pattern: `^[a-zA-Z0-9_-]{1,100}$`

Required: No

### TaskStatus

Current status of the individual speech synthesis task.

Type: String

Valid Values: `scheduled` | `inProgress` | `completed` | `failed`

Required: No

### TaskStatusReason

Reason for the current status of a specific speech synthesis task, including errors if the task has failed.

Type: String

Required: No

### TextType

Specifies whether the input text is plain text or SSML. The default value is plain text.

Type: String

Valid Values: `ssml` | `text`

Required: No

### Voiceld

Voice ID to use for the synthesis.

Type: String

Valid Values: `Aditi` | `Amy` | `Astrid` | `Bianca` | `Brian` | `Camila` | `Carla` | `Carmen` | `Celine` | `Chantal` | `Conchita` | `Cristiano` | `Dora` | `Emma` | `Enrique` | `Ewa` | `Filiz` | `Geraint` | `Giorgio` | `Gwyneth` | `Hans` | `Ines` | `Ivy` | `Jacek` | `Jan` | `Joanna` | `Joey` | `Justin` | `Karl` | `Kendra` | `Kimberly` | `Lea` | `Liv` | `Lotte` | `Lucia` | `Lupe` | `Mads` | `Maja` | `Marlene` | `Mathieu` | `Matthew` | `Maxim` | `Mia` | `Miguel` | `Mizuki` | `Naja` | `Nicole` | `Penelope` | `Raveena` | `Ricardo` | `Ruben` | `Russell` | `Salli` | `Seoyeon` | `Takumi` | `Tatyana` | `Vicki` | `Vitoria` | `Zeina` | `Zhiyu`

Required: No

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V3](#)

## Voice

Description of the voice.

## Contents

### AdditionalLanguageCodes

Additional codes for languages available for the specified voice in addition to its default language.

For example, the default language for Aditi is Indian English (en-IN) because it was first used for that language. Since Aditi is bilingual and fluent in both Indian English and Hindi, this parameter would show the code `hi-IN`.

Type: Array of strings

Valid Values: `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`

Required: No

### Gender

Gender of the voice.

Type: String

Valid Values: `Female` | `Male`

Required: No

### Id

Amazon Polly assigned voice ID. This is the ID that you specify when calling the `SynthesizeSpeech` operation.

Type: String

Valid Values: `Aditi` | `Amy` | `Astrid` | `Bianca` | `Brian` | `Camila` | `Carla` | `Carmen` | `Celine` | `Chantal` | `Conchita` | `Cristiano` | `Dora` | `Emma` | `Enrique` | `Ewa` | `Filiz` | `Geraint` | `Giorgio` | `Gwyneth` | `Hans` | `Ines` | `Ivy` | `Jacek` | `Jan` | `Joanna` | `Joey` | `Justin` | `Karl` | `Kendra` | `Kimberly` | `Lea` | `Liv` | `Lotte` | `Lucia` | `Lupe` | `Mads` | `Maja` | `Marlene` | `Mathieu` | `Matthew` | `Maxim` | `Mia` | `Miguel` | `Mizuki` | `Naja` | `Nicole` | `Penelope` | `Raveena` | `Ricardo` | `Ruben` | `Russell` | `Salli` | `Seoyeon` | `Takumi` | `Tatyana` | `Vicki` | `Vitoria` | `Zeina` | `Zhiyu`

Required: No

### LanguageCode

Language code of the voice.

Type: String

Valid Values: `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`

Required: No

LanguageName

Human readable name of the language in English.

Type: String

Required: No

Name

Name of the voice (for example, Salli, Kendra, etc.). This provides a human readable voice name that you might display in your application.

Type: String

Required: No

SupportedEngines

Specifies which engines (`standard` or `neural`) that are supported by a given voice.

Type: Array of strings

Valid Values: `standard` | `neural`

Required: No

## See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V3](#)

# Amazon Polly 문서 기록

다음 표는 Amazon Polly 개발자 안내서의 각 릴리스에서 변경된 중요 사항에 대해 설명합니다. 이 설명서에 대한 업데이트 알림을 받으려면 RSS 피드를 구독하면 됩니다.

- 최신 설명서 업데이트: 2019년 11월 25일

update-history-change	update-history-description	update-history-date
<a href="#">새로운 기능 (p. 225)</a>	뉴스 진행자 말투 외에도, Amazon Polly은 이제 더 나은 텍스트 투 스피치 구절을 합성하는데 도움이 되는 두 번째 NTTS 말투를 제공합니다. 대화 스타일은 신경 시스템을 사용하여 많은 사용 사례에서 사용할 수 있는 보다 친근하고 표현적인 대화 스타일로 스피치를 생성합니다. 자세한 내용은 <a href="#">NTTS 말투</a> 를 참조하십시오.	November 25, 2019
<a href="#">새 음성이 추가됨 (p. 225)</a>	두 개의 새로운 음성, 즉 Camila(여성, 포르투갈어-브라질)와 Lupe(여성, 스페인어-미국)가 추가되었습니다.	October 23, 2019
<a href="#">새 기능이 추가됨 (p. 225)</a>	<a href="#">Windows용 Amazon Polly 플러그인</a> 을 추가하면 광범위한 Amazon Polly 음성을 Windows SAPI 규정 준수 애플리케이션에 통합할 수 있습니다.	September 26, 2019
<a href="#">새로운 주요 기능 (p. 225)</a>	출시 이후 Amazon Polly에서 지원하는 TTS(텍스트 투 스피치) 음성 외에도 이제 Amazon Polly는 훨씬 자연스러운 음성을 제공하도록 개선된 NTTS(신경 TTS) 시스템을 제공하게 되었으므로 최대한 자연스럽고 사람과 유사한 텍스트 투 스피치 음성을 제공할 수 있습니다. 자세한 내용은 <a href="#">신경 텍스트 투 스피치</a> 를 참조하십시오.	July 30, 2019
<a href="#">새 음성이 추가됨 (p. 225)</a>	새 음성인 Lucia(여성, 스페인어) 및 Bianca(여성, 이탈리아어)가 추가되었습니다.	August 2, 2018
<a href="#">새 언어가 추가됨 (p. 225)</a>	새 언어인 멕시코 스페인어(es-MX)가 추가되었습니다. 이 언어는 Mia의 여성 음성을 사용합니다.	August 2, 2018
<a href="#">새 언어가 추가됨 (p. 225)</a>	새 언어인 힌디어(hi-IN)가 추가되었습니다. 이 음성은 Aditi의 여성 음성을 사용합니다. Aditi는 인도 영어에도 사용되며, 이는 Aditi	August 2, 2018

	Amazon Polly의 첫 번째 이중 언어 음성입니다.	
새 기능이 추가됨 (p. 225)	긴 텍스트 구절의 <a href="#">스피치 합성</a> 이 추가되었습니다(최대 100,000자).	July 17, 2018
새 SSML 기능이 추가됨 (p. 225)	합성된 <a href="#">스피치</a> 의 <a href="#">최대 기간</a> 이 추가되었습니다.	July 17, 2018
새 음성이 추가됨 (p. 225)	새 음성인 Léa(여성, 프랑스어)가 추가되었습니다.	June 5, 2018
확장된 기능 (p. 225)	Amazon Translate 기능의 추가를 비롯한 Amazon Polly WordPress 플러그인 기능이 확장되었습니다.	June 4, 2018
새 언어가 추가됨 (p. 225)	새 언어인 한국어(ko-KR)가 추가되었습니다.	June 4, 2018
리전 확장 (p. 225)	Amazon Polly 서비스가 모든 상용 리전으로 확장되었습니다.	June 4, 2018
새 음성이 추가됨 (p. 225)	두 개의 새 음성인 Aditi(여성, 인도 영어) 및 Seoyeon(여성, 한국어)이 추가되었습니다.	November 15, 2017
새로운 기능 (p. 225)	새로운 <a href="#">스피치 마크</a> 기능이 추가되었으며, <a href="#">SSML</a> 기능이 확장되었습니다.	April 19, 2017
새 설명서 (p. 225)	이 문서는 첫 번째 Amazon Polly 개발자 안내서 릴리스입니다.	November 30, 2016



# AWS Glossary

For the latest AWS terminology, see the [AWS Glossary](#) in the AWS General Reference.