



Panduan Developer

Amazon Polly



Amazon Polly: Panduan Developer

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

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

Table of Contents

Apa itu Amazon Polly?	1
Cara kerjanya	1
Manfaat	2
Apakah Anda pengguna baru?	3
Bekerja dengan AWS SDKs	3
Memulai	5
Mendaftar untuk AWS	5
Mendaftar untuk Akun AWS	6
Menyiapkan AWS CLI	6
Mengkonfigurasi ulang AWS CLI	7
Contoh ucapan sintesis	9
Streaming dua arah	13
SynthesizeSpeech dan StartSpeechSynthesisStream dibandingkan	13
Mengirim teks dan menerima audio	15
Buka aliran	15
Kirim teks	15
Menerima audio	16
Tutup aliran	17
Contoh kode	17
Suara di Amazon Polly	27
Suara yang tersedia	27
Suara merek	35
Suara bilingual	35
Suara dwibahasa beraksen	35
Suara dwibahasa sepenuhnya	36
Menerapkan suara penyiar berita	37
Mendengarkan suara	40
Pengaturan waktu kecepatan suara	40
Mengubah kecepatan suara	41
Bahasa di Amazon Polly	43
Arab (arb)	46
Arab (Teluk) (Ar-ae)	51
Catalan (CA-es)	57
Tionghoa (Kanton) (Yue-cn)	61

Mandarin (Mandarin) (Cmn-cn)	66
Ceko (cs-cz)	70
Denmark (da-dk)	74
Belanda (Belgia) (NL-be)	79
Belanda (nl-NL)	83
Inggris (AS) (en-US)	86
Inggris (Australia) (en-AU)	90
Inggris (Inggris) (en-GB)	95
Inggris (India) (en-in)	99
Inggris (Irlandia) (en-IE)	103
Inggris (Selandia Baru) (en-NZ)	107
Inggris (Singapura) (en-SG)	113
Inggris (Afrika Selatan) (en-ZA)	117
Bahasa Inggris (Welsh) (en-GB-WLS)	122
Finlandia (Fi-Fi)	126
Perancis (FR-fr)	132
Prancis (Belgia) (FR-be)	135
Prancis (Kanada) (fr-Ca)	139
Jerman (De-de)	143
Jerman (Austria) (de-at)	147
Jerman (standar Swiss) (De-ch)	152
Hindi (Hi-in)	157
Islandia (IS-IS)	160
Italia (IT-it)	165
Jepang (Ja-jp)	168
Korea (KO-kr)	172
Norwegia (NB-No)	175
Polandia (PL-PL)	179
Portugis (Pt-PT)	183
Portugis (Brasil) (pt-BR)	186
Rumania (Ro-ro)	190
Rusia (Ru-ru)	193
Spanyol (ES-es)	197
Spanyol (Meksiko) (es-Mx)	200
Spanyol (AS) (es-AS)	203
Swedia (SV-se)	206

Turki (Tr-tr)	211
Welsh (cy-GB)	215
Mesin suara	220
Mesin generatif	220
Suara generatif yang tersedia	221
Kompatibilitas fitur dan wilayah	223
Long-form mesin	225
Tersedia suara bentuk panjang	226
Kompatibilitas fitur dan wilayah	226
Mesin saraf	227
Suara saraf yang tersedia	228
Kompatibilitas fitur dan wilayah	231
Mesin standar	233
Tersedia suara Standar	233
Kompatibilitas fitur dan wilayah	236
Memilih mesin suara	237
Tanda bicara	239
Jenis tanda bicara	239
Visemes dan Amazon Polly	240
Output tanda ucapan	241
Meminta tanda pidato	242
Tanda bicara tanpa contoh SSML	244
Tanda ucapan dengan contoh SSML	245
Menggunakan SSML	247
Karakter yang dipesan	248
Menggunakan SSML di konsol	251
Menggunakan SSML dengan perintah Synthesize-Speech	252
Mensintesis dokumen yang disempurnakan SSML	253
Tag SSML yang didukung	254
Mengidentifikasi teks yang disempurnakan SSML	257
Menambahkan jeda	257
Menekankan kata-kata	258
Menentukan bahasa lain untuk kata-kata tertentu	259
Menempatkan tag khusus dalam teks Anda	260
Menambahkan jeda antar paragraf	261
Menggunakan pengucapan fonetik	261

Mengontrol volume, kecepatan berbicara, dan nada	263
Menetapkan durasi maksimum untuk ucapan yang disintesis	266
Menambahkan jeda di antara kalimat	269
Mengontrol bagaimana jenis kata khusus diucapkan	270
Mengucapkan akronim dan singkatan	274
Meningkatkan pengucapan dengan menentukan bagian pidato	274
Menambahkan suara pernapasan	276
Gaya berbicara penyiar berita	280
Menambahkan kompresi rentang dinamis	280
Berbicara dengan lembut	283
Mengontrol timbre	283
Berbisik	285
Mengelola leksikon	287
Menggunakan beberapa leksikon	288
Mengunggah leksikon	289
Menerapkan leksikon (Synthesizing Speech)	295
Memfilter daftar leksikon di konsol	298
Mengunduh leksikon di konsol	300
Menghapus leksikon	302
File audio panjang	303
Menyiapkan kebijakan IAM untuk sintesis asinkron	304
Membuat file audio yang panjang	304
Kuota	309
Wilayah yang didukung	310
Kuota dan laju throttle	310
Permintaan bersamaan	311
Praktik terbaik untuk mengurangi pelambatan	311
Leksikon pengucapan	312
SynthesizeSpeech Operasi API	312
SpeechSynthesisTask Operasi API	313
Bahasa Markup Sintesis Pidato (SSML)	313
Contoh kode dan aplikasi	315
Sampel Java	315
DeleteLexicon	316
DescribeVoices	318
GetLexicon	319

ListLexicons	321
PutLexicon	322
StartSpeechSynthesisTask	325
Tanda Pidato	328
SynthesizeSpeech	331
Sampel Python	333
DeleteLexicon	333
GetLexicon	334
ListLexicon	336
PutLexicon	336
StartSpeechSynthesisTask	338
SynthesizeSpeech	338
Contoh Java	339
Contoh Python	344
Contoh Python: index.html	346
Contoh Python: server.py	350
Contoh iOS	357
Contoh Android	359
Contoh kode	362
Hal-hal mendasar	363
Tindakan	363
Skenario	414
Mengonversi teks menjadi ucapan dan kembali ke teks	415
Buat aplikasi lip-sync	416
Buat aplikasi untuk menganalisis umpan balik pelanggan	416
Memulai dengan sintesis text-to-speech	423
Keamanan	430
Perlindungan Data	431
Enkripsi saat Data Tidak Berpindah	431
Enkripsi Saat Data Berpindah	432
Privasi Lalu Lintas Kerja Internet	432
Identity and Access Management	432
Audiens	432
Mengautentikasi dengan identitas	433
Mengelola akses menggunakan kebijakan	434
Bagaimana Amazon Polly bekerja dengan IAM	436

Identity-based contoh kebijakan	442
Referensi Izin API Amazon Polly	449
Pemecahan masalah	451
Pembuatan Log dan Pemantauan	453
Validasi Kepatuhan	453
Ketahanan	454
Keamanan Infrastruktur	455
Praktik Terbaik Keamanan	455
Menggunakan Endpoint VPC Antarmuka	455
Ketersediaan	456
Membuat titik akhir VPC untuk Amazon Polly	456
Menguji koneksi antara VPC Anda dan Amazon Polly	457
Mengontrol akses ke titik akhir Amazon Polly Anda	457
Support untuk kunci konteks VPC	458
Mencatat panggilan API Amazon Polly dengan AWS CloudTrail	459
Informasi Amazon Polly di CloudTrail	459
Contoh: Entri File Log Amazon Polly	460
CloudWatch integrasi	462
Mendapatkan CloudWatch Metrik (Konsol)	462
Mendapatkan CloudWatch metrik di AWS CLI	462
Metrik Amazon Polly	463
Dimensi untuk Metrik Amazon Polly	464
Referensi API	466
Tindakan	466
DeleteLexicon	467
DescribeVoices	469
GetLexicon	473
GetSpeechSynthesisTask	476
ListLexicons	479
ListSpeechSynthesisTasks	482
PutLexicon	485
StartSpeechSynthesisStream	488
StartSpeechSynthesisTask	494
SynthesizeSpeech	503
Tipe Data	510
AudioEvent	511

CloseStreamEvent	512
FlushStreamConfiguration	513
Lexicon	514
LexiconAttributes	515
LexiconDescription	517
StartSpeechSynthesisStreamActionStream	518
StartSpeechSynthesisStreamEventStream	519
StreamClosedEvent	521
SynthesisTask	522
TextEvent	527
ThrottlingReason	528
ValidationExceptionField	529
Voice	530
Jenis Kesalahan Umum	532
Parameter Umum	535
Riwayat Dokumen	538
.....	dlvi

Apa itu Amazon Polly?

Amazon Polly adalah layanan cloud yang mengubah teks menjadi ucapan yang hidup. Anda dapat menggunakan Amazon Polly untuk mengembangkan aplikasi yang meningkatkan keterlibatan dan aksesibilitas. Amazon Polly mendukung berbagai bahasa dan mencakup berbagai suara yang hidup. Dengan Amazon Polly, Anda dapat membuat aplikasi yang mendukung ucapan yang berfungsi di beberapa lokasi dan menggunakan suara yang ideal untuk pelanggan Anda. Juga, Anda hanya membayar untuk teks yang Anda sintesis. Anda juga dapat men-cache dan memutar ulang pidato yang dihasilkan Amazon Polly tanpa biaya tambahan.

Amazon Polly menawarkan banyak opsi suara, termasuk opsi generatif, bentuk panjang, saraf, dan standar text-to-speech (TTS). Suara-suara ini memberikan peningkatan terobosan dalam kualitas bicara menggunakan teknologi pembelajaran mesin baru untuk menawarkan suara yang paling alami dan mirip manusia text-to-speech. Teknologi Neural TTS juga mendukung gaya berbicara Newscaster, yang disesuaikan dengan kasus penggunaan narasi berita.

Kasus penggunaan umum untuk Amazon Polly termasuk, tetapi tidak terbatas pada: aplikasi seluler seperti pembaca berita, game, platform eLearning, aplikasi aksesibilitas untuk orang tunanetra, dan segmen Internet of Things (IoT) yang berkembang pesat.

Amazon Polly disertifikasi untuk digunakan dengan beban kerja yang diatur untuk HIPAA (Undang-Undang Portabilitas dan Akuntabilitas Asuransi Kesehatan tahun 1996), dan Standar Keamanan Data Industri Kartu Pembayaran (PCI DSS).

Bagaimana Amazon Polly bekerja

Amazon Polly mengubah teks input menjadi ucapan yang hidup. Untuk menggunakan suara Amazon Polly, pilih [mesin suara](#), panggil metode sintesis ucapan, berikan teks yang ingin Anda sintesis, lalu tentukan format output audio. Amazon Polly kemudian mensintesis teks yang disediakan ke dalam aliran audio ucapan berkualitas tinggi.

- Teks masukan — Berikan teks yang ingin Anda sintesis, dan Amazon Polly mengembalikan aliran audio. Anda dapat memberikan masukan sebagai plaintext atau dalam format Speech Synthesis Markup Language (SSML). Dengan SSML Anda dapat mengontrol berbagai aspek bicara, seperti pengucapan, volume, nada, dan kecepatan bicara. Untuk informasi selengkapnya, lihat [Menghasilkan pidato dari dokumen SSML](#).
- Suara yang tersedia - Amazon Polly menyediakan portofolio bahasa dan berbagai suara, termasuk suara bilingual (untuk bahasa Inggris dan Hindi). Untuk sebagian besar bahasa, Anda dapat

memilih dari beberapa suara, baik pria maupun wanita. Saat meluncurkan tugas sintesis ucapan, Anda menentukan ID suara, lalu Amazon Polly menggunakan suara ini untuk mengonversi teks menjadi ucapan. Amazon Polly bukanlah layanan penerjemahan—pidato yang disintesis dalam bahasa yang sama dengan teks. Angka yang direpresentasikan sebagai digit (misalnya, 53, bukan lima puluh tiga) disintesis dalam bahasa suara dan bukan teks. Untuk informasi selengkapnya, lihat [Suara di Amazon Polly](#).

- **Format keluaran** — Amazon Polly dapat menyampaikan pidato yang disintesis dalam berbagai format. Anda dapat memilih format audio yang sesuai dengan kebutuhan Anda. Misalnya, Anda dapat meminta pidato dalam format Ogg Vorbis MP3 atau Ogg untuk dikonsumsi oleh aplikasi web dan seluler. Atau, Anda dapat meminta format output PCM untuk konsumsi oleh AWS IoT perangkat dan solusi telepon. Untuk aplikasi telepon, Anda dapat menggunakan format MU-law atau A-law.

Note

Untuk mendengar contoh suara Amazon Polly di browser Anda, lihat ikhtisar produk [Amazon Polly](#).

Manfaat

Beberapa manfaat menggunakan Amazon Polly meliputi:

- **Kualitas tinggi** - Amazon Polly menawarkan suara generatif, bentuk panjang, saraf, dan berkualitas tinggi (TTS) yang berkinerja tinggi. text-to-speech Teknologi ini mensintesis ucapan alami dengan akurasi pengucapan yang tinggi (termasuk singkatan, ekspansi akronim, interpretasi tanggal/waktu, dan disambiguasi homograf).
- **Latensi rendah** - Amazon Polly mencapai respons cepat, yang menjadikannya opsi yang layak untuk kasus penggunaan latensi rendah seperti sistem dialog.
- **Dukungan untuk portofolio besar bahasa dan suara** - Amazon Polly mendukung lusinan suara dan bahasa, menawarkan opsi suara pria dan wanita untuk sebagian besar bahasa. Jumlah ini akan terus meningkat karena kami menghadirkan lebih banyak suara saraf online. Suara Inggris AS Matthew dan Joanna juga dapat menggunakan gaya berbicara Neural Newscaster, mirip dengan apa yang mungkin Anda dengar dari pembawa berita profesional.

- Hemat biaya - pay-per-use Model Amazon Polly berarti tidak ada biaya penyiapan. Mulai dari yang kecil dan tingkatkan saat aplikasi Anda tumbuh.
- Solusi berbasis cloud — Solusi TTS di perangkat memerlukan sumber daya komputasi yang signifikan, terutama daya CPU, RAM, dan ruang disk. Ini dapat menghasilkan biaya pengembangan yang lebih tinggi dan konsumsi daya yang lebih tinggi pada perangkat seperti tablet, smartphone, dan sebagainya. Sebaliknya, konversi TTS dilakukan secara AWS Cloud dramatis mengurangi kebutuhan sumber daya lokal. Ini memungkinkan dukungan semua bahasa dan suara yang tersedia dengan kualitas luar biasa. Selain itu, peningkatan ucapan langsung tersedia untuk semua pengguna akhir dan tidak memerlukan pembaruan tambahan untuk perangkat.

Note

Untuk mendengar contoh suara Amazon Polly di browser Anda, lihat ikhtisar produk [Amazon Polly](#).

Apakah Anda pengguna baru?

Jika Anda adalah pengguna pertama kali Amazon Polly, kami sarankan Anda membaca bagian berikut dalam urutan yang tercantum:

1. [Bagaimana Amazon Polly bekerja](#)— Bagian ini memperkenalkan berbagai input dan opsi Amazon Polly yang dapat Anda gunakan untuk menciptakan pengalaman sederhana.
2. [Memulai dengan Amazon Polly](#)— Di bagian ini, Anda mengatur akun Anda dan menguji sintesis ucapan Amazon Polly.
3. [Contoh kode dan aplikasi untuk Amazon Polly](#)— Bagian ini memberikan contoh tambahan yang dapat Anda gunakan untuk menjelajahi Amazon Polly.

Menggunakan Amazon Polly dengan SDK AWS

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

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

Ketersediaan contoh

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

Memulai dengan Amazon Polly

Amazon Polly menyediakan beberapa operasi API yang dapat Anda integrasikan dengan mudah dengan aplikasi yang ada. Untuk daftar operasi yang didukung, lihat [Tindakan](#).

Anda dapat melakukan hampir semua operasi yang sama di konsol Amazon Polly dan AWS CLI. Namun, Anda tidak dapat mendengarkan pidato yang disintesis di AWS CLI. Untuk bekerja dengan audio di AWS CLI, simpan teks Anda ke file. Kemudian buka file dalam aplikasi audio pilihan Anda.

Anda dapat menggunakan salah satu dari opsi berikut:

- **AWS SDK** — Saat menggunakan SDK, permintaan Anda ke Amazon Polly secara otomatis ditandatangani dan diautentikasi menggunakan kredensial yang Anda berikan. Ini adalah pilihan yang disarankan untuk membangun aplikasi Anda.
- **AWS CLI** — Anda dapat menggunakan AWS CLI untuk menggunakan Amazon Polly tanpa menulis kode apa pun.

Sebelum Anda menggunakan Amazon Polly untuk pertama kalinya, Anda harus mendaftar. AWS Saat Anda mendaftar ke Amazon Web Services (AWS), AWS akun Anda secara otomatis mendaftar untuk semua layanan AWS, termasuk Amazon Polly. Anda hanya dikenakan biaya untuk layanan dan sumber daya yang Anda gunakan. Jika Anda adalah AWS pelanggan baru, Anda dapat memulai Amazon Polly tanpa biaya. Untuk informasi selengkapnya, lihat [Tingkat Penggunaan Gratis AWS](#).

Bagian berikut menjelaskan cara memulai menggunakan Amazon Polly.

Topik

- [Mendaftar untuk AWS](#)
- [Menyiapkan AWS CLI](#)
- [Mengkonfigurasi ulang AWS CLI](#)

Mendaftar untuk AWS

Sebelum Anda dapat menggunakan AWS layanan apa pun, termasuk Amazon Polly, Anda harus mendaftar. AWS

Mendaftar untuk Akun AWS

Untuk memulai AWS, Anda membutuhkan Akun AWS. Untuk informasi tentang membuat Akun AWS, lihat [Memulai dengan Akun AWS](#) di Panduan AWS Account Management Referensi.

Untuk informasi selengkapnya tentang IAM, lihat berikut ini:

- [AWS Identity and Access Management \(IAM\)](#)
- [Memulai dengan IAM](#)
- [Panduan Pengguna IAM](#)

Note

Catat ID AWS akun Anda. Anda akan membutuhkannya di langkah selanjutnya.

Menyiapkan AWS CLI

Ikuti langkah-langkah ini untuk mengunduh dan mengonfigurasi AWS CLI agar berfungsi dengan Amazon Polly.

Untuk mengatur AWS Command Line Interface

1. Unduh dan konfigurasikan AWS CLI. Untuk melakukannya, lihat topik berikut di Panduan Pengguna AWS Command Line Interface :
 - [Mendapatkan Set Up dengan AWS Command Line Interface](#)
 - [Mengkonfigurasi AWS Command Line Interface](#)
2. Tambahkan profil bernama untuk pengguna administrator di file AWS CLI AWS Config. Anda dapat menggunakan profil ini saat menjalankan AWS CLI perintah. Untuk informasi selengkapnya tentang profil yang diberi nama, lihat [Profil yang Diberi Nama](#) dalam Panduan Pengguna AWS Command Line Interface .

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

Untuk daftar AWS Wilayah yang tersedia dan yang didukung oleh Amazon Polly, lihat [Wilayah dan Titik Akhir](#) di. Referensi Umum Amazon Web Services

 Note

Jika Anda menggunakan Wilayah yang didukung oleh Amazon Polly yang Anda tentukan saat mengonfigurasinya AWS CLI, hilangkan baris berikut dari contoh kode. AWS CLI

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

3. Verifikasi pengaturan dengan mengetikkan perintah bantuan berikut pada prompt perintah.

```
aws help
```

Daftar AWS perintah yang valid akan muncul di AWS CLI jendela.

Mengkonfigurasi ulang AWS CLI

Jika sebelumnya Anda telah mengunduh dan mengonfigurasinya AWS CLI, Amazon Polly mungkin tidak tersedia kecuali Anda mengonfigurasi ulang. AWS CLI Prosedur berikut memeriksa untuk melihat apakah ini perlu.

Untuk mengaktifkan kembali Amazon Polly dari AWS CLI

1. Verifikasi ketersediaan Amazon Polly dengan mengetikkan perintah bantuan berikut di prompt AWS CLI perintah.

```
aws polly help
```

Jika Anda melihat deskripsi Amazon Polly dan daftar perintah yang valid muncul di AWS CLI jendela, Anda dapat langsung menggunakan Amazon Polly. AWS CLI Dalam hal ini, Anda dapat melewati sisa prosedur ini. Jika ini tidak ditampilkan, lanjutkan dengan Langkah 2.

2. Aktifkan Amazon Polly menggunakan salah satu dari dua opsi berikut:
 - a. Copot pemasangan dan instal ulang file. AWS CLI

Untuk petunjuk, lihat [Menginstal AWS Command Line Interface](#) di Panduan AWS Command Line Interface Pengguna.

atau

- b. Unduh file [layanan-2.json](#).

Pada prompt perintah, jalankan perintah berikut:

```
aws configure add-model --service-model file://service-2.json --service-name  
polly
```

3. Verifikasi ulang ketersediaan Amazon Polly.

```
aws polly help
```

Deskripsi Amazon Polly harus terlihat.

Mensintesis pidato dengan contoh Amazon Polly

Halaman ini menyajikan contoh sintesis ucapan singkat yang dilakukan di konsol, the AWS CLI, dan dengan Python. Contoh ini melakukan sintesis ucapan dari teks biasa, bukan SSML.

Console

Sintesis ucapan di konsol

1. Masuk ke AWS Management Console dan buka konsol Amazon Polly di. <https://console.aws.amazon.com/polly/>
2. Pilih tab Text-to-Speech. Bidang teks akan dimuat dengan contoh teks sehingga Anda dapat dengan cepat mencoba Amazon Polly.
3. Matikan SSML.
4. Ketik atau tempel teks ini ke dalam kotak input.

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

5. Di bawah Engine, pilih Generative, Long Form, Neural, atau Standard.
6. Pilih bahasa dan AWS Wilayah, lalu pilih suara. (Jika Anda memilih Neural for Engine, hanya bahasa dan suara yang mendukung NTTTS yang tersedia. Semua suara Standar dan Formulir Panjang dinonaktifkan.)
7. Untuk segera mendengarkan pidato, pilih Dengarkan.
8. Untuk menyimpan pidato ke file, lakukan salah satu hal berikut:
 - a. Pilih Unduh.
 - b. Untuk mengubah ke format file yang berbeda, perluas Pengaturan tambahan, aktifkan Pengaturan format file ucapan, pilih format file yang Anda inginkan (seperti MP3, OGG, PCM, MU-law, atau A-law), lalu pilih Unduh.

AWS CLI

Dalam latihan ini, Anda memanggil SynthesizeSpeech operasi dengan melewati teks input. Anda dapat menyimpan audio yang dihasilkan sebagai file dan memverifikasi kontennya.

1. Jalankan `synthesize-speech` AWS CLI perintah untuk mensintesis teks sampel ke file audio (`hello.mp3`).

AWS CLI Contoh berikut diformat untuk Unix, Linux, dan macOS. Untuk Windows, ganti karakter kelanjutan backslash (\) Unix di akhir setiap baris dengan tanda sisipan (^) dan gunakan tanda kutip penuh (") di sekitar teks input dengan tanda kutip tunggal (') untuk tag interior.

```
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
```

Dalam panggilan `synthesize-speech`, Anda memberikan contoh teks untuk disintesis dengan suara pilihan Anda. Anda harus memberikan ID suara (dijelaskan pada langkah berikut) dan format output. Perintah menyimpan audio yang dihasilkan ke `hello.mp3` file. Selain MP3 file, operasi mengirimkan output berikut ke konsol.

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

2. Putar `hello.mp3` file yang dihasilkan untuk memverifikasi ucapan yang disintesis.

Python

Untuk menguji kode contoh Python, Anda memerlukan kode. AWS SDK for Python (Boto) Untuk instruksi, lihat [AWS SDK untuk Python \(Boto3\)](#).

Kode Python dalam contoh ini melakukan tindakan berikut:

- Memanggil AWS SDK for Python (Boto) untuk mengirim `SynthesizeSpeech` permintaan ke Amazon Polly (dengan memberikan beberapa teks sebagai input).
- Mengakses aliran audio yang dihasilkan dalam respons dan menyimpan audio ke file (`speech.mp3`) di disk lokal Anda.
- Memutar file audio dengan pemutar audio default untuk sistem lokal Anda.

Simpan kode ke file (example.py) dan jalankan.

```
"""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])
```

Untuk contoh lebih mendalam, lihat topik berikut:

- [Menggunakan SSML di konsol](#)
- [Menerapkan leksikon \(Synthesizing Speech\)](#)
- [Contoh kode dan aplikasi untuk Amazon Polly](#)

Mensintesis ucapan dengan streaming dua arah

Amazon Polly menyediakan `StartSpeechSynthesisStream` operasi yang membuat HTTP/2 koneksi dengan komunikasi dua arah antara aplikasi Anda dan layanan. Teks mengalir dari aplikasi Anda ke Amazon Polly sementara audio yang disintesis mengalir kembali. Anda mengirim teks saat tersedia, dan Amazon Polly mengembalikan audio saat disintesis, tanpa kedua sisi menunggu yang lain selesai.

Ini berguna ketika teks diproduksi secara progresif daripada sekaligus. Misalnya, chatbot layanan pelanggan yang didukung oleh model foundation di Amazon Bedrock menghasilkan token responsnya dengan token. Dengan streaming dua arah, aplikasi Anda dapat meneruskan setiap potongan teks ke Amazon Polly saat model menghasilkannya dan mulai memutar audio kembali ke pemanggil saat model masih menghasilkan respons lainnya.

Operasi ini membutuhkan mesin generatif dan AWS SDK yang mendukung aliran HTTP/2 peristiwa. Audio tiba sebagai urutan potongan yang diakumulasikan aplikasi Anda menjadi output audio yang lengkap. Tanda bicara tidak didukung oleh operasi ini.

Note

AWS CLI (v1 dan v2), AWS Alat untuk PowerShell (v4 dan v5), Python, dan .NET v3 tidak didukung. Anda dapat menggunakan API streaming dua arah dengan SDK berikut: SDK for AWS Java 2.x JavaScript , v3, .NET v4, C++, Go v2, Kotlin, PHP v3, Ruby v3, Rust, dan Swift.

Topik

- [SynthesizeSpeech dan StartSpeechSynthesisStream dibandingkan](#)
- [Mengirim teks dan menerima audio](#)
- [Contoh kode](#)

SynthesizeSpeech dan StartSpeechSynthesisStream dibandingkan

[SynthesizeSpeech](#) adalah operasi permintaan-respons. Anda memberikan teks lengkap dalam satu permintaan dan menerima audio yang disintesis penuh dalam satu respons. Ini mendukung semua mesin (standar, saraf, bentuk panjang, generatif), semua format output termasuk tanda bicara,

dan memiliki batas teks 6.000 karakter total (yang tidak lebih dari 3.000 karakter dapat ditagih) per permintaan. Respons mengalirkan audio kembali segera setelah byte pertama tersedia. Gunakan operasi ini ketika Anda memiliki semua teks yang tersedia di muka.

[StartSpeechSynthesisStream](#) adalah operasi streaming dua arah. Ini membuka HTTP/2 koneksi di mana Anda mengirim teks secara bertahap dan menerima audio saat disintesis. Tidak ada batas teks per permintaan karena teks dialirkan terus menerus. Ini membutuhkan mesin generatif dan tidak mendukung tanda bicara. Gunakan operasi ini ketika teks tiba secara bertahap dan Anda ingin output audio dimulai sebelum semua input tersedia. Skenario umum meliputi:

- AI percakapan dan asisten suara. Model bahasa besar menghasilkan teks respons dalam potongan kecil (token). Teruskan setiap potongan teks ke Amazon Polly saat tiba sehingga pengguna mendengar ucapan saat model masih menghasilkan.
- Real-time terjemahan. Sebuah sistem terjemahan menghasilkan teks yang diterjemahkan segmen demi segmen. Streaming setiap segmen untuk sintesis tanpa menunggu terjemahan lengkap selesai.
- Long-form konten melebihi SynthesizeSpeech batas. Teks yang lebih panjang dari 6.000 karakter dapat dialirkan terus menerus tanpa membagi menjadi beberapa permintaan atau mengelola batas potongan.

Perbandingan SynthesizeSpeech dan StartSpeechSynthesisStream

Aspek	SynthesizeSpeech	StartSpeechSynthesisStream
Protokol	Request-response	Aliran acara dua arah () HTTP/2
Pengiriman teks	Teks lengkap di badan permintaan	Streaming masukan teks melalui TextEvent pesan
Pengiriman audio	Streaming respons audio melalui badan respons HTTP	Streaming respons audio melalui AudioEvent pesan
Dukungan mesin	standar, saraf, bentuk panjang, generatif	generatif saja
Dukungan SSML	Ya (semua mesin; tag yang didukung bervariasi menurut mesin)	Ya (hanya tag mesin generatif)

Aspek	SynthesizeSpeech	StartSpeechSynthesisStream
Leksikon	Ya	Ya
Tanda bicara	Ya	Tidak
Batas teks	6.000 karakter total (3.000 ditagih) per permintaan	6.000 karakter total (3.000 tagihan) per TextEvent
AWS CLI dukungan	Ya	Tidak (streaming dua arah membutuhkan SDK)

Mengirim teks dan menerima audio

Sesi streaming dua arah melibatkan membuka koneksi, mengirim teks dan menerima audio secara bersamaan, kemudian menutup aliran saat input selesai. Bagian berikut menjelaskan setiap fase secara rinci.

Buka aliran

Aplikasi Anda memanggil [StartSpeechSynthesisStream](#) operasi melalui SDK, menentukan parameter sintesis (Engine, VoiceId, dan opsional OutputFormatLanguageCode,). LexiconNames SampleRate SDK membuat HTTP/2 koneksi dan aliran dua arah siap menerima peristiwa masukan.

Kirim teks

Klien mengirim satu atau lebih [TextEvent](#) pesan pada aliran input. Setiap acara dapat dikirim segera setelah teks tersedia, tanpa menunggu input lengkap siap. Peristiwa teks tidak perlu diselaraskan dengan batas kalimat atau tanda baca. Amazon Polly menyusun kembali teks secara internal dan menghasilkan ucapan yang terdengar alami terlepas dari bagaimana masukan dibagi di seluruh peristiwa.

Note

Saat menggunakan [SSML](#), setiap dokumen SSML harus mandiri dalam satu dokumen. TextEvent Anda tidak dapat membagi tag SSML di beberapa peristiwa. Namun, Anda dapat mencampur peristiwa teks biasa dan peristiwa SSML dalam aliran yang sama.

Aliran tunduk pada batas waktu berikut:

- Durasi streaming maksimum: 10 menit. Amazon Polly menutup aliran setelah 10 menit terlepas dari aktivitasnya. Jika konten Anda membutuhkan lebih banyak waktu, buka aliran baru untuk teks yang tersisa.
- Batas waktu idle antara peristiwa berturut-turut: 5 detik. Jika tidak ada peristiwa input yang dikirim selama 5 detik, Amazon Polly menutup aliran. Jika sumber teks Anda memiliki jeda lebih dari 5 detik, kirim keep-alive `TextEvent` dengan string kosong atau spasi putih untuk mencegah batas waktu.

Memaksa sintesis teks buffer dengan pembilasan

Secara default, Amazon Polly memutuskan kapan harus mensintesis teks buffer berdasarkan batas bahasa alami. Ini menghasilkan kualitas audio terbaik tetapi berarti audio mungkin tidak dikembalikan segera setelah Anda mengirim `fileTextEvent`.

Pembilasan memberi Anda kendali atas kapan sintesis terjadi. Saat Anda menyiram, Amazon Polly segera mensintesis semua teks yang telah disangga sejauh ini, terlepas dari apakah teks berakhir pada batas alami. Ini berguna ketika sumber teks Anda berhenti di antara bagian logis dan Anda ingin mengirimkan audio untuk apa yang telah dikirim sejauh ini.

Untuk menyiram, atur [FlushStreamConfiguration](#) `Force` parameter ke `true` pada `aTextEvent`. Anda juga dapat mengirim kosong `TextEvent` dengan flag flush disetel untuk memicu sintesis tanpa menambahkan konten baru.

Flushing adalah tradeoff. Pengaturan `Force` ke `true` tengah kalimat dapat mempengaruhi pengucapan dan intonasi karena synthesizer tidak memiliki konteks tentang apa yang berikut. Untuk hasil terbaik, izinkan Amazon Polly untuk menyangga batas alami bila memungkinkan dan hanya memaksa sintesis ketika persyaratan latensi menuntutnya.

Menerima audio

Saat Amazon Polly mensintesis teks, ia mengembalikan [AudioEvent](#) pesan pada aliran output. Setiap acara berisi sepotong data audio. Aplikasi Anda harus mengumpulkan potongan-potongan ini (misalnya, dengan menuliskannya secara berurutan ke file atau buffer audio) untuk menghasilkan output audio yang lengkap. Acara audio dapat tiba saat Anda masih mengirim acara teks.

Tutup aliran

Ketika semua teks input telah dikirim, klien mengirimkan file [CloseStreamEvent](#). Amazon Polly selesai memproses teks buffer yang tersisa, mengirimkan peristiwa audio akhir, dan mengembalikan file [StreamClosedEvent](#) yang berisi jumlah total karakter yang disintesis. Selalu kirim a `CloseStreamEvent` daripada mengandalkan pembilasan untuk mengakhiri aliran. Penutupan memastikan bahwa semua teks buffer disintesis dan dikembalikan.

Untuk detail selengkapnya tentang parameter permintaan, jenis peristiwa, dan kesalahan, lihat [referensi StartSpeechSynthesisStream API](#).

Contoh kode

Contoh berikut menunjukkan sesi streaming dua arah lengkap. Program ini membuat klien Amazon Polly async, membuka aliran yang dikonfigurasi dengan mesin generatif dan output MP3, mengirim teks sebagai serangkaian `TextEvent` pesan, mengakumulasi potongan yang `AudioEvent` dikembalikan ke dalam file output, dan menutup aliran dengan file `CloseStreamEvent`. Karena input dan output terjadi secara bersamaan, data audio mulai berdatangan sebelum semua teks dikirim.

Java

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.polly.PollyAsyncClient;
import software.amazon.awssdk.services.polly.model.*;

import java.io.FileOutputStream;
import java.io.OutputStream;
import java.util.concurrent.CompletableFuture;
import java.util.concurrent.Semaphore;
import org.reactivestreams.Publisher;
import org.reactivestreams.Subscription;

public class BidirectionalStreamExample {

    public static void main(String[] args) throws Exception {
        try (PollyAsyncClient pollyClient = PollyAsyncClient.builder()
            .region(Region.US_EAST_1)
            .build()) {
```

```

        StartSpeechSynthesisStreamRequest request =
StartSpeechSynthesisStreamRequest.builder()
    .engine(Engine.GENERATIVE)
    .voiceId(VoiceId.TIFFANY)
    .outputFormat(OutputFormat.MP3)
    .sampleRate("24000")
    .build();

    try (OutputStream audioOutput = new FileOutputStream("output.mp3")) {

        StartSpeechSynthesisStreamResponseHandler responseHandler =
            StartSpeechSynthesisStreamResponseHandler.builder()

                .subscriber(StartSpeechSynthesisStreamResponseHandler.Visitor.builder()
                    .onAudioEvent(audioEvent -> {
                        try {
                            byte[] audio =
audioEvent.audioChunk().asByteArray();
                            System.out.println("Received AudioEvent: " +
audio.length + " bytes");

                            audioOutput.write(audio);
                        } catch (Exception e) {
                            throw new RuntimeException(e);
                        }
                    })
                    .onStreamClosedEvent(closedEvent -> {
                        System.out.println("Stream closed. Characters
synthesized: "
                                + closedEvent.requestCharacters());
                    })
                    .onDefault(event -> {})
                    .build())
                .onError(error -> {
                    System.err.println("Stream error: " +
error.getMessage());
                })
                .build();

        String[] textChunks = {
            "The weather forecast for today shows clear skies ",
            "with temperatures reaching twenty five degrees. ",
            "Tomorrow we expect some cloud cover in the morning ",
            "but it should clear up by the afternoon. ",
            "The rest of the week looks mostly sunny ",

```

```

        "with a slight chance of rain on Friday. ",
        "Overall a great week to spend time outdoors."
    };

    Publisher<StartSpeechSynthesisStreamActionStream> inputPublisher =
subscriber -> {
    subscriber.onSubscribe(new Subscription() {
        private final Semaphore permits = new Semaphore(0);
        private volatile boolean cancelled = false;

        @Override
        public void request(long n) {
            permits.release((int) Math.min(n, Integer.MAX_VALUE));
        }

        @Override
        public void cancel() {
            cancelled = true;
            permits.release();
        }

        {
            new Thread(() -> {
                for (String chunk : textChunks) {
                    try { permits.acquire(); } catch (InterruptedException
e) { return; }

                    if (cancelled) return;
                    System.out.println("Sending TextEvent: " +
chunk.trim());

                    subscriber.onNext(StartSpeechSynthesisStreamActionStream.textEventBuilder()
                        .text(chunk).textType(TextType.TEXT).build());
                    // Simulate delay between chunks (e.g. waiting for LLM
tokens)
                    try { Thread.sleep(300); } catch (InterruptedException
e) { return; }

                }
                if (!cancelled) {
                    subscriber.onNext(StartSpeechSynthesisStreamActionStream
                        .closeStreamEventBuilder().build());
                    subscriber.onComplete();
                }
            }).start();
        }
    }
}

```

```

        });
    };

    CompletableFuture<Void> future = pollyClient.startSpeechSynthesisStream(
        request, inputPublisher, responseHandler);

    future.join();
} // audioOutput closed
} // pollyClient closed
}
}

```

JavaScript

```

import { PollyClient, StartSpeechSynthesisStreamCommand } from "@aws-sdk/client-polly";
import { createWriteStream } from "fs";

const client = new PollyClient({ region: "us-east-1" });

const textChunks = [
    "The weather forecast for today shows clear skies ",
    "with temperatures reaching twenty five degrees. ",
    "Tomorrow we expect some cloud cover in the morning ",
    "but it should clear up by the afternoon. ",
    "The rest of the week looks mostly sunny ",
    "with a slight chance of rain on Friday. ",
    "Overall a great week to spend time outdoors.",
];

const sleep = (ms) => new Promise((resolve) => setTimeout(resolve, ms));

async function* createInputEvents() {
    for (const chunk of textChunks) {
        console.log(`Sending TextEvent: ${chunk.trim()}`);
        yield {
            TextEvent: {
                Text: chunk,
                TextType: "text",
            },
        };
    }
    // Simulate delay between chunks (e.g. waiting for LLM tokens)
    await sleep(300);
}

```

```

    }

    yield { CloseStreamEvent: {} };
  }

  async function synthesizeStream() {
    const command = new StartSpeechSynthesisStreamCommand({
      Engine: "generative",
      VoiceId: "Tiffany",
      OutputFormat: "mp3",
      SampleRate: "24000",
      ActionStream: createInputEvents(),
    });

    const response = await client.send(command);
    const outputStream = createWriteStream("output.mp3");

    for await (const event of response.EventStream) {
      if (event.AudioEvent) {
        console.log(`Received AudioEvent: ${event.AudioEvent.AudioChunk.length}
bytes`);
        outputStream.write(event.AudioEvent.AudioChunk);
      } else if (event.StreamClosedEvent) {
        console.log(
          `Stream closed. Characters synthesized:
${event.StreamClosedEvent.RequestCharacters}`
        );
      }
    }

    outputStream.end();
  }

  synthesizeStream().catch(console.error);

```

Go

```

package main

import (
    "context"
    "fmt"
    "os"

```

```
"strings"
"time"

"github.com/aws/aws-sdk-go-v2/aws"
"github.com/aws/aws-sdk-go-v2/config"
"github.com/aws/aws-sdk-go-v2/service/polly"
"github.com/aws/aws-sdk-go-v2/service/polly/types"
)

func main() {
    ctx := context.Background()

    cfg, err := config.LoadDefaultConfig(ctx, config.WithRegion("us-east-1"))
    if err != nil {
        fmt.Fprintf(os.Stderr, "Failed to load config: %v\n", err)
        os.Exit(1)
    }

    client := polly.NewFromConfig(cfg)

    output, err := client.StartSpeechSynthesisStream(ctx,
        &polly.StartSpeechSynthesisStreamInput{
            Engine:          types.EngineGenerative,
            VoiceId:        types.VoiceIdTiffany,
            OutputFormat:  types.OutputFormatMp3,
            SampleRate:    aws.String("24000"),
        })
    if err != nil {
        fmt.Fprintf(os.Stderr, "Failed to start stream: %v\n", err)
        os.Exit(1)
    }

    stream := output.GetStream()
    defer stream.Close()

    audioFile, err := os.Create("output.mp3")
    if err != nil {
        fmt.Fprintf(os.Stderr, "Failed to create output file: %v\n", err)
        os.Exit(1)
    }
    defer audioFile.Close()

    textChunks := []string{
        "The weather forecast for today shows clear skies ",
    }
```

```

    "with temperatures reaching twenty five degrees. ",
    "Tomorrow we expect some cloud cover in the morning ",
    "but it should clear up by the afternoon. ",
    "The rest of the week looks mostly sunny ",
    "with a slight chance of rain on Friday. ",
    "Overall a great week to spend time outdoors.",
}

// Send text events in a goroutine
go func() {
    for _, chunk := range textChunks {
        fmt.Printf("Sending TextEvent: %s\n", strings.TrimSpace(chunk))
        err := stream.Send(ctx,
&types.StartSpeechSynthesisStreamActionStreamMemberTextEvent{
            Value: types.TextEvent{
                Text:      aws.String(chunk),
                TextType: types.TextTypeText,
            },
        })
        if err != nil {
            fmt.Fprintf(os.Stderr, "Failed to send text event: %v\n", err)
            return
        }
        // Simulate delay between chunks (e.g. waiting for LLM tokens)
        time.Sleep(300 * time.Millisecond)
    }

    // Signal end of input
    stream.Send(ctx,
&types.StartSpeechSynthesisStreamActionStreamMemberCloseStreamEvent{
        Value: types.CloseStreamEvent{},
    })
}()

// Receive audio events
for event := range stream.Events() {
    switch v := event.(type) {
    case *types.StartSpeechSynthesisStreamEventStreamMemberAudioEvent:
        fmt.Printf("Received AudioEvent: %d bytes\n", len(v.Value.AudioChunk))
        audioFile.Write(v.Value.AudioChunk)
    case *types.StartSpeechSynthesisStreamEventStreamMemberStreamClosedEvent:
        fmt.Printf("Stream closed. Characters synthesized: %d\n",
v.Value.RequestCharacters)
    }
}

```

```

}

if err := stream.Err(); err != nil {
    fmt.Fprintf(os.Stderr, "Stream error: %v\n", err)
    os.Exit(1)
}
}

```

Rust

```

use aws_sdk_polly::types::{
    CloseStreamEvent, Engine, OutputFormat,
    StartSpeechSynthesisStreamActionStream,
    StartSpeechSynthesisStreamEventStream, TextEvent, TextType, VoiceId,
};
use aws_sdk_polly::Client;
use std::fs::File;
use std::io::Write;
use tokio::time::{sleep, Duration};

#[tokio::main]
async fn main() {
    let config = aws_config::defaults(aws_config::BehaviorVersion::latest())
        .region("us-east-1")
        .load()
        .await;

    let client = Client::new(&config);

    let text_chunks = vec![
        "The weather forecast for today shows clear skies ",
        "with temperatures reaching twenty five degrees. ",
        "Tomorrow we expect some cloud cover in the morning ",
        "but it should clear up by the afternoon. ",
        "The rest of the week looks mostly sunny ",
        "with a slight chance of rain on Friday. ",
        "Overall a great week to spend time outdoors.",
    ];

    let input_stream = async_stream::stream! {
        for chunk in &text_chunks {
            println!("Sending TextEvent: {}", chunk.trim());
            yield Ok(StartSpeechSynthesisStreamActionStream::TextEvent(

```

```

TextEvent::builder().text(*chunk).text_type(TextType::Text).build().unwrap(),
    ));
    // Simulate delay between chunks (e.g. waiting for LLM tokens)
    sleep(Duration::from_millis(300)).await;
}
yield Ok(StartSpeechSynthesisStreamActionStream::CloseStreamEvent(
    CloseStreamEvent::builder().build(),
));
};

let mut output = client
    .start_speech_synthesis_stream()
    .engine(Engine::Generative)
    .voice_id(VoiceId::Tiffany)
    .output_format(OutputFormat::Mp3)
    .sample_rate("24000")
    .action_stream(input_stream.into())
    .send()
    .await
    .expect("Failed to start stream");

let mut audio_file = File::create("output.mp3").expect("Failed to create output
file");

while let Ok(Some(event)) = output.event_stream.recv().await {
    match event {
        StartSpeechSynthesisStreamEventStream::AudioEvent(audio_event) => {
            if let Some(chunk) = audio_event.audio_chunk() {
                let bytes = chunk.as_ref();
                println!("Received AudioEvent: {} bytes", bytes.len());
                audio_file.write_all(bytes).unwrap();
            }
        }
        StartSpeechSynthesisStreamEventStream::StreamClosedEvent(closed_event)
=> {
            println!(
                "Stream closed. Characters synthesized: {}",
                closed_event.request_characters()
            );
        }
        _ => {}
    }
}
}

```

```
}
```

Suara di Amazon Polly

Amazon Polly menyediakan lusinan suara yang hidup dan dukungan untuk berbagai bahasa. Setiap suara dibuat menggunakan penutur bahasa asli, sehingga ada variasi dari suara ke suara, bahkan dalam bahasa yang sama. Anda juga dapat menggunakan AWS Management Console untuk menguji setiap suara dengan teks pilihan Anda. Untuk sebagian besar bahasa, setidaknya akan ada satu suara pria dan satu suara wanita, dan seringkali lebih dari satu suara. Beberapa bahasa hanya memiliki satu suara.

Inventarisasi suara dan jumlah bahasa yang disertakan terus diperbarui untuk menyertakan pilihan tambahan. Untuk menyarankan bahasa atau suara baru, berikan umpan balik di halaman ini. Sayangnya, kami tidak dapat mengomentari rencana untuk bahasa baru tertentu sebelum dirilis.

Note

Untuk mendengar contoh suara Amazon Polly di browser Anda, lihat ikhtisar produk [Amazon Polly](#).

Topik

- [Suara yang tersedia](#)
- [Suara bilingual](#)
- [Menerapkan suara penyiar berita](#)
- [Mendengarkan suara](#)
- [Pengaturan waktu kecepatan suara](#)
- [Mengubah kecepatan suara](#)

Suara yang tersedia

Amazon Polly menyediakan berbagai suara yang hidup dalam berbagai bahasa untuk mensintesis ucapan dari teks. Tabel berikut menunjukkan semua suara yang ditawarkan Amazon Polly.

	Varian bahasa dan bahasa	Kode bahasa	Nama/ ID	Gender	Suara generatif	Suara Formulir Panjang	Suara saraf	Suara standar
1	Bahasa Arab	arb	Zeina	Perempuan	Tidak	Tidak	Tidak	Ya
2	Arab (Teluk)	AR-ae	Hala*	Perempuan	Tidak	Tidak	Ya	Tidak
			Zayd*	Laki-laki	Tidak	Tidak	Ya	Tidak
3	Belanda (Belgia)	NI-be	Lisa	Perempuan	Ya	Tidak	Ya	Tidak
4	Catalan	CA-es	Arlet	Perempuan	Tidak	Tidak	Ya	Tidak
5	Ceko	cs-cz	Jitka	Perempuan	Tidak	Tidak	Ya	Tidak
6	Tionghoa (Kanton)	Yue-CN	Hiujin	Perempuan	Tidak	Tidak	Ya	Tidak
7	Mandarin (Mandarin)	Cmn-cn	Zhiyu	Perempuan	Tidak	Tidak	Ya	Ya
8	Denmark	Da-dk	Naja	Perempuan	Tidak	Tidak	Tidak	Ya
			Ibu	Laki-laki	Tidak	Tidak	Tidak	Ya
			Sofie	Perempuan	Tidak	Tidak	Ya	Tidak
9	Belanda	NI-NL	Laura	Perempuan	Ya	Tidak	Ya	Tidak
			Lotte	Perempuan	Tidak	Tidak	Tidak	Ya
			Ruben	Laki-laki	Tidak	Tidak	Tidak	Ya
10	Inggris (Australia)	En-au	Nicole	Perempuan	Tidak	Tidak	Tidak	Ya
			Olivia	Perempuan	Ya	Tidak	Ya	Tidak

	Varian bahasa dan bahasa	Kode bahasa	Nama/ ID	Gender	Suara generatif	Suara Formulir Panjang	Suara saraf	Suara standar
			Russell	Laki-laki	Tidak	Tidak	Tidak	Ya
11	Inggris (Inggris)	id - GB	Amy**	Perempuan	Ya	Tidak	Ya	Ya
			Emma	Perempuan	Tidak	Tidak	Ya	Ya
			Brian	Laki-laki	Ya	Tidak	Ya	Ya
			Arthur	Laki-laki	Tidak	Tidak	Ya	Tidak
12	Inggris (India)	En-in	Aditi*	Perempuan	Tidak	Tidak	Tidak	Ya
			Raveena	Perempuan	Tidak	Tidak	Tidak	Ya
			Kajal*	Perempuan	Ya	Tidak	Ya	Tidak
13	Inggris (Irlandia)	En-ie	Niamh	Perempuan	Ya	Tidak	Ya	Tidak
14	Inggris (Selandia Baru)	en-NZ	Aria	Perempuan	Ya	Tidak	Ya	Tidak
15	English (Singaporean)	En-SG	Melati	Perempuan	Ya	Tidak	Ya	Tidak
16	Inggris (Afrika Selatan)	En-za	Ayanda	Perempuan	Ya	Tidak	Ya	Tidak

	Varian bahasa dan bahasa	Kode bahasa	Nama/ ID	Gender	Suara generatif	Suara Formulir Panjang	Suara saraf	Suara standar
17	Inggris (US)	en-US	Danielle	Perempuan	Ya	Ya	Ya	Tidak
			Gregory	Laki-laki	Tidak	Ya	Ya	Tidak
			Ivy	Perempuan (anak)	Tidak	Tidak	Ya	Ya
			Joanna**		Ya	Tidak	Ya	Ya
			Kendra	Perempuan	Tidak	Tidak	Ya	Ya
			Kimberly	Perempuan	Tidak	Tidak	Ya	Ya
			Salli	Perempuan	Ya	Tidak	Ya	Ya
			Joey	Perempuan	Tidak	Tidak	Ya	Ya
			Justin	Laki-laki	Tidak	Tidak	Ya	Tidak
			Kevin	Laki-laki (anak)	Tidak	Tidak	Ya	Ya
			Matthew**	Laki-laki (anak)	Ya	Tidak	Ya	Tidak
			Ruth		Ya	Ya	Ya	Tidak
			Stephen	Laki-laki	Ya	Tidak	Ya	Tidak
			Tiffany	Perempuan	Ya	Tidak	Tidak	Tidak
			Patrick	Laki-laki	Tidak	Ya	Tidak	Tidak
				Perempuan				
				Laki-laki				
18	Inggris (Welsh)	En-GB-WLS	Geraint	Laki-laki	Tidak	Tidak	Tidak	Ya

	Varian bahasa dan bahasa	Kode bahasa	Nama/ ID	Gender	Suara generatif	Suara Formulir Panjang	Suara saraf	Suara standar
19	Finlandia	Fi-Fi	Suvi	Perempuan	Tidak	Tidak	Ya	Tidak
20	Perancis	FR-fr	Ambre	Perempuan	Ya	Tidak	Tidak	Tidak
			Seline/ Celine	Perempuan	Ya	Tidak	Tidak	Ya
			Laki-laki	Ya	Tidak	Tidak	Tidak	Tidak
			Florian	Perempuan	Ya	Tidak	Ya	Ya
			Léa	Laki-laki	Tidak	Tidak	Tidak	Ya
			Mathieu	Laki-laki	Ya	Tidak	Ya	Tidak
			Remi					
21	Perancis (Belgia)	FR-Be	Isabelle	Perempuan	Ya	Tidak	Ya	Tidak
22	Perancis (Kanada)	FR-Ca	Chantal	Perempuan	Tidak	Tidak	Tidak	Ya
			Gabrielle	Perempuan	Ya	Tidak	Ya	Tidak
			Liam	Laki-laki	Ya	Tidak	Ya	Tidak
23	Jerman	De-de	Marlene	Perempuan	Tidak	Tidak	Tidak	Ya
			Vicki	Perempuan	Ya	Tidak	Ya	Ya
			Hans	Laki-laki	Tidak	Tidak	Tidak	Ya
			Daniel	Laki-laki	Ya	Tidak	Ya	Tidak
			Lennart	Laki-laki	Ya	Tidak	Tidak	Tidak
24	Jerman (Austria)	De-at	Hannah	Perempuan	Ya	Tidak	Ya	Tidak

	Varian bahasa dan bahasa	Kode bahasa	Nama/ ID	Gender	Suara generatif	Suara Formulir Panjang	Suara saraf	Suara standar
25	Jerman (Swiss)	De-ch	Sabrina	Perempuan	Ya	Tidak	Ya	Tidak
26	Hindi	Hi-in	Aditi*	Perempuan	Tidak	Tidak	Tidak	Ya
			Kajal*	Perempuan	Tidak	Tidak	Ya	Tidak
27	Islandia	As-adalah	Dora/ Dora	Perempuan	Tidak	Tidak	Tidak	Ya
			Karl	Laki-laki	Tidak	Tidak	Tidak	Ya
28	Italia	It-itu	Beatrice	Perempuan	Ya	Tidak	Tidak	Tidak
			Carla	Perempuan	Tidak	Tidak	Tidak	Ya
			Bianca	Perempuan	Ya	Tidak	Ya	Ya
			Lorenzo	Laki-laki	Ya	Tidak	Tidak	Tidak
			Giorgio	Laki-laki	Tidak	Tidak	Tidak	Ya
			Adriano	Laki-laki	Tidak	Tidak	Ya	Tidak
29	Jepang	Ja-JP	Mizuki	Perempuan	Tidak	Tidak	Tidak	Ya
			Takumi	Laki-laki	Tidak	Tidak	Ya	Ya
			Kazuha	Perempuan	Tidak	Tidak	Ya	Tidak
			Tomoko	Perempuan	Tidak	Tidak	Ya	Tidak
30	Korea	Ko-kr	Seoyeon	Perempuan	Ya	Tidak	Ya	Ya
			Jihye	Perempuan	Tidak	Tidak	Ya	Tidak

	Varian bahasa dan bahasa	Kode bahasa	Nama/ ID	Gender	Suara generatif	Suara Formulir Panjang	Suara saraf	Suara standar
31	Norwegia	NB-Tidak	Liv	Perempuan	Tidak	Tidak	Tidak	Ya
			Ida	Perempuan	Tidak	Tidak	Ya	Tidak
32	Polandia	PL-PL	Ewa	Perempuan	Ya	Tidak	Tidak	Ya
			Maja	Perempuan	Tidak	Tidak	Tidak	Ya
			Jacek	Laki-laki	Tidak	Tidak	Tidak	Ya
			Jan	Laki-laki	Tidak	Tidak	Tidak	Ya
			Ola	Perempuan	Ya	Tidak	Ya	Tidak
33	Portugis (Brasil)	Pt-BR	Camila	Perempuan	Ya	Tidak	Ya	Ya
			Vitoria/ Vitoria	Perempuan	Tidak	Tidak	Ya	Ya
				Laki-laki	Tidak	Tidak	Tidak	Ya
			Ricardo	Laki-laki	Tidak	Tidak	Ya	Tidak
			Thiago					
34	Portugis (Eropa)	Pt-PT	Inês/ Ines	Perempuan	Tidak	Tidak	Ya	Ya
			Cristiano	Laki-laki	Tidak	Tidak	Tidak	Ya
35	Rumania	Ro-ro	Carmen	Perempuan	Tidak	Tidak	Tidak	Ya
36	Rusia	Ru-ru	Tatiana	Perempuan	Tidak	Tidak	Tidak	Ya
			Maxim	Laki-laki	Tidak	Tidak	Tidak	Ya

	Varian bahasa dan bahasa	Kode bahasa	Nama/ ID	Gender	Suara generatif	Suara Formulir Panjang	Suara saraf	Suara standar
37	Spanyol (Spanyol)	ES-es	Conchita	Perempuan	Tidak	Tidak	Tidak	Ya
			Lucia	Perempuan	Ya	Tidak	Ya	Ya
			Alba	Perempuan	Tidak	Ya	Tidak	Tidak
			Enrique	Laki-laki	Tidak	Tidak	Tidak	Ya
			Sergio	Laki-laki	Ya	Tidak	Ya	Tidak
			Raul	Laki-laki	Tidak	Ya	Tidak	Tidak
38	Spanyol (Meksiko)	ES-mx	Mia	Perempuan	Ya	Tidak	Ya	Ya
			Andrés	Laki-laki	Ya	Tidak	Ya	Tidak
39	Spanyol (AS)	es-AS	Lupe**	Perempuan	Ya	Tidak	Ya	Ya
			Penélope/ Penelope	Perempuan	Tidak	Tidak	Tidak	Ya
				Laki-laki	Tidak	Tidak	Tidak	Ya
			Miguel	Laki-laki	Ya	Tidak	Ya	Tidak
40	Swedia	SV-SE	Astrid	Perempuan	Tidak	Tidak	Tidak	Ya
			Elin	Perempuan	Tidak	Tidak	Ya	Tidak
41	Turki	TR-TR	Filiz	Perempuan	Tidak	Tidak	Tidak	Ya
			Burcu	Perempuan	Tidak	Tidak	Ya	Tidak
42	Welsh	Cy-GB	Gwyneth	Perempuan	Tidak	Tidak	Tidak	Ya

* Suara ini bilingual. Untuk informasi selengkapnya, lihat [Suara bilingual](#).

** Suara-suara ini dapat digunakan dengan gaya berbicara Newscaster saat digunakan dengan format Neural. Untuk informasi selengkapnya, lihat [Menerapkan suara penyiar berita](#).

Setiap mesin suara Amazon Polly memiliki fitur unik. Pelajari lebih lanjut tentang fitur dan ketersediaan Wilayah untuk mesin suara yang ditawarkan oleh Amazon Polly:

- [Suara generatif](#)
- [Suara bentuk panjang](#)
- [Suara saraf](#)
- [Suara standar](#)

Suara merek

Selain suara yang tersedia yang tercantum dalam tabel sebelumnya, Anda dapat menggunakan Amazon Polly untuk membangun suara khusus untuk persona merek Anda. Dengan suara merek, Anda dapat menawarkan suara unik dan eksklusif kepada pelanggan Anda. [Untuk mempelajari lebih lanjut tentang suara merek Amazon Polly, lihat Suara Merek.](#)

Suara bilingual

Amazon Polly memiliki dua cara untuk menghasilkan suara bilingual:

- [Suara dwibahasa beraksen](#)
- [Suara dwibahasa sepenuhnya](#)

Suara dwibahasa beraksen

Suara bilingual beraksen dapat dibuat menggunakan suara Amazon Polly apa pun, tetapi hanya saat menggunakan tag SSML.

Biasanya, semua kata dalam teks input diucapkan dalam bahasa default suara yang ditentukan yang Anda gunakan.

Misalnya, jika Anda menggunakan suara Joanna (yang berbicara bahasa Inggris AS), Amazon Polly berbicara berikut dengan suara Joanna tanpa aksen Prancis:

```
<speak>
```

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

Dalam hal ini, kata-kata *Je ne parle pas français* diucapkan sebagaimana adanya jika mereka adalah bahasa Inggris.

Namun, jika Anda menggunakan suara Joanna dengan `<lang>`tag, Amazon Polly mengucapkan kalimat dalam suara Joanna dalam bahasa Prancis beraksen Amerika:

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

Karena Joanna bukan suara asli Prancis, pengucapan didasarkan pada bahasa ibunya, Inggris AS. Misalnya, meskipun pengucapan bahasa Prancis yang sempurna menampilkan getaran uvular /R/ dalam kata *français*, suara Inggris AS Joanna mengucapkan fonem ini sebagai suara yang sesuai /r/.

Jika Anda menggunakan suara Giorgio, yang berbicara bahasa Italia, dengan teks berikut, Amazon Polly mengucapkan kalimat dalam suara Giorgio dengan pengucapan Italia:

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

Suara dwibahasa sepenuhnya

Suara dwibahasa sepenuhnya seperti Aditi atau Kajal (bahasa Inggris India dan Hindi) dapat berbicara dua bahasa dengan lancar. Ini memberi Anda kemampuan untuk menggunakan kata dan frasa dari kedua bahasa dalam satu teks menggunakan suara yang sama.

Saat ini, Aditi, Kajal, Hala, dan Zayd adalah satu-satunya suara dwibahasa yang tersedia.

Menggunakan Suara Bilingual (contoh: Aditi)

Aditi berbicara bahasa Inggris India (En-in) dan Hindi (Hi-in) dengan lancar. Anda dapat mensintesis ucapan dalam bahasa Inggris dan Hindi, dan suara dapat beralih di antara dua bahasa bahkan dalam kalimat yang sama.

Hindi dapat digunakan dalam dua bentuk yang berbeda:

- Devanagari: “”
- Romanagari (menggunakan alfabet Latin): “Usne kahan, khel toh ab shuru hoga”

Selain itu, dimungkinkan untuk mencampur bahasa Inggris dan Hindi dari salah satu atau kedua bentuk dalam satu kalimat:

- Devanagari + English: “Ini lagunya”
- Romanagari + Inggris: “Ini adalah lagu dari film Jaane Tu Ya Jaane Na.”
- Devanagari + Romanagari+Bahasa Inggris: “Ini adalah lagu dari film Jaane Tu Ya Jaane Na.”

Karena Aditi adalah suara bilingual, teks dalam semua kasus ini akan dibaca dengan benar, karena Amazon Polly dapat membedakan antara bahasa dan skrip.

Amazon Polly juga mendukung angka, tanggal, waktu, dan ekspansi mata uang dalam bahasa Inggris (angka Arab) dan Hindi (angka Devanagari). Secara default, angka Arab dibaca dalam bahasa Inggris India. Untuk membuat Amazon Polly membacanya dalam bahasa Hindi, Anda harus menggunakan parameter kode `hi-IN` bahasa.

Menerapkan suara penyiar berita

Orang menggunakan gaya berbicara yang berbeda, tergantung pada konteksnya. Percakapan santai, misalnya, terdengar sangat berbeda dari siaran berita TV atau radio. Karena cara suara standar dibuat, mereka tidak dapat menghasilkan gaya berbicara yang berbeda. Namun, suara saraf bisa. Mereka dapat dilatih untuk gaya berbicara tertentu, dengan variasi dan penekanan pada bagian-bagian tertentu dari ucapan yang melekat dalam gaya itu.

Selain suara saraf default, Amazon Polly menyediakan gaya berbicara penyiar berita yang menggunakan sistem saraf untuk menghasilkan ucapan dalam gaya penyiar TV atau radio. Gaya penyiar berita tersedia dengan suara Matthew dan Joanna dalam bahasa Inggris AS (en-US), suara Lupe dalam bahasa Spanyol AS (es-US), dan suara Amy dalam bahasa Inggris Inggris (en-GB).

Untuk menggunakan gaya Newscaster, pertama-tama pilih mesin saraf dan kemudian gunakan sintaks yang dijelaskan dalam langkah-langkah berikut dalam teks input Anda.

 Note

- Untuk menggunakan gaya berbicara saraf apa pun, Anda harus menggunakan salah satu AWS Wilayah yang mendukung suara saraf. Opsi ini tidak tersedia di semua Wilayah. Untuk informasi selengkapnya, lihat [Kompatibilitas fitur dan wilayah](#).

Console

Untuk menerapkan gaya Newscaster

1. Buka konsol Amazon Polly di <https://console.aws.amazon.com/polly/>
2. Pastikan Anda menggunakan AWS Wilayah di mana suara saraf didukung.
3. Pada Text-to-Speech halaman, untuk Engine, pilih Neural.
4. Pilih bahasa dan suara yang ingin Anda gunakan. Hanya Matthew dan Joanna untuk bahasa Inggris AS (en-US), Lupe untuk bahasa Spanyol AS (es-AS), dan Amy untuk Inggris Inggris (en-GB) yang tersedia dalam suara penyiar berita.
5. Nyalakan SSML.
6. Tambahkan teks masukan ke text-to-speech permintaan Anda menggunakan sintaks SSML gaya Newscaster.

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

Misalnya, Anda dapat menggunakan tag penyiar berita sebagai berikut:

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

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

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

```
</amazon:domain>  
</speak>
```

7. Pilih Dengarkan.

AWS CLI

Untuk menerapkan gaya Newscaster

1. Dalam permintaan API Anda, sertakan parameter engine dengan `neural` nilai:

```
--engine neural
```

2. Tambahkan teks masukan ke permintaan API Anda menggunakan sintaks SSML gaya Newscaster.

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

Misalnya, Anda dapat menggunakan tag penyiar berita sebagai berikut:

```
<speak>  
<amazon:domain name="news">  
From the Tuesday, April 16th, 1912 edition of The Guardian newspaper:  
  
The maiden voyage of the White Star liner Titanic, the largest ship ever  
launched  
ended in disaster.  
  
The Titanic started her trip from Southampton for New York on Wednesday. Late  
on  
Sunday night she struck an iceberg off the Grand Banks of Newfoundland. By  
wireless telegraphy she sent out signals of distress, and several liners were  
near enough to catch and respond to the call.  
</amazon:domain>  
</speak>
```

Untuk informasi selengkapnya tentang SSML, lihat [Tag SSML yang didukung](#)

Mendengarkan suara

Setelah [menyiapkan](#) Amazon Polly, Anda dapat menguji suara menggunakan teks khusus di konsol.

Untuk mendengarkan suara Amazon Polly di konsol

1. Masuk ke AWS Management Console dan buka konsol Amazon Polly di. <https://console.aws.amazon.com/polly/>
2. Pilih tab Text-to-Speech.
3. Untuk Engine, pilih Generative, Long Form, Neural, atau Standard.
4. Pilih bahasa dan Wilayah. Kemudian pilih suara.
5. Masukkan teks untuk suara yang akan diucapkan atau gunakan frasa default, lalu pilih Dengarkan.

Pengaturan waktu kecepatan suara

Karena variasi alami antara suara, setiap suara yang tersedia berbicara dengan kecepatan yang sedikit berbeda. Misalnya, dengan suara Inggris AS, Ivy dan Joanna sedikit lebih cepat daripada Matthew, dan jauh lebih cepat daripada Joey. Karena ada begitu banyak variasi antara suara, tidak ada kecepatan standar (kata per menit) yang tersedia untuk suara Amazon Polly. Namun, Anda dapat menemukan berapa lama waktu yang dibutuhkan suara Anda untuk mengucapkan teks yang dipilih menggunakan [Tanda Pidato](#).

Untuk mengatur waktu panjang bagian teks yang diucapkan

1. Buka AWS CLI.
2. Jalankan kode berikut, isi sesuai kebutuhan.

```
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. Buka LengthOfText.txt.

Jika teksnya adalah “Mary have a little lamb,” beberapa baris terakhir yang dikembalikan oleh Amazon Polly adalah:

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

Viseme terakhir, pada dasarnya suara untuk huruf terakhir dalam “domba” dimulai 1082 milidetik setelah awal pidato. Meskipun ini tidak persis panjang audio, itu dekat dan dapat berfungsi sebagai dasar untuk perbandingan antara suara.

Mengubah kecepatan suara

Untuk aplikasi tertentu, Anda mungkin menemukan bahwa Anda lebih suka suara yang Anda sukai diperlambat, atau dipercepat. Jika kecepatan suara menjadi perhatian, Amazon Polly menyediakan kemampuan untuk memodifikasi ini menggunakan tag SSML. Misalnya, jika organisasi Anda membuat aplikasi yang membaca buku untuk audiens imigran, Anda mungkin ingin memvariasikan kecepatan suara. Audiens Anda mungkin berbicara bahasa Inggris, tetapi kefasihan mereka terbatas. <prosody>Amazon Polly membantu Anda memperlambat laju bicara menggunakan tag SSML.

Anda dapat menggunakan persentase:

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

Atau kecepatan preset:

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

Dua opsi kecepatan tersedia untuk Anda saat menggunakan SSML dengan Amazon Polly:

- Kecepatan preset: `x-slow`, `slow`, `mediumfast`, dan `x-fast`. Dalam kasus ini, kecepatan setiap opsi adalah perkiraan, tergantung pada suara pilihan Anda. `medium` Pilihannya adalah kecepatan normal suara.

- n% dari tingkat bicara: setiap persentase dari tingkat bicara, antara 20% dan 200% dapat digunakan. Dalam kasus ini, Anda dapat memilih kecepatan yang Anda inginkan. Namun, kecepatan suara yang sebenarnya adalah perkiraan, tergantung pada suara yang Anda pilih. 100% dianggap sebagai kecepatan normal suara.

 Note

Uji suara yang Anda pilih dengan berbagai kecepatan. Kecepatan setiap opsi adalah perkiraan dan tergantung pada suara yang Anda pilih.

Untuk informasi selengkapnya tentang penggunaan prosody tag, lihat [Mengontrol volume, kecepatan berbicara, dan nada](#).

Bahasa di Amazon Polly

Bahasa-bahasa berikut didukung oleh Amazon Polly dan dapat digunakan untuk mensintesis ucapan. Setiap bahasa memiliki kode bahasa yang unik. Kode bahasa ini adalah [tag identifikasi bahasa W3C](#) (*ISO 639-3* untuk nama bahasa dan *ISO 3166* untuk kode negara).

Pilih bahasa dari tabel berikut untuk detail tentang fonem dan viseme yang disediakan Amazon Polly.

Bahasa	Kode bahasa
Bahasa Arab	arb
Arab (Teluk)	AR-ae
Catalan	CA-es
Tionghoa (Kanton)	Yue-CN
Mandarin (Mandarin)	Cmn-cn
Ceko	cs-cz
Denmark	Da-dk
Belanda (Belgia)	Nl-be
Belanda	Nl-NL

Bahasa	Kode bahasa	
Inggris (Australia)	En-au	
Inggris (Inggris)	En-GB	
Inggris (India)	En-in	
Inggris (Selandia Baru)	en-NZ	
English (Singaporean)	En-SG	
Inggris (Afrika Selatan)	En-za	
Inggris (US)	en-US	
Inggris (Welsh)	En-GB-WLS	
Finlandia	Fi-Fi	
Perancis	FR-fr	
Perancis (Belgia)	FR-Be	

Bahasa	Kode bahasa	
Perancis (Kanada)	FR-Ca	
Hindi	Hi-in	
Jerman	De-de	
Jerman (Austria)	De-at	
Jerman (standar Swiss)	De-ch	
Islandia	As-adalah	
Italia	It-itu	
Jepang	Ja-JP	
Korea	Ko-kr	
Norwegia	NB-Tidak	
Polandia	PL-PL	
Portugis (Brasil)	Pt-BR	
Portugis (Eropa)	Pt-PT	

Bahasa	Kode bahasa	
Rumania	Ro-ro	
Rusia	Ru-ru	
Spanyol (Spanyol)	ES-es	
Spanyol (Meksiko)	ES-mx	
Spanyol (AS)	es-AS	
Swedia	SV-SE	
Turki	TR-TR	
Welsh	Cy-GB	

Arab (arb)

Tabel berikut mencantumkan fonem Alfabet Fonetik Internasional (IPA), simbol Alfabet Fonetik Metode Penilaian Pidato Diperpanjang (X-SAMPA), dan viseme yang sesuai untuk suara Arab Zeina yang didukung oleh Amazon Polly.

Tabel Fonem/Viseme

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Konsonan				

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
4	ʔ	berhenti glotal	إلا	
ʕ	ʔ\	frikatif faring bersuara	ممر	k
b	b	plosif bilabial bersuara	الد	p
d	d	plosif alveolar bersuara	ماررالالارر	t
d	d_ʔ \	plosif alveolar bersuara tegas	و	t
ɗ	dʒ	africate postalveo lar bersuara	ممالل	D
ð	D	frikatif gigi bersuara	ج	T
ð	D_ʔ \	frikatif gigi bersuara tegas	لام	T
f	f	frikatif labiodental tanpa suara	الل	f
g	g	plosif velar bersuara	اللررا	k
ɣ	G	frikatif velar bersuara	رر	k
-h	-h	frikatif glottal tanpa suara	ما	k
j	j	aproksiman palatal	امالالالالار	saya

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
k	k	plosif velar tanpa suara	لم	k
l	l	aproksiman lateral alveolar	لار	t
l	L_g	aproksiman lateral alveolar empatik	سرالللرالرالرالر	t
m	m	hidung bilabial	ملا	p
n	n	hidung alveolar	سور	t
p	p	plosif bilabial tak bersuara	مالس	p
q	q	plosif uvular tak bersuara	مراررر	k
r	r	getar alveolar	رمل	r
detik	detik	frikatif alveolar tak bersuara	سال	detik
s	s_? \	frikatif alveolar tanpa suara empatik	16)	detik
ʃ	D	frikatif postalveolar tanpa suara	رر	D
t	t	plosif alveolar tak bersuara	ممر	t
t	t_? \	plosif alveolar tanpa suara yang tegas	16)	t

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
θ	T	frikatif gigi tanpa suara	١٢	T
v	v	frikatif labiodental bersuara	رالف	f
w	w	pendekatan labio-velar	ولد	u
x	x	frikatif velar tanpa suara	و	k
ħ	X\	frikatif faring tak bersuara	مول	k
z	z	frikatif alveolar bersuara	رور	detik
Vokal				
a	a	buka vokal depan yang tidak dibulatkan	ارد	a
a:	a:	vokal depan terbuka panjang yang tidak dibulatkan	ار	a
ɑ	A_? \	vokal terbuka kembali yang tidak dibulatkan	الل	a
ɑ:	A_? \:	vokal panjang terbuka belakang yang tidak dibulatkan	الم	a

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
u	u	tutup kembali vokal bulat	رر	u
u:	u:	vokal bulat panjang dekat	سور	u
u	u_? \	vokal bulat tutup belakang yang tegas	سال	u
u:	u_? \:	vokal bulat panjang dekat belakang yang tegas	ول	u
saya	saya	tutup vokal depan yang tidak dibulatkan	الالالالال	saya
i:	i:	vokal panjang dekat depan yang tidak dibulatkan	الالالالال	saya
saya	saya_? \	vokal depan yang tidak dibulatkan	ضدّ	saya
saya:	saya_? \:	vokal panjang dekat depan yang tidak dibulatkan	ما	saya
e	e	vokal dekat-tengah depan yang tidak dibulatkan	مارررت	e
e:	e:	vokal panjang dekat-tengah depan yang tidak dibulatkan	مورارل	e

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ɔ	O	vokal bulat punggung tengah terbuka	الرالورالالالالال	O
ɔ:	O:	vokal bulat punggung tengah terbuka panjang	سلالالارا لالالالالال	O

Arab (Teluk) (Ar-ae)

Tabel berikut mencantumkan fonem Alfabet Fonetik Internasional (IPA), simbol Alfabet Fonetik Metode Penilaian Pidato Diperpanjang (X-SAMPA), dan viseme yang sesuai untuk suara Arab Hala yang didukung oleh Amazon Polly.

Tabel Fonem/Viseme

IPA	X-SAMPA	Deskripsi	Contoh	Pengucapan	Viseme
Konsonan					
b	b	plosif bilabial bersuara	سلد	/"b a. l a d/	b
d	d	plosif alveolar bersuara	رد	/"r d d/	d
d	d_? \	plosif alveolar bersuara faring	و	/"d_? \ w? /	D
f	f	frikatif labiodental tanpa suara	مrrrrrrال	/"f l. r l n/	f
g	g	plosif velar bersuara	سال	/"g a: l/	k

IPA	X-SAMPA	Deskripsi	Contoh	Pengucapan	Viseme
j	j	pendekata n palatal bersuara	امالالالال	/'j Saya m. S i:/	saya
k	k	plosif velar tanpa suara	سامل	/'k a:. m i l/	k
l	l	pendekatan lateral alveolar bersuara	لال	/'l e: l/	t
l	l_G	aproksiman lateral alveolar bersuara faring	سرااللالر ررالرالر	/? \ a b. " d A_? \ L_g. L_g A_? V	t
m	m	penghentian hidung bilabial	مrrrrرالر	/'m Saya j. j a/	p
n	n	penghentian hidung alveolar	سور	/'n u: r/	t
p	p	plosif bilabial tak bersuara	وررا	/'? O. p e. r a:/	p
q	q	plosif uvular tak bersuara	ررر	/'q A_? \ s_? \ r/	k
r	r	getar alveolar	رمل	/'r a. m l l/	r
detik	detik	frikatif alveolar tak bersuara	سمسم	/'s Saya m/	detik
s	s_? \	frikatif alveolar tak bersuara faring	الرر	/'s_? \ A_? :. X \ l b/	detik

IPA	X-SAMPA	Deskripsi	Contoh	Pengucapan	Viseme
t	t	plosif alveolar tak bersuara	ا م ر	/ˈt a. m a r/	t
t	t_? \	frikatif alveolar tak bersuara faring	ا ل ر	/ˈt_? \ A_? :. l l b/	t
v	v	frikatif labiodental bersuara	ر ا ل ا م ا ر ا ل ر ا ل ا ل ا ل	/v i:. t A. “ m i: n/	f
w	w	pendekata n labiovelar bersuara	و ا ل ل د	/ˈw a:. j l d/	u
x	x	frikatif velar tanpa suara	ر و	/x a. “ r u: f/	k
z	z	frikatif velar tanpa suara	ز و ر	/ˈz h u: r/	detik
ð	D	frikatif interdental bersuara	ل ك	/ˈD a:. l l k/	D
ð	D_? \	frikatif interdental bersuara faring	ل ا م	/D_? \ A_? \ . “ l a: m/	D
ħ	X\	frikatif faring tak bersuara	ا ل ا ل ا ل ا ل ا ل ر	/? l. “ X\ i: n/	k
ŋ	T	berhenti hidung velar	ر و ر ر و ر	/h O N. “ K O N g/	k
ɣ	G	frikatif velar bersuara	ر ا ر ر ر ر ر ر ر ر	/G l. “ r i:. b a/	k

IPA	X-SAMPA	Deskripsi	Contoh	Pengucapan	Viseme
ʃ	D	frikatif postalveolar tanpa suara	مس	/ˈS a m s/	D
ʒ	Z	frikatif postalveolar bersuara	م الارا رالت	/Z a. ˈ k e: t/	D
ʔ	ʔ	berhenti glotal	مس س س ر ر ر لال	/m u. ˈ ʔ a s. s a/	
ʕ	ʔ\	frikatif faring bersuara	سام	/ˈʔ \ a: m m/	k
dʒ	dZ	africate postalveolar bersuara	الم ر ر ر ر ر ر لال	/ˈdZ a: m.ʔ \ a/	D
θ	T	frikatif interdental bersuara	لار	/T a. ˈ l a: . T a/	T
ħ	-h	frikatif glottal bersuara	س ل ل ل	/ˈh l a: l/	k
Vokal					
æ	a	vokal pendek depan terbuka tengah yang tidak dibulatkan	س ر ر	/ˈs a. f a r/	a
ɑ	A_ʔ \	faring terbuka kembali vokal pendek yang tidak dibulatkan	ل ر ر ل ل ر	/ˈs_ʔ \ A_ʔ \ l b/	a

IPA	X-SAMPA	Deskripsi	Contoh	Pengucapan	Viseme
æ:	a:	vokal panjang depan terbuka tengah yang tidak dibulatkan	الارار	/"b a: b/	a
ɑ:	A_? \:	faring terbuka kembali vokal panjang yang tidak dibulatkan	الاج	/"n A_? :. D_? \ aki_? \ dZ/	a
a	A	buka vokal pendek tengah yang tidak dibulatkan	wifi	/"w A j. f A j/	a
saya	saya	tegang dekat depan vokal pendek tidak bulat (MSA)	سسااق	/? i s. " X\ A_? \: q/	saya
ɛl	l	longgar tutup depan vokal pendek yang tidak dibulatkan	سرت	/"b l n t/	saya
saya	saya_? \	faring dekat depan vokal pendek tidak bulat	ل	/"t_? \ aki_? \ f l l/	saya

IPA	X-SAMPA	Deskripsi	Contoh	Pengucapan	Viseme
i:	i:	tutup depan vokal panjang yang tidak dibulatkan	س ر ا ل ل	/s a. " b i: l/	saya
saya:	saya_? :	faring dekat depan vokal panjang yang tidak dibulatkan	1)	/r A_? \. " t_? \ aki_? : b/	saya
u	u	tegang dekat kembali vokal pendek bulat (MSA)	م ر و ر ع	/"m u x. t a. r i? V	u
ʊ	U	longgar menutup kembali vokal pendek bulat	ر س و م	/r U. " s u: m/	u
u	u_? \	faring dekat belakang vokal pendek bulat	ا ل ر و ر	/? \ u_? \ s_? \. " f u: r/	u
u:	u:	tutup kembali vokal panjang bulat	م و ت	/"t u: t/	u
u:	u_? \:	faring dekat belakang vokal panjang bulat	و ر	/"s_? \ u_? \: r/	u

IPA	X-SAMPA	Deskripsi	Contoh	Pengucapan	Viseme
e	e	vokal pendek tengah depan yang tidak dibulatkan	رررالالالال	/'s e n t/	e
e:	e:	vokal panjang tengah depan yang tidak dibulatkan	ال	/'? e: S/	e
ɔ	O	vokal pendek bulat punggung tengah terbuka	ا ولار	/D O. " l A r/	O
ɔ:	O:	vokal panjang bulat punggung tengah terbuka	لون	/'l O: n/	O

Catalan (CA-es)

Tabel berikut mencantumkan fonem Alfabet Fonetik Internasional (IPA), simbol Alfabet Fonetik Metode Penilaian Pidato Diperpanjang (X-SAMPA), dan viseme yang sesuai untuk suara Catalan Arlet yang didukung oleh Amazon Polly.

Tabel Fonem/Viseme

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Konsonan				
p	p	plosif bilabial tak bersuara	p loure	p

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
t	t	plosif alveolar tak bersuara	T arragona	t
k	k	plosif velar tanpa suara	c om	k
b	b	plosif bilabial bersuara	b ata	p
d	d	plosif alveolar bersuara	en d oll	t
g	g	plosif velar bersuara	g ros	k
m	m	hidung bilabial bersuara	m anera	p
n	n	hidung alveolar bersuara	lakukan n ar	t
ɲ	J	hidung palatal bersuara	sebuah ny	J
ŋ	T	hidung velar bersuara	pi n güí	k
ɫ	5	pendekatan lateral alveolar velarisasi bersuara (l gelap)	a l bercoc	l
ʎ	L	pendekatan lateral palatal bersuara	ll op	J
r	r	getar alveolar bersuara	pa rr	r

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
r	4	keran alveolar bersuara	pa r a	t
f	f	frikatif labiodental tanpa suara	em f asi	f
detik	detik	frikatif alveolar tak bersuara	s ac	detik
z	z	frikatif alveolar bersuara	cal z es	detik
ʃ	D	frikatif postalveolar tanpa suara	gui x	D
ʒ	Z	frikatif postalveolar bersuara	col·le g i	D
tʃ	tS	africate postalveolar tanpa suara	co tx e	D
dʒ	dZ	africate postalveolar bersuara	pla tj	D
β	B	pendekatan bilabial bersuara	o b ert	B
ð	D	pendekatan gigi bersuara	menjadi d oll	T
j	j	pendekatan palatal bersuara	tidak saya	saya
ɣ	G	mendekati velar bersuara	pe g	k
v	v	frikatif labiodental bersuara	a f g à	f

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
w	w	pendekatan labiovelar bersuara	aig u	u
x	x	frikatif velar tanpa suara	J iménez	k
j	j\	frikatif palatal bersuara	y eso	J
l	l	pendekatan lateral alveolar bersuara	satu l ondra	t
θ	T	frikatif gigi tanpa suara	Gon z ález	T
Vokal				
a	a	buka kembali vokal	c a sa	a
e	e	vokal dekat-tengah depan yang tidak dibulatkan	ll e nya	e
ɛ	E	vokal terbuka tengah depan yang tidak dibulatkan	x e c	E
ɔ	ɔ	vokal depan tertutup yang tidak dibulatkan	v i sca	saya
o	o	vokal bulat dekat-tengah belakang	g o s	o
ɒ	O	vokal bulat punggung tengah terbuka	j o c	O

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
u	u	vokal bulat tertutup	u n	u
ə	@	vokal tengah tengah	cas	@
Simbol Tambahan				
'	"	stress primer	Ala ba ma	
,	%	stress sekunder	Sebuah labama	
.	.	batas suku kata	A.la.ba.ma	

Tionghoa (Kanton) (Yue-cn)

Tabel berikut mencantumkan fonem Jyutping dan Alfabet Fonetik Internasional (IPA) untuk suara Kanton yang didukung oleh Amazon Polly. Jyutping adalah sistem romanisasi bahasa Kanton yang biasa digunakan di dunia akademis dan di kalangan penutur bahasa Kanton. IPA dan X-SAMPA tidak umum digunakan tetapi tersedia untuk dukungan bahasa Inggris. Simbol IPA dan X-SAMPA dalam tabel hanya untuk referensi dan tidak boleh digunakan untuk transkripsi bahasa Mandarin. Contoh jyutping dan viseme yang sesuai juga ditampilkan.

Untuk membuat Amazon Polly menggunakan pengucapan fonetik dengan Jyutping, gunakan tag `phoneme alphabet="x-amazon-jyutping"`

Contoh berikut menunjukkan ini dengan masing-masing standar.

Jyutping:

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

IPA:

```
<speak>
```

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

X-SAMPEL:

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

Note

Amazon Polly menerima masukan Kanton yang dikodekan dalam UTF-8 saja.

Tabel Fonem/Viseme

Jyutping	IPA	X-SAMPA	Deskripsi	Contoh Jyutping	Viseme
Konsonan					
b	p	p	plosif bilabial tak bersuara	, b aa1	p
c	ts	ts_h	aspirasi afikat alveolar tanpa suara	, c aa1	detik
d	t	t	plosif alveolar tak bersuara	, d aa2	t
f	f	f	frikatif labiodental tanpa suara	, f aa1	f
g	k	k	plosif velar tanpa suara	1, g aa1	k
gw	k	k_w	plosif velar tanpa suara labialisasi	, gw aa1	u

Jyutping	IPA	X-SAMPA	Deskripsi	Contoh Jyutping	Viseme
-h	-h	-h	frikatif glottal tanpa suara	, h aa1	k
k	k	k_h	aspirasi velar plosif tanpa suara	, k aa1	k
kw	k·	k_wh	plosif velar tanpa suara aspirasi yang disedot labialisasi	, kw aa1	u
l	l	l	pendekatan lateral alveolar	, l aa1	t
m	m	m	hidung bilabial	, m aa1	p
m	m	m =	suku kata bilabial nasal	, m 4	p
ng	ŋ	T	hidung velar	, ng aa4	k
ng	ŋ	N =	suku kata velar nasal	, ng 4	k
n	n	n	hidung alveolar	, n aa4	t
p	p	p_h	plosif bilabial tanpa suara yang disedot	, p aa1	p
detik	detik	detik	frikatif alveolar tak bersuara	, s aa1	detik
t	t	t_h	plosif alveolar tak bersuara yang disedot	1, t aa1	t
w	w	w	pendekatan labio-velar	, w aa1	u
y	j	j	aproksiman palatal	, j aa5	saya

Jyutping	IPA	X-SAMPA	Deskripsi	Contoh Jyutping	Viseme
z	ts	ts	africate alveolar tak bersuara	, z aa1	detik
Vokal					
a	ɐ	6	vokal tengah yang hampir terbuka	, g a t1	a
aa	ɑ	A	buka kembali vokal yang tidak dibulatkan	1, g aa 1	a
aai	ai	Ai	diphthong	, g aai 1	a
aau	au	Au	diphthong	交, g aau 1	a
ai	i	6i	diphthong	, g ai 1	a
au	u	6u	diphthong	, k au 1	a
e	ɛ	E	vokal terbuka tengah depan yang tidak dibulatkan	, d e 1	E
ei	ei	ei	diphthong	, g ei 1	e
eo	ɵ	8	vokal bulat tengah dekat-tengah	, c eo n1	o
eo	y	8y	diftong	, g eo 1	o
eu	ɛu	EU	diftong	di, d eu 6	E
saya	saya	saya	tutup vokal depan yang tidak dibulatkan	, si 1	saya
saya	ɪ	ɪ	vokal tidak bulat dekat-dekat-dekat-depan	, gik 1	saya

Jyutping	IPA	X-SAMPA	Deskripsi	Contoh Jyutping	Viseme
iu	iu	iu	diftong	, g iu 1	saya
o	ɔ	O	vokal bulat punggung tengah terbuka	, g o 1	O
oe	œ	9	vokal bulat depan terbuka	, g oe 3	O
oi	ɔi	Oi	diphthong	, g oi 1	O
ou	ou	ou	diphthong	, g ou 1	o
u	u	u	tutup kembali vokal bulat	, g u 1	u
u	ʊ	U	vokal bulat dekat-dekat belakang	, g u k5	u
ui	ui	ui	diphthong	, g ui 6	u
yu	y	y	tutup vokal bulat depan	, j yu 1	u
Tanda nada dan Simbol Tambahan					
1			tingkat tinggi	, si 1	
2			naik sedang	, si 2	
3			tingkat menengah	, si 3	
4			tingkat yang sangat rendah	, si 4	
5			naik rendah	, si 5	
6			tingkat rendah	, si 6	
-	.	.	batas suku kata	jyu5-jam1	

Mandarin (Mandarin) (Cmn-cn)

Tabel berikut mencantumkan fonem Pinyin dan Alfabet Fonetik Internasional (IPA) untuk suara Mandarin Mandarin yang didukung oleh Amazon Polly. Pinyin adalah standar internasional untuk romanisasi standar China. IPA dan X-SAMPA tidak umum digunakan tetapi tersedia untuk dukungan bahasa Inggris. Simbol IPA dan X-SAMPA dalam tabel hanya untuk referensi dan tidak boleh digunakan untuk transkripsi bahasa Mandarin. Contoh Pinyin dan viseme yang sesuai juga ditampilkan.

Untuk membuat Amazon Polly menggunakan pengucapan fonetik dengan Pinyin, gunakan tag `phoneme alphabet="x-amazon-phonetic standard used"`

Contoh berikut menunjukkan ini dengan masing-masing standar.

Pinyin:

```
<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-SAMPEL:

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

Note

Amazon Polly menerima input Mandarin Mandarin yang dikodekan dalam UTF-8 saja. Standar pengkodean GB 18030 saat ini tidak didukung oleh Amazon Polly.

Tabel Fonem/Viseme

Pinyin	IPA	X-SAMPA	Deskripsi	Contoh Pinyin	Viseme
Konsonan					
f	f	f	frikatif labiodental tanpa suara	, f a1	f
-h	-h	-h	frikatif glottal tanpa suara	Dan, h e2	k
g	k	k	plosif velar tanpa suara	, g u3	k
k	k	k_h	aspirasi velar plosif tanpa suara	, k u3	k
l	l	l	aproksiman lateral alveolar	, l a1	t
m	m	m	hidung bilabial	, m a4	p
n	n	n	hidung alveolar	, n a4	t
ng	ŋ	T	hidung velar	, zhe ng 4	k
b	p	p	plosif bilabial tak bersuara	, b a4	p
p	p	p_h	plosif bilabial tanpa suara yang disedot	, p a4	p
detik	detik	detik	frikatif alveolar tak bersuara	, s i4	detik
x	ç	s\	frikatif alveolo-palatal tanpa suara	, x i1	J
sh	ʂ	s`	frikatif retroflex tanpa suara	, sh i4	D

Pinyin	IPA	X-SAMPA	Deskripsi	Contoh Pinyin	Viseme
d	t	t	plosif alveolar tak bersuara	, d a3	t
t	t	t_h	plosif alveolar tak bersuara yang disedot	1, t a1	t
zh	ʈ	t̚s̚	afrikate retroflex tanpa suara	, zh i1	D
ch	ʈ	t̚s̚_h	aspirasi retroflex tanpa suara affricate	, ch i1	D
detik	ʈs	ts	afrikate alveolar tak bersuara	, z i4	detik
j	ʈ̺	ts̺	afrikate alveolo-palatal tanpa suara	, j i1	J
q	ʈ̺	ts̺_h	afrikate alveolo-palatal tanpa suara aspirasi	, q i1	J
c	ʈs	ts_h	aspirasi afikat alveolar tanpa suara	1, c i4	detik
w	w	w	pendekatan labio-velar	, w o3	u
r	ʐ	z̥	frikatif retroflex bersuara	1, r i4	D

Suku kata berwarna “er” dan “r”

er	ə̯	@̥	vokal tengah tengah berwarna r	, er 4	@
-r			suku kata berwarna r	, xian r 4	@

Vokal

Pinyin	IPA	X-SAMPA	Deskripsi	Contoh Pinyin	Viseme
e	ɤ	7	vokal dekat-tengah belakang yang tidak dibulatkan	, e 4	e
e	ə	@	vokal tengah tengah	, e n1	@
a	a	a	buka vokal depan yang tidak dibulatkan	, sebuah n1	a
ai	aɪ	AI	diftong	, ai 4	a
ao	a	aU	diftong	, ao 4	a
ei	eɪ	e	diftong	, ei 4	e
e	ɛ	E	vokal terbuka tengah depan yang tidak dibulatkan	, dari e 3	E
saya	saya	saya	tutup vokal depan yang tidak dibulatkan	, j i 1	saya
ou	o	oU	diftong	, ou 1	o
o	ɔ	O	vokal bulat punggung tengah terbuka	, o 4	o
u	u	u	tutup kembali vokal bulat	, zh u 3	u
yu	y	y	tutup vokal bulat depan	, yu 2	u

Tanda nada dan Simbol Tambahan

1			nada tingkat tinggi	, yu1	
2			nada naik	, yu2	
3			nada rendah (jatuh naik)	, yu3	

Pinyin	IPA	X-SAMPA	Deskripsi	Contoh Pinyin	Viseme
4			nada jatuh	, yu4	
0			nada netral	, de0	
-	.	.	batas suku kata	yu3-yin1	

Ceko (cs-cz)

Tabel berikut mencantumkan fonem Alfabet Fonetik Internasional (IPA), simbol Alfabet Fonetik Metode Penilaian Pidato Diperpanjang (X-SAMPA), dan viseme yang sesuai untuk suara Ceko yang didukung oleh Amazon Polly.

Tabel Fonem/Viseme

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Konsonan				
p	p	plosif bilabial tak bersuara	p es	p
t	t	plosif alveolar tak bersuara	t ok	t
c	c	plosif palatal tanpa suara	t' uk	J
k	k	plosif velar tanpa suara	k os	k
b	b	plosif bilabial bersuara	b ez	p
d	d	plosif alveolar bersuara	d ok	t

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ɟ	ɟ\	plosif palatal bersuara	d' sebagai	J
g	g	plosif velar bersuara	g um	k
f	f	frikatif labiodental tanpa suara	f ilm	f
v	v	frikatif labiodental bersuara	v es	f
detik	detik	frikatif alveolar tak bersuara	s en	detik
z	z	frikatif alveolar bersuara	z el	detik
ʃ	D	frikatif postalveolar tanpa suara	š el	D
ʒ	Z	frikatif postalveolar bersuara	ž en	D
x	x	frikatif velar tanpa suara	ch di	k
ħ	-h	frikatif glottal bersuara	h kami	k
ts	ts	africate alveolar tak bersuara	c o	detik
tʃ	tS	africate postalveolar tanpa suara	č di	D
dʒ	dz	africate alveolar bersuara	špi c berský	detik

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
dʒ	dʒ	africate alveolar tak bersuara	dʒ di	D
m	m	hidung bilabial	m atau	p
n	n	hidung alveolar	n os	t
ɲ	J	hidung palatal	ñ ader	J
ŋ	T	hidung velar	ba n ka	k
r	r	getar alveolar bersuara	r ys	r
ɾ	r_ɾ	suara getar frikatif alveolar yang dinaikkan	ř ez	r
ɻ	r_0_ɻ	getar frikatif alveolar tanpa suara	ke ř	r
l	l	aproksiman lateral alveolar	l es	t
j	j	aproksiman palatal	j id	saya
w	w	aproksiman labiovelar	W atson	u
ɾ̥	r_ɾ̥	getaran alveolar bersuara suku kata	k k k	r
l̥	l_ɾ̥	pendekatan lateral alveolar suku kata	v l na	t
Vokal				

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
a	a	buka vokal depan yang tidak dibulatkan	l a n	a
a:	a:	vokal depan terbuka panjang yang tidak dibulatkan	l a n	a
ɛ	E	vokal terbuka tengah depan yang tidak dibulatkan	l e t	E
ɛ:	E:	vokal panjang terbuka tengah depan yang tidak dibulatkan	l e t	E
ɪ	l	vokal tidak bulat dekat-dekat-dekat- depan	b i t	saya
i:	i:	vokal panjang dekat depan yang tidak dibulatkan	b i t	saya
o	o	vokal bulat dekat- tengah belakang	h o l	o
o:	o:	vokal bulat panjang dekat-tengah belakang	g o l	o
u	u	tutup kembali vokal bulat	p u l	u

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
u:	u:	vokal bulat panjang dekat	p p p l	u
āu	au	diftong	au ke	a
ēu	Uni Eropa	diftong	eu ro	E
ōu	ou	diftong	m ou k	o
Simbol Tambahan				
'	“	stress primer		
.	.	batas suku kata		

Denmark (da-dk)

Tabel berikut mencantumkan fonem Alfabet Fonetik Internasional (IPA), simbol Alfabet Fonetik Metode Penilaian Pidato Diperpanjang (X-SAMPA), dan viseme yang sesuai untuk suara Denmark yang didukung oleh Amazon Polly.

Tabel Fonem/Viseme

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Konsonan				
b	b	plosif bilabial bersuara	b di	p
d	d	plosif alveolar bersuara	d	t
ð	D	frikatif gigi bersuara	ma d, th riller	T
f	f	frikatif labiodental tanpa suara	f di	f

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
g	g	plosif velar bersuara	g di	k
-h	-h	frikatif glottal tanpa suara	h di	k
j	j	aproksiman palatal	j	saya
k	k	plosif velar tanpa suara	k di	k
l	l	pendekatan lateral alveolar	l iklan	t
m	m	hidung bilabial	m di	p
n	n	hidung alveolar	n ay	t
ŋ	T	hidung velar	la ng	k
p	p	plosif bilabial tak bersuara	p ande	p
r	r	getar alveolar	th riller, sto r y	r
ʀ	R	frikatif uvular bersuara	r di	k
detik	detik	frikatif alveolar tak bersuara	s di	detik
t	t	plosif alveolar tak bersuara	t al	t
v	v	frikatif labiodental bersuara	v di	f

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
w	w	pendekatan labial-velar	ha v, w ekend	u
Vokal				
ø	2	vokal bulat dekat-tengah depan	ø st	o
ø:	2:	vokal bulat depan dekat-tengah panjang	ø se	o
e	6	vokal tengah hampir terbuka	m atau	a
œ	9	vokal bulat depan terbuka tengah	sk ø n, gr ø nt	O
:	9:	vokal bulat depan terbuka panjang	h ø ne, g ø re	O
ə	@	vokal tengah tengah	sebuah e	@
æ:	{:	vokal depan yang tidak dibulatkan yang panjang hampir terbuka	m a le	a
a	a	buka vokal depan yang tidak dibulatkan	m a n	a
æ	{	vokal depan yang tidak dibulatkan hampir terbuka	adr e sse	a

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ɑ	A	buka kembali vokal yang tidak dibulatkan	l a k, t a k	a
:	A:	vokal panjang terbuka belakang yang tidak dibulatkan	r a se	a
e	e	vokal dekat-tengah depan yang tidak dibulatkan	m i dt	e
e:	e:	vokal panjang dekat-tengah depan yang tidak dibulatkan	m e le	e
ɛ	E	vokal terbuka tengah depan yang tidak dibulatkan	m æ t	E
:	E:	vokal panjang terbuka tengah depan yang tidak dibulatkan	m æ le	E
saya	saya	tutup vokal depan yang tidak dibulatkan	m i t	saya
saya:	saya:	vokal panjang dekat depan yang tidak dibulatkan	m i le	saya
o	o	vokal bulat dekat- tengah belakang	f o t o	o

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
o:	o:	vokal bulat panjang dekat-tengah belakang	m o le	o
ɔ	O	vokal bulat punggung tengah terbuka	m u nd	O
:	O:	vokal bulat punggung tengah terbuka panjang	m ɔ̃ le	O
:	Q:	vokal bulat belakang terbuka panjang	m o rse	O
u	u	tutup kembali vokal bulat	l u sk	u
u:	u:	vokal bulat dekat belakang panjang	m u le	u
ʌ	V	punggung tengah terbuka tidak dibulatkan	kør er	E
y	y	tutup vokal bulat depan	y t	u
y:	y:	vokal bulat depan dekat panjang	h y le	u
Simbol Tambahan				
4	"	stress primer	Ala ba ma	
,	%	stress sekunder	Sebuah labama	

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
.	.	batas suku kata	A.la.ba.ma	

Belanda (Belgia) (NL-be)

Tabel berikut mencantumkan fonem Alfabet Fonetik Internasional (IPA), simbol Alfabet Fonetik Metode Penilaian Pidato Diperpanjang (X-SAMPA), dan viseme yang sesuai untuk suara Belanda Belgia (Flemish) yang didukung oleh Amazon Polly.

Tabel Fonem/Viseme

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Konsonan				
b	b	plosif bilabial bersuara	b ak	p
d	d	plosif alveolar bersuara	d ak	t
ɖ	dZ	africate postalveolar bersuara	mana g er	D
f	f	frikatif labiodental tanpa suara	f el	f
g	g	plosif velar bersuara	g oal	k
ɣ	G	frikatif velar bersuara	h oed	k
ɦ	h\	frikatif glottal bersuara	h dan	k
j	j	aproksiman palatal	j	saya

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
k	k	plosif velar tanpa suara	k ap	k
l	l	aproksiman lateral alveolar	l dan	t
m	m	hidung bilabial	m et	p
n	n	hidung alveolar	n et	t
ŋ	T	hidung velar	ba ng	k
p	p	plosif bilabial tak bersuara	p ak	p
r	r	getar alveolar	r dan	r
detik	detik	frikatif alveolar tak bersuara	s ein	detik
ʃ	D	frikatif postalveolar tanpa suara	sh ow	D
t	t	plosif alveolar tak bersuara	t ak	t
v	v	frikatif labiodental bersuara	v el	f
ʋ	\v	aproksiman labiodental	w itu	f
x	x	frikatif velar tanpa suara	ke ch	k
z	z	frikatif alveolar bersuara	z iin	detik

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ʒ	Z	frikatif postalveolar bersuara	baga g e	D

Vokal

ø:	2:	vokal bulat depan dekat-tengah panjang	n eu s	o
œy	9y	diphthong	b ui t	O
ə	@	vokal tengah tengah	d e	@
a:	a:	vokal depan terbuka panjang yang tidak dibulatkan	b aa d	a
:	A	buka kembali vokal yang tidak dibulatkan	b d	a
e:	e:	vokal panjang dekat-tengah depan yang tidak dibulatkan	b e t	e
:	3:	vokal panjang terbuka tengah yang tidak dibulatkan	barri dan re	E
ɛ	E	vokal terbuka tengah depan yang tidak dibulatkan	b e d	E

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ɛi	Ei	diphthong	b e t	E
saya	saya	tutup vokal depan yang tidak dibulatkan	v yaitu r	saya
ɪ	l	vokal tidak bulat dekat-dekat-dekat- depan	p i t	saya
o:	o:	vokal bulat panjang dekat-tengah belakang	b ya t	o
ɔ	O	vokal bulat punggung tengah terbuka	p o t	O
u	u	tutup kembali vokal bulat	h oe d	u
u	Vu	diphthong	f atau t	E
y:	y:	vokal bulat depan dekat yang panjang	f uu t	u
ʏ	Y	vokal bulat dekat- dekat-dekat-depan	h u t	u

Simbol Tambahan

'	"	stress primer	Ala ba ma	
,	%	stress sekunder	Sebuah labama	
.	.	batas suku kata	A.la.ba.ma	

Belanda (nl-NL)

Tabel berikut mencantumkan fonem Alfabet Fonetik Internasional (IPA), simbol Alfabet Fonetik Metode Penilaian Pidato Diperpanjang (X-SAMPA), dan viseme yang sesuai untuk suara Belanda yang didukung oleh Amazon Polly.

Tabel Fonem/Viseme

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Konsonan				
b	b	plosif bilabial bersuara	b ak	p
d	d	plosif alveolar bersuara	d ak	t
ɖ	dZ	africate postalveolar bersuara	mana g er	D
f	f	frikatif labiodental tanpa suara	f el	f
g	g	plosif velar bersuara	g oal	k
ɣ	G	frikatif velar bersuara	h oed	k
ɦ	h\	frikatif glottal bersuara	h dan	k
j	j	pendekatan palatal	j	saya
k	k	plosif velar tanpa suara	k ap	k
l	l	aproksiman lateral alveolar	l dan	t

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
m	m	hidung bilabial	m et	p
n	n	hidung alveolar	n et	t
ŋ	T	hidung velar	ba ng	k
p	p	plosif bilabial tak bersuara	p ak	p
r	r	getar alveolar	r dan	r
detik	detik	frikatif alveolar tak bersuara	s ein	detik
ʃ	D	frikatif postalveolar tanpa suara	sh ow	D
t	t	plosif alveolar tak bersuara	t ak	t
v	v	frikatif labiodental bersuara	v el	f
ʋ	\v	pendekatan labiodental	w itu	f
x	x	frikatif velar tanpa suara	ke ch	k
z	z	frikatif alveolar bersuara	z iin	detik
ʒ	Z	frikatif postalveolar bersuara	baga g e	D
Vokal				

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ø:	2:	vokal bulat depan dekat-tengah panjang	n eu s	o
œy	9y	diphthong	b ui t	O
ə	@	vokal tengah tengah	d e	@
a:	a:	vokal depan terbuka panjang yang tidak dibulatkan	b aa d	a
:	A	buka kembali vokal yang tidak dibulatkan	b a d	a
e:	e:	vokal panjang dekat-tengah depan yang tidak dibulatkan	b e t	e
:	3:	vokal panjang terbuka tengah yang tidak dibulatkan	barri dan re	E
ɛ	E	vokal terbuka tengah depan yang tidak dibulatkan	b e d	E
ɛi	Ei	diphthong	b e t	E
saya	saya	tutup vokal depan yang tidak dibulatkan	v yaitu r	saya

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ɪ	l	vokal tidak bulat dekat-dekat-dekat-depan	p i t	saya
o:	o:	vokal bulat panjang dekat-tengah belakang	b ya t	o
ɔ	O	vokal bulat punggung tengah terbuka	p o t	O
u	u	tutup kembali vokal bulat	h oe d	u
u	Vu	diphthong	f atau t	E
y:	y:	vokal bulat depan dekat panjang	f uu t	u
ʏ	Y	vokal bulat dekat- dekat-dekat-depan	h u t	u
Simbol Tambahan				
'	"	stress primer	Ala ba ma	
,	%	stress sekunder	Sebuah labama	
.	.	batas suku kata	A.la.ba.ma	

Inggris (AS) (en-US)

Tabel berikut mencantumkan fonem Alfabet Fonetik Internasional (IPA), simbol Alfabet Fonetik Metode Penilaian Pidato Diperpanjang (X-SAMPA), dan viseme yang sesuai untuk suara Inggris Amerika yang didukung oleh Amazon Polly.

Tabel Fonem/Viseme

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Konsonan				
b	b	plosif bilabial bersuara	b ed	p
d	d	plosif alveolar bersuara	d ig	t
ɖ	dZ	africate postalveolar bersuara	j ump	D
ð	D	frikatif gigi bersuara	th en	T
f	f	frikatif labiodental tanpa suara	f saya	f
g	g	plosif velar bersuara	g ame	k
-h	-h	frikatif glottal tanpa suara	h rumah	k
j	j	aproksiman palatal	y es	saya
k	k	plosif velar tanpa suara	c di	k
l	l	pendekatan lateral alveolar	l ay	l
m	m	hidung bilabial	m rumah	p
n	n	hidung alveolar	n ap	t
ŋ	T	hidung velar	thi ng	k

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
p	p	plosif bilabial tak bersuara	s p eak	p
ɹ	\r	pendekatan alveolar	r ed	r
detik	detik	frikatif alveolar tak bersuara	s eem	detik
ʃ	D	frikatif postalveolar tanpa suara	sh ip	D
t	t	plosif alveolar tak bersuara	t rap	t
t͡ʃ	tS	africate postalveolar tanpa suara	seni ch	D
θ	T	frikatif gigi tanpa suara	th di	T
v	v	frikatif labiodental bersuara	v est	f
w	w	pendekatan labial-velar	w est	u
z	z	frikatif alveolar bersuara	z ero	detik
ʒ	Z	frikatif postalveolar bersuara	vi s ion	D
Vokal				
ə	@	vokal tengah tengah	sebuah ren a	@

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ə	@`	vokal berwarna r tengah tengah	baca er	@
æ	{	dekat vokal terbuka terbuka yang tidak dibulatkan	tr ke p	a
aɪ	AI	diftong	pr i ce	a
a	aU	diftong	m ou th	a
ɑ	A	vokal terbuka panjang yang tidak dibulatkan	f satu lagi	a
eɪ	el	diftong	f a ce	e
ɜ̃	3`	buka vokal berwarna r tengah yang tidak dibulatkan	n ur se	E
ɛ	E	buka vokal tengah depan yang tidak dibulatkan	dr e ss	E
saya	saya	vokal panjang dekat depan yang tidak dibulatkan	fl e ce	saya
ɪ	I	vokal tidak bulat dekat-dekat-dekat- depan	k i t	saya
o	oU	diftong	g oa t	o

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ɔ	O	vokal bulat tengah belakang terbuka panjang	th ou ght	O
ɔɪ	Oɪ	diftong	ch oi ce	O
u	u	vokal bulat jarak dekat yang panjang	g oo se	u
ʊ	U	vokal bulat dekat-dekat belakang	f oo t	u
ʌ	V	open-mid-back vokal tidak dibulatkan	str u t	E
Simbol Tambahan				
'	"	stress primer	Ala ba ma	
,	%	stress sekunder	Sebuah labama	
.	.	batas suku kata	A.la.ba.ma	

Inggris (Australia) (en-AU)

Tabel berikut mencantumkan fonem Alfabet Fonetik Internasional (IPA), simbol Alfabet Fonetik Metode Penilaian Pidato Diperpanjang (X-SAMPA), dan viseme yang sesuai untuk suara Inggris Australia yang didukung oleh Amazon Polly.

Tabel Fonem/Viseme

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Konsonan				

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
b	b	plosif bilabial bersuara	b ed	p
d	d	plosif alveolar bersuara	d ig	t
ɖ	dZ	afrikate postalveolar bersuara	j ump	D
ð	D	frikatif gigi bersuara	th en	T
f	f	frikatif labiodental tanpa suara	f saya	f
g	g	plosif velar bersuara	g ame	k
-h	-h	frikatif glottal tanpa suara	h rumah	k
j	j	aproksiman palatal	y es	saya
k	k	plosif velar tanpa suara	c di	k
l	l	pendekatan lateral alveolar	l ay	t
ɭ	ɭ =	pendekatan lateral alveolar suku kata	batt le	t
m	m	hidung bilabial	m rumah	p
ɱ	ɱ =	suku kata bilabial nasal	anth em	p
n	n	hidung alveolar	n ap	t

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
né	n =	suku kata alveolar nasal	butto n	t
ŋ	T	hidung velar	thi ng	k
p	p	plosif bilabial tak bersuara	p di	p
ɹ	\r	pendekatan alveolar	r	r
detik	detik	frikatif alveolar tak bersuara	s eem	detik
ʃ	D	frikatif postalveolar tanpa suara	sh ip	D
t	t	plosif alveolar tak bersuara	t bertanya	t
t͡ʃ	tS	afrikate postalveolar tanpa suara	seni ch	D
θ	T	frikatif gigi tanpa suara	th di	T
v	v	frikatif labiodental bersuara	v est	f
w	w	pendekatan labial-velar	w	u
z	z	frikatif alveolar bersuara	z ero	detik
ʒ	Z	frikatif postalveolar bersuara	vi s ion	D

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Vokal				
ə	@	vokal tengah tengah	sebuah ren	@
ə	@U	diftong	g oa t	@
æ	{	dekat vokal terbuka terbuka yang tidak dibulatkan	tr ke p	a
aɪ	AI	diftong	pr i ce	a
a	aU	diftong	m ou th	a
ɑ:	A:	vokal terbuka panjang yang tidak dibulatkan	f satu lagi	a
eɪ	el	diftong	f a ce	e
:	3:	vokal panjang terbuka tengah yang tidak dibulatkan	n ur se	E
ɛ	E	buka vokal tengah depan yang tidak dibulatkan	dr e ss	E
ɛə	E@	diftong	squ adalah	E
saya:	saya	vokal panjang dekat depan yang tidak dibulatkan	fl e ce	saya

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ɛɪ	I	vokal tidak bulat dekat-dekat-dekat- depan	k i t	saya
ɪə	SAYA@	diftong	n telinga	saya
ɔ:	OI	vokal bulat punggung tengah terbuka panjang	th ou ght	O
ɔɪ	OI	Diftong	ch oi ce	O
ɒ	Q	buka kembali vokal bulat	l o t	O
u:	u:	vokal bulat jarak dekat yang panjang	g oo se	u
ʊ	U	vokal bulat dekat- dekat belakang	f oo t	u
ʊə	U@	diftong	c u re	u
ʌ	V	Open-mid- back vokal tidak dibulatkan	str u t	E
Simbol Tambahan				
ˈ	"	stress primer	Ala ba ma	
ˌ	%	stress sekunder	Sebuah labama	
.	.	batas suku kata	A.la.ba.ma	

Inggris (Inggris) (en-GB)

Tabel berikut mencantumkan fonem Alfabet Fonetik Internasional (IPA), simbol Alfabet Fonetik Metode Penilaian Pidato Diperpanjang (X-SAMPA), dan viseme yang sesuai untuk suara Inggris Inggris yang didukung oleh Amazon Polly.

Tabel Fonem/Viseme

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Konsonan				
b	b	plosif bilabial bersuara	b ed	p
d	d	plosif alveolar bersuara	d ig	t
ɹ̥	dZ	africate postalveolar bersuara	j ump	D
ð	D	frikatif gigi bersuara	th en	T
f	f	frikatif labiodental tanpa suara	f saya	f
g	g	plosif velar bersuara	g ame	k
-h	-h	frikatif glottal tanpa suara	h rumah	k
j	j	aproksiman palatal	y es	saya
k	k	plosif velar tanpa suara	c di	k
l	l	aproksiman lateral alveolar	l ay	t

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
l	l =	pendekatan lateral alveolar suku kata	batt le	t
m	m	hidung bilabial	m rumah	p
m	m =	suku kata bilabial nasal	anth em	p
n	n	hidung alveolar	n ap	t
né	n =	suku kata alveolar nasal	butto n	t
ŋ	T	hidung velar	thi ng	k
p	p	plosif bilabial tak bersuara	p di	p
ɹ	\r	pendekatan alveolar	r ed	r
detik	detik	frikatif alveolar tak bersuara	s eem	detik
ʃ	D	frikatif postalveolar tanpa suara	sh ip	D
t	t	plosif alveolar tak bersuara	t bertanya	t
t̚	tS	africate postalveolar tanpa suara	seni ch	D
θ	T	frikatif gigi tanpa suara	th di	T
v	v	frikatif labiodental bersuara	v est	f

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
w	w	pendekatan labial-velar	w	u
z	z	frikatif alveolar bersuara	z ero	detik
ʒ	Z	frikatif postalveolar bersuara	vi s ion	D

Vokal

ə	@	vokal tengah tengah	sebuah ren	@
ə	@U	diftong	g oa t	@
æ	{	dekat vokal terbuka terbuka yang tidak dibulatkan	tr ke p	a
aɪ	AI	diftong	pr i ce	a
a	aU	diftong	m ou th	a
ɑ:	A:	vokal terbuka panjang yang tidak dibulatkan	f satu lagi	a
eɪ	el	diftong	f a ce	e
:	3:	vokal panjang terbuka tengah yang tidak dibulatkan	n ur se	E
ɛ	E	buka vokal tengah depan yang tidak dibulatkan	dr e ss	E

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ɛə	E@	diftong	squ adalah	E
saya:	saya	vokal panjang dekat depan yang tidak dibulatkan	fl e ce	saya
ɛl	l	vokal tidak bulat dekat-dekat-dekat- depan	k i t	saya
ɪə	SAYA@	diftong	n telinga	saya
ɔ:	O:	vokal bulat punggung tengah terbuka panjang	th ou ght	O
ɔɪ	Oɪ	Diftong	ch oi ce	O
ɒ	Q	buka kembali vokal bulat	l o t	O
u:	u:	vokal bulat jarak dekat yang panjang	g oo se	u
ʊ	U	vokal bulat dekat- dekat belakang	f oo t	u
ʊə	U@	diftong	c u re	u
ʌ	V	Open-mid- back vokal tidak dibulatkan	str u t	E
Simbol Tambahan				
4	"	stress primer	Ala ba ma	

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
,	%	stress sekunder	Sebuah labama	
.	.	batas suku kata	A.la.ba.ma	

Inggris (India) (en-in)

Tabel berikut mencantumkan fonem Alfabet Fonetik Internasional (IPA), simbol Alfabet Fonetik Metode Penilaian Pidato Diperpanjang (X-SAMPA), dan viseme yang sesuai untuk suara Inggris India yang didukung oleh Amazon Polly.

Untuk fonem tambahan yang digunakan bersama dengan bahasa Inggris India, lihat. [Hindi \(Hi-in\)](#)

Tabel Fonem/Viseme

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Konsonan				
b	b	plosif bilabial bersuara	b ed	p
d	d	plosif alveolar bersuara	d ig	t
ɖ	dZ	africate postalveolar bersuara	j ump	D
ð	D	frikatif gigi bersuara	th en	T
f	f	frikatif labiodental tanpa suara	f saya	f
g	g	plosif velar bersuara	g ame	k

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
-h	-h	frikatif glottal tanpa suara	h rumah	k
j	j	aproksiman palatal	y es	saya
k	k	plosif velar tanpa suara	c di	k
l	l	pendekatan lateral alveolar	l ay	t
l	l =	pendekatan lateral alveolar suku kata	batt le	t
m	m	hidung bilabial	m rumah	p
m	m =	suku kata bilabial nasal	anth em	p
n	n	hidung alveolar	n ap	t
né	n =	suku kata alveolar nasal	n ap	t
ŋ	T	hidung velar	thi ng	k
p	p	plosif bilabial tak bersuara	p di	p
r	\r	pendekatan alveolar	r	r
detik	detik	frikatif alveolar tak bersuara	s eem	detik
ʃ	D	frikatif postalveolar tanpa suara	sh ip	D

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
t	t	plosif alveolar tak bersuara	t bertanya	t
ʈ	tS	afrikate postalveolar tanpa suara	seni ch	D
θ	T	frikatif gigi tanpa suara	th di	T
v	v	frikatif labiodental bersuara	v est	f
w	w	pendekatan labial-velar	w	u
z	z	frikatif alveolar bersuara	z ero	detik
ʒ	Z	frikatif postalveolar bersuara	vi s ion	D
Vokal				
ə	@	vokal tengah tengah	sebuah ren	@
ə	@U	diftong	g oa t	@
æ	{	dekat vokal terbuka terbuka yang tidak dibulatkan	tr ke p	a
aɪ	AI	diftong	pr i ce	a
a	aU	diftong	m ou th	a

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ɑ:	A:	vokal terbuka panjang yang tidak dibulatkan	f satu lagi	a
eɪ	el	diftong	f a ce	e
:	3:	vokal panjang terbuka tengah yang tidak dibulatkan	n ur se	E
ɛ	E	buka vokal tengah depan yang tidak dibulatkan	dr e ss	E
ɛə	E@	diftong	squ adalah	E
saya:	saya	vokal panjang dekat depan yang tidak dibulatkan	fl e ce	saya
ɛl	l	vokal tidak bulat dekat-dekat-dekat-depan	k i t	saya
ɪə	SAYA@	diftong	n telinga	saya
ɔ:	OI	vokal bulat punggung tengah terbuka panjang	th ou ght	O
ɔɪ	OI	Diftong	ch oi ce	O
ɒ	Q	buka kembali vokal bulat	l o t	O

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
u:	u:	vokal bulat jarak dekat yang panjang	g oo se	u
ʊ	U	vokal bulat dekat-dekat belakang	f oo t	u
ʊə	U@	diftong	c u re	u
ʌ	V	Open-mid-back vokal tidak dibulatkan	str u t	E
Simbol Tambahan				
4	"	stress primer	Ala ba ma	
,	%	stress sekunder	Sebuah labama	
.	.	batas suku kata	A.la.ba.ma	

Inggris (Irlandia) (en-IE)

Tabel berikut mencantumkan fonem Alfabet Fonetik Internasional (IPA), simbol Alfabet Fonetik Metode Penilaian Pidato Diperpanjang (X-SAMPA), dan viseme yang sesuai untuk suara Inggris Irlandia yang didukung oleh Amazon Polly.

Tabel Fonem/Viseme

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Konsonan				
b	b	plosif bilabial bersuara	b ed	p

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
d	d	plosif alveolar bersuara	d ig	t
ɖ	dʒ	afrikate postalveolar bersuara	j ump	D
ð	D	frikatif gigi bersuara	th en	T
f	f	frikatif labiodental tanpa suara	f saya	f
g	g	plosif velar bersuara	g ame	k
-h	-h	frikatif glottal tanpa suara	h rumah	k
j	j	aproksiman palatal	y es	saya
k	k	plosif velar tanpa suara	c di	k
l	l	aproksiman lateral alveolar	l ay	t
m	m	hidung bilabial	m rumah	p
n	n	hidung alveolar	n ap	t
ŋ	T	hidung velar	thi ng	k
p	p	plosif bilabial tak bersuara	s p eak	p
ɹ	\r	pendekatan alveolar	r	r

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
detik	detik	frikatif alveolar tak bersuara	s eem	detik
ʃ	D	frikatif postalveolar tanpa suara	sh ip	D
t	t	plosif alveolar tak bersuara	t rap	t
t̪	tS	africate postalveolar tanpa suara	seni ch	D
θ	T	frikatif gigi tanpa suara	th di	T
v	v	frikatif labiodental bersuara	v est	f
w	w	pendekatan labial-velar	w	u
z	z	frikatif alveolar bersuara	z ero	detik
ʒ	Z	frikatif postalveolar bersuara	vi s ion	D
Vokal				
ə	@	vokal tengah tengah	sebuah ren	@
æ	@`	vokal berwarna r tengah tengah	baca er	@
æ	{	dekat vokal terbuka terbuka yang tidak dibulatkan	tr ke p	a

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
aɪ	Aɪ	diftong	pr i ce	a
a	aU	diftong	m ou th	a
ɑ	A	vokal terbuka panjang yang tidak dibulatkan	f satu lagi	a
eɪ	el	diftong	f a ce	e
ɜ̃	3`	buka vokal berwarna r tengah yang tidak dibulatkan	n ur se	E
ɛ	E	buka vokal tengah depan yang tidak dibulatkan	dr e ss	E
saya	saya	vokal panjang dekat depan yang tidak dibulatkan	fl e ce	saya
ɪ	l	vokal tidak bulat dekat-dekat-dekat-depan	k i t	saya
o	oU	diftong	g oa t	o
ɔ	O	vokal bulat tengah belakang terbuka panjang	th ou ght	O
ɔɪ	Oɪ	diftong	ch oi ce	O

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
u	u	vokal bulat jarak dekat yang panjang	g oo se	u
ʊ	U	vokal bulat dekat-dekat belakang	f oo t	u
ʌ	V	open-mid-back vokal tidak dibulatkan	str u t	E
Simbol Tambahan				
'	"	stress primer	Ala ba ma	
,	%	stress sekunder	Sebuah labama	
.	.	batas suku kata	A.la.ba.ma	

Inggris (Selandia Baru) (en-NZ)

Tabel berikut mencantumkan fonem Alfabet Fonetik Internasional (IPA), simbol Alfabet Fonetik Metode Penilaian Pidato Diperpanjang (X-SAMPA), dan viseme yang sesuai untuk suara Inggris Selandia Baru yang didukung oleh Amazon Polly.

Tabel Fonem/Viseme

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Konsonan				
b	b	plosif bilabial bersuara	b ed	p
d	d	plosif alveolar bersuara	d ig	t

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ɖ	dZ	africate postalveolar bersuara	j ump	D
ð	D	frikatif gigi bersuara	th en	T
f	f	frikatif labiodental tanpa suara	f saya	f
g	g	plosif velar bersuara	g ame	k
-h	-h	frikatif glottal tanpa suara	h rumah	k
j	j	aproksiman palatal	y es	saya
k	k	plosif velar tanpa suara	c di	k
l	l	aproksiman lateral alveolar	l ay	t
l	l =	pendekatan lateral alveolar suku kata	batt le	t
m	m	hidung bilabial	m rumah	p
m	m =	suku kata bilabial nasal	anth em	p
n	n	hidung alveolar	n ap	t
né	n =	suku kata alveolar nasal	butto n	t
ŋ	T	hidung velar	thi ng	k

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
p	p	plosif bilabial tak bersuara	p di	p
ɹ	ɹ	pendekatan alveolar	r ed	r
detik	detik	frikatif alveolar tak bersuara	s eem	detik
ʃ	D	frikatif postalveolar tanpa suara	sh ip	D
t	t	plosif alveolar tak bersuara	t bertanya	t
tʃ	tS	africate postalveolar tanpa suara	seni ch	D
θ	T	frikatif gigi tanpa suara	th di	T
v	v	frikatif labiodental bersuara	v est	f
w	w	pendekatan labial-velar	w est	u
z	z	frikatif alveolar bersuara	z ero	detik
ʒ	Z	frikatif postalveolar bersuara	vi s ion	D
Vokal				
ə	@	vokal tengah tengah	sebuah ren	@

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ə	@U	diftong	g oa t	@
æ	{	dekat vokal terbuka terbuka yang tidak dibulatkan	tr ke p	a
aɪ	AI	diftong	pr i ce	a
a	aU	diftong	m ou th	a
ɑ:	A:	vokal terbuka panjang yang tidak dibulatkan	f satu lagi	a
eɪ	el	diftong	f a ce	e
:	3:	vokal panjang terbuka tengah yang tidak dibulatkan	n ur se	E
ɛ	E	buka vokal tengah depan yang tidak dibulatkan	dr e ss	E
ɛə	E@	diftong	squ adalah	E
saya:	saya	vokal panjang dekat depan yang tidak dibulatkan	fl e ce	saya
ɛl	l	vokal tidak bulat dekat-dekat-dekat- depan	k i t	saya
ɪə	SAYA@	diftong	n telinga	saya

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ɔ:	O:	vokal bulat punggung tengah terbuka panjang	th ou ght	O
ɔɪ	Oɪ	Diftong	ch oi ce	O
ɒ	Q	buka kembali vokal bulat	l o t	O
u:	u:	vokal bulat jarak dekat yang panjang	g oo se	u
ʊ	U	vokal bulat dekat- dekat belakang	f oo t	u
ʊə	U@	diftong	c u re	u
ʌ	V	Open-mid- back vokal tidak dibulatkan	str u t	E
Simbol Tambahan				
4	"	stress primer	Ala ba ma	
,	%	stress sekunder	Sebuah labama	
.	.	batas suku kata	A.la.ba.ma	

Suara Aria berbicara bahasa Inggris Selandia Baru dan menawarkan dukungan terbatas untuk Maori. Itu dapat mengucapkan kata-kata dan frasa Maori berikut. Ungkapan Maori peka huruf besar/kecil.

Bahasa Inggris	Maori
Halo/sorak-sorai	Kia ora

Bahasa Inggris	Maori
Selamat datang (ke)	Nau mai (ki)
Halo (satu orang) /terima kasih	Tēnā koe
Halo (tiga orang atau lebih) /terima kasih	Tēnā koutou
Selamat pagi	Ata mārie
Selamat pagi	Morena
Terima kasih	Ngā mihi
Jaga	Ngā manaakitanga
Sampai jumpa	Ka layang-layang
Sampai jumpa nanti	Mā te wā
Semoga hari yang baik	Kia pai tō rā
Selamat Natal	Meri Kirihimete
Maori	Māori
bahasa Maori	te reo Māori
Minggu bahasa Maori	Wiki o te reo Māori
Selandia Baru	Aotearoa
Tahun Baru Maori	Mātāriki
Kota di Selandia Baru/Hari Waitangi adalah hari nasional Selandia Baru	Waitangi
Satu	tahi
Dua	rua
Tiga	toru

Bahasa Inggris	Maori
Empat	whā
Lima	rima
Enam	ono
Tujuh	whitu
Delapan	waru
Sembilan	iwa
Sepuluh	tekau
Dua puluh	rua tekau
Tiga puluh	Toru tekau

Inggris (Singapura) (en-SG)

Tabel berikut mencantumkan fonem Alfabet Fonetik Internasional (IPA), simbol Alfabet Fonetik Metode Penilaian Pidato Diperpanjang (X-SAMPA), dan viseme yang sesuai untuk suara Inggris Singapura yang didukung oleh Amazon Polly.

Tabel Fonem/Viseme

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Konsonan				
b	b	plosif bilabial bersuara	b ed	p
d	d	plosif alveolar bersuara	d ig	t
ɹ̥	dʒ	africate postalveolar bersuara	j ump	D

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ð	D	frikatif gigi bersuara	th en	T
f	f	frikatif labiodental tanpa suara	f saya	f
g	g	plosif velar bersuara	g ame	k
-h	-h	frikatif glottal tanpa suara	h rumah	k
j	j	aproksiman palatal	y es	saya
k	k	plosif velar tanpa suara	c di	k
l	l	aproksiman lateral alveolar	l ay	t
l	l =	pendekatan lateral alveolar suku kata	batt le	t
m	m	hidung bilabial	m rumah	p
m	m =	suku kata bilabial nasal	anth em	p
n	n	hidung alveolar	n ap	t
né	n =	suku kata alveolar nasal	butto n	t
ŋ	T	hidung velar	thi ng	k
p	p	plosif bilabial tak bersuara	p di	p

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ɹ	ɹ	pendekatan alveolar	r	r
detik	detik	frikatif alveolar tak bersuara	s eem	detik
ʃ	D	frikatif postalveolar tanpa suara	sh ip	D
t	t	plosif alveolar tak bersuara	t bertanya	t
t͡ʃ	tS	africate postalveolar tanpa suara	seni ch	D
θ	T	frikatif gigi tanpa suara	th di	T
v	v	frikatif labiodental bersuara	v est	f
w	w	pendekatan labial-velar	w	u
z	z	frikatif alveolar bersuara	z ero	detik
ʒ	Z	frikatif postalveolar bersuara	vi s ion	D
Vokal				
ə	@	vokal tengah tengah	sebuah ren	@
ə	@U	diftong	g oa t	@

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
æ	{	dekat vokal terbuka terbuka yang tidak dibulatkan	tr ke p	a
aɪ	Aɪ	diftong	pr i ce	a
a	aU	diftong	m ou th	a
ɑ:	A:	vokal terbuka panjang yang tidak dibulatkan	f satu lagi	a
eɪ	el	diftong	f a ce	e
ɜ:	3:	vokal panjang terbuka tengah yang tidak dibulatkan	n ur se	E
ɛ	E	buka vokal tengah depan yang tidak dibulatkan	dr e ss	E
ɛə	E@	diftong	squ adalah	E
saya:	saya	vokal panjang dekat depan yang tidak dibulatkan	fl e ce	saya
ɛɪ	ɪ	vokal tidak bulat dekat-dekat-dekat- depan	k i t	saya
ɪə	SAYA@	diftong	n telinga	saya

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ɔ:	O:	vokal bulat punggung tengah terbuka panjang	th ou ght	O
ɔɪ	Oɪ	Diftong	ch oi ce	O
ɒ	Q	buka kembali vokal bulat	l o t	O
u:	u:	vokal bulat jarak dekat yang panjang	g oo se	u
ʊ	U	vokal bulat dekat- dekat belakang	f oo t	u
ʊə	U@	diftong	c u re	u
ʌ	V	Open-mid- back vokal tidak dibulatkan	str u t	E
Simbol Tambahan				
ˈ	"	stress primer	Ala ba ma	
ˌ	%	stress sekunder	Sebuah labama	
ˑ	.	batas suku kata	A.la.ba.ma	

Inggris (Afrika Selatan) (en-ZA)

Tabel berikut mencantumkan fonem Alfabet Fonetik Internasional (IPA), simbol Alfabet Fonetik Metode Penilaian Pidato Diperpanjang (X-SAMPA), dan viseme yang sesuai untuk suara Inggris Afrika Selatan yang didukung oleh Amazon Polly.

Tabel Fonem/Viseme

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Konsonan				
b	b	plosif bilabial bersuara	b ed	p
d	d	plosif alveolar bersuara	d ig	t
ɖ	dZ	africate postalveolar bersuara	j ump	D
ð	D	frikatif gigi bersuara	th en	T
f	f	frikatif labiodental tanpa suara	f saya	f
g	g	plosif velar bersuara	g ame	k
-h	-h	frikatif glottal tanpa suara	h rumah	k
j	j	pendekatan palatal	y es	saya
k	k	plosif velar tanpa suara	c di	k
l	l	pendekatan lateral alveolar	l ay	t
ɭ	l =	pendekatan lateral alveolar suku kata	batt le	t
ɬ	K	frikatif lateral tanpa suara	um hl anga	t

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
m	m	hidung bilabial	m rumah	p
m	m =	suku kata bilabial nasal	anth em	p
n	n	hidung alveolar	n ap	t
n	n =	suku kata alveolar nasal	bokong di	t
ŋ	T	hidung velar	thi ng	k
p	p	plosif bilabial tak bersuara	p di	p
ɹ	\r	pendekatan alveolar	r	r
r	r	getar alveolar	pa r ini	r
detik	detik	frikatif alveolar tak bersuara	s eem	detik
ʃ	D	frikatif postalveolar tanpa suara	sh ip	D
t	t	plosif alveolar tak bersuara	t bertanya	t
tʃ	tS	africate postalveolar tanpa suara	seni ch	D
θ	T	frikatif gigi tanpa suara	th di	T
v	v	frikatif labiodental bersuara	v est	f

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
w	w	pendekatan labial-velar	w est	u
x	x	frikatif velar tanpa suara	g mobil	k
z	z	frikatif alveolar bersuara	z ero	detik
!	!\	klik pasca-alveolar	gq eberha	k
	\	klik gigi	n c ube	t
	\	klik lateral	xh osa	t
Vokal				
ə	@	vokal tengah tengah	sebuah rena	@
əi	@i	diftong	nelspr ui t	saya
ə	@U	diftong	g oa t	@
æ	{	dekat vokal terbuka terbuka yang tidak dibulatkan	tr ke p	a
aɪ	AI	diftong	pr i ce	a
a	aU	diftong	m ou th	a
ɑ:	A:	vokal terbuka panjang yang tidak dibulatkan	f satu lagi	a
eɪ	el	diftong	f a ce	e

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
:	3:	vokal panjang terbuka tengah yang tidak dibulatkan	n ur se	E
ɛ	E	buka vokal tengah depan yang tidak dibulatkan	dr e ss	E
ɛə	E@	diftong	sq u adalah	E
saya:	saya	vokal panjang dekat depan yang tidak dibulatkan	fl e ce	saya
iə	SAYA@	diftong	du pr eez	saya
ɪ	I	vokal tidak bulat dekat-dekat-dekat- depan	k i t	saya
ɪə	SAYA@	diftong	n telinga	saya
ɔ:	O:	vokal bulat punggung tengah terbuka panjang	th ou ght	O
ɔɪ	OI	Diftong	ch oi ce	O
ɒ	Q	buka kembali vokal bulat	l o t	O
u:	u:	vokal bulat jarak dekat yang panjang	g oo se	u

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ʊ	U	vokal bulat dekat-dekat belakang	f oo t	u
ʊə	U@	diftong	c Anda e	u
ʌ	V	Open-mid-back vokal tidak dibulatkan	str u t	E
y	y	tutup vokal bulat depan	van v uu ren	u
Simbol Tambahan				
'	"	stress primer	Ala ba ma	
,	%	stress sekunder	Sebuah labama	
.	.	batas suku kata	A.la.ba.ma	

Bahasa Inggris (Welsh) (en-GB-WLS)

Tabel berikut mencantumkan fonem Alfabet Fonetik Internasional (IPA), simbol Alfabet Fonetik Metode Penilaian Pidato Diperpanjang (X-SAMPA), dan viseme yang sesuai untuk suara Bahasa Inggris Welsh yang didukung oleh Amazon Polly.

Tabel Fonem/Viseme

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Konsonan				
b	b	plosif bilabial bersuara	b ed	p
d	d	plosif alveolar bersuara	d ig	t

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ɹ̥	dZ	africate postalveolar bersuara	j ump	D
ð	D	frikatif gigi bersuara	th en	T
f	f	frikatif labiodental tanpa suara	f saya	f
g	g	plosif velar bersuara	g ame	k
-h	-h	frikatif glottal tanpa suara	h rumah	k
j	j	aproksiman palatal	y es	saya
k	k	plosif velar tanpa suara	c di	k
l	l	aproksiman lateral alveolar	l ay	t
l	l =	pendekatan lateral alveolar suku kata	batt le	t
m	m	hidung bilabial	m rumah	p
m	m =	suku kata bilabial nasal	anth em	p
n	n	hidung alveolar	n ap	t
né	n =	suku kata alveolar nasal	n ap	t
ŋ	T	hidung velar	thi ng	k

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
p	p	plosif bilabial tak bersuara	p di	p
ɹ	ɹ	pendekatan alveolar	r ed	r
detik	detik	frikatif alveolar tak bersuara	s eem	detik
ʃ	D	frikatif postalveolar tanpa suara	sh ip	D
t	t	plosif alveolar tak bersuara	t bertanya	t
t͡ʃ	tS	africate postalveolar tanpa suara	seni ch	D
θ	T	frikatif gigi tanpa suara	th di	T
v	v	frikatif labiodental bersuara	v est	f
w	w	pendekatan labial-velar	w est	u
z	z	frikatif alveolar bersuara	z ero	detik
ʒ	Z	frikatif postalveolar bersuara	vi s ion	D
Vokal				
ə	@	vokal tengah tengah	sebuah ren	@

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ə	@U	diftong	g oa t	@
æ	{	dekat vokal terbuka terbuka yang tidak dibulatkan	tr ke p	a
aɪ	AI	diftong	pr i ce	a
a	aU	diftong	m ou th	a
ɑ:	A:	vokal terbuka panjang yang tidak dibulatkan	f satu lagi	a
eɪ	el	diftong	f a ce	e
:	3:	vokal panjang terbuka tengah yang tidak dibulatkan	n ur se	E
ɛ	E	buka vokal tengah depan yang tidak dibulatkan	dr e ss	E
ɛə	E@	diftong	squ adalah	E
saya:	saya	vokal panjang dekat depan yang tidak dibulatkan	fl e ce	saya
ɛl	l	vokal tidak bulat dekat-dekat-dekat- depan	k i t	saya
ɪə	SAYA@	diftong	n telinga	saya

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ɔ:	OI	vokal bulat punggung tengah terbuka panjang	th ou ght	O
ɔɪ	OI	Diftong	ch oi ce	O
ɒ	Q	buka kembali vokal bulat	l o t	O
u:	u:	vokal bulat jarak dekat yang panjang	g oo se	u
ʊ	U	vokal bulat dekat- dekat belakang	f oo t	u
ʊə	U@	diftong	c u re	u
ʌ	V	Open-mid- back vokal tidak dibulatkan	str u t	E
Simbol Tambahan				
4	"	stress primer	Ala ba ma	
,	%	stress sekunder	Sebuah labama	
.	.	batas suku kata	A.la.ba.ma	

Finlandia (Fi-Fi)

Tabel berikut mencantumkan fonem Alfabet Fonetik Internasional (IPA), simbol Alfabet Fonetik Metode Penilaian Pidato Diperpanjang (X-SAMPA), dan viseme yang sesuai untuk suara Finlandia yang didukung oleh Amazon Polly.

Tabel Fonem/Viseme

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Konsonan Finlandia				
p	p	plosif bilabial tak bersuara	[p] kaki	p
t	t	plosif alveolar tak bersuara	[t] alo	t
k	k	plosif velar tanpa suara	[k] aali	k
d	d	plosif alveolar bersuara	[d] ata	t
detik	detik	frikatif alveolar tak bersuara	[s] ali	detik
-h	-h	frikatif glottal tanpa suara	[h] attu	k
u	\v	perkiraan labiodental bersuara	[v] aiva	v
j	j	aproksiman palatal	[j] oki	saya
l	l	aproksiman lateral alveolar	[l] oma	t
r	r	getar alveolar bersuara	[r] iita	r
m	m	hidung bilabial	[m] ato	p
n	n	hidung alveolar	[n] enää	t
ŋ	T	hidung velar	dia [ŋ] ki	k

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
-----	---------	-----------	--------	--------

Konsonan ditemukan dalam kata pinjaman

b	b	plosif bilabial bersuara	[b] ussi	p
f	f	frikatif labiodental tanpa suara	[f] irma	v
w	w	pendekatan labial-velar	[w] iki	u
z	z	frikatif alveolar bersuara	[z] ulu	detik
g	g	plosif velar bersuara	[g] aala	k
ʃ	D	frikatif postalveolar tanpa suara	[sh] akki	D
ʒ	Z	frikatif postalveolar bersuara	[g] enre	D
θ	T	frikatif gigi tanpa suara	telinga [th]	T
ð	D	frikatif gigi bersuara	ei [th] er	T

Vokal pendek

saya	saya	tutup vokal depan yang tidak dibulatkan	k [i] lo	saya
ɛ	E	buka vokal tengah depan yang tidak dibulatkan	k [e] untuk	E

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
æ	{	dekat vokal terbuka terbuka yang tidak dibulatkan	k [ä] ly	A
y	y	tutup vokal bulat depan	k [y] untuk	u
ø	2	tutup vokal bulat tengah depan	p [ö] ly	O
u	u	tutup kembali vokal bulat	k [u] lo	u
ɔ	O	buka vokal bulat bagian tengah belakang	k [o] lo	O
ɑ	A	buka kembali vokal yang tidak dibulatkan	k [a] la	A
Vokal panjang				
i:	i:	vokal panjang dekat depan yang tidak dibulatkan	s [ii] li	saya
ɛ:	E:	vokal panjang terbuka tengah depan yang tidak dibulatkan	[ya] tu	E
æ:	{:	vokal panjang dekat depan terbuka yang tidak dibulatkan	t [ää] llä	A

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
y:	y:	vokal panjang dekat depan yang tidak dibulatkan	t [yy] di	u
ø:	2:	vokal bulat tengah depan yang panjang	t [öö] lö	O
u:	u:	vokal bulat panjang dekat	t [uu] li	u
ɔ:	O:	vokal bulat tengah belakang terbuka panjang	r [oo] li	O
ɑ:	A:	vokal panjang terbuka belakang yang tidak dibulatkan	k [aa] su	A

Diphthong

ɛi	Ei	diphthong	l [ei] untuk	E
æi	{saya	diphthong	[äi] ti	A
ui	ui	diphthong	k [ui] n	u
ai	Ai	diphthong	k [ai] kki	A
ɔi	Oi	diphthong	p [oi] ka	O
øi	2i	diphthong	s [öi] n	O
yi	yi	diphthong	l [yi] jy	u
au	Au	diphthong	s [au] na	A

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ɔu	Ou	diphthong	k [ou] lu	O
ɛu	Uni Eropa	diphthong	r [eu] na	E
iu	iu	diphthong	v [iu] lu	saya
æy	{y	diphthong	t [äy] tidak	A
øɣ	2y	diphthong	k [öy] di	O
ɛy	Ey	diphthong	pes [ey] untuk	E
iy	iy	diphthong	käär [iy] untuk	saya
i	iE	diphthong	t [yaitu]	saya
yø	y2	diphthong	[yö]	u
u	uO	diphthong	t [uo]	u
Vokal ditemukan dalam kata pinjaman bahasa Inggris				
ɪ	l	vokal tidak bulat dekat-dekat-dekat-depan	b [i] t	saya
ʊ	U	vokal bulat dekat-dekat belakang	b [oo] k	u
ə	@	vokal tengah tengah	[a] pertarungan	@
ʌ	V	open-mid-back vokal tidak dibulatkan	c [u] t	E

Perancis (FR-fr)

Tabel berikut mencantumkan fonem Alfabet Fonetik Internasional (IPA), simbol Alfabet Fonetik Metode Penilaian Pidato Diperpanjang (X-SAMPA), dan viseme yang sesuai untuk suara Prancis yang didukung oleh Amazon Polly.

Tabel Fonem/Viseme

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Konsonan				
b	b	plosif bilabial bersuara	b oire	p
d	d	plosif alveolar bersuara	ma d ame	t
f	f	frikatif labiodental tanpa suara	f emme	f
g	g	plosif velar bersuara	g rand	k
ɥ	H	pendekatan labial-palatal	br u itu	u
j	j	pendekatan palatal	Mei ll eur	saya
k	k	plosif velar tanpa suara	q uatre	k
l	l	pendekatan lateral alveolar	ma l ade	t
m	m	hidung bilabial	m aison	p
n	n	hidung alveolar	astro n ome	t
ɲ	J	hidung palatal	bai gn er	J

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ŋ	T	hidung velar	taman	k
p	p	plosif bilabial tak bersuara	p omme	p
ʁ	R	frikatif uvular bersuara	amou r eux	k
detik	detik	frikatif alveolar tak bersuara	s anté	detik
ʃ	D	frikatif postalveolar tanpa suara	ch di	D
t	t	plosif alveolar tak bersuara	t éléphone	t
v	v	frikatif labiodental bersuara	v rai	f
w	w	pendekatan labial-velar	s o ir	u
z	z	frikatif alveolar bersuara	rai s pada	detik
ʒ	Z	frikatif postalveolar bersuara	auber g ine	D

Vokal

ø	2	vokal bulat dekat-tengah depan	d eu x	o
œ	9	vokal bulat depan terbuka tengah	n eu f	O

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
œ	9 ~	vokal bulat depan terbuka hidung	br un	O
ə	@	vokal tengah tengah	j e	@
a	a	buka vokal depan yang tidak dibulatkan	t ble	a
ɑ	A ~	hidung terbuka kembali vokal yang tidak dibulatkan	cam em bert	a
e	e	vokal dekat-tengah depan yang tidak dibulatkan	maret é	e
ɛ	E	vokal terbuka tengah depan yang tidak dibulatkan	n ei ge	E
ɛ	E ~	vokal terbuka bagian depan terbuka hidung yang tidak dibulatkan	getah di	E
ɪ	ɪ	tutup vokal depan yang tidak dibulatkan	m i lle	ɪ
o	o	vokal bulat dekat- tengah belakang	h ô pital	o

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ɔ	O	vokal bulat punggung tengah terbuka	h o mme	O
ɔ	O ~	vokal bulat punggung tengah terbuka hidung	b pada	O
u	u	tutup kembali vokal bulat	s ou s	u
y	y	tutup vokal bulat depan	d u r	u

Simbol Tambahan

4	"	stress primer	Ala ba ma	
,	%	stress sekunder	Sebuah labama	
.	.	batas suku kata	A.la.ba.ma	

Prancis (Belgia) (FR-be)

Tabel berikut mencantumkan fonem Alfabet Fonetik Internasional (IPA), simbol Alfabet Fonetik Metode Penilaian Pidato Diperpanjang (X-SAMPA), dan viseme yang sesuai untuk suara Prancis Belgia yang didukung oleh Amazon Polly.

Tabel Fonem/Viseme

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Konsonan				
b	b	plosif bilabial bersuara	b oire	p

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
d	d	plosif alveolar bersuara	ma d ame	t
f	f	frikatif labiodental tanpa suara	f emme	f
g	g	plosif velar bersuara	g rand	k
ɥ	H	aproksiman labial-palatal	br u itu	u
j	j	aproksiman palatal	mei ll eur	saya
k	k	plosif velar tanpa suara	q uatre	k
l	l	aproksiman lateral alveolar	ma l ade	t
m	m	hidung bilabial	m aison	p
n	n	hidung alveolar	astro n ome	t
ɲ	J	hidung palatal	bai gn er	J
ŋ	T	hidung velar	parki	k
p	p	plosif bilabial tak bersuara	p omme	p
ʁ	R	frikatif uvular bersuara	amou r eux	k
detik	detik	frikatif alveolar tak bersuara	s anté	detik

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ʃ	D	frikatif postalveolar tanpa suara	ch di	D
t	t	plosif alveolar tak bersuara	t éléphone	t
v	v	frikatif labiodental bersuara	v rai	f
w	w	pendekatan labial- velar	s o ir	u
z	z	frikatif alveolar bersuara	rai s pada	detik
ʒ	Z	frikatif postalveolar bersuara	auber g ine	D

Vokal

ø	2	vokal bulat dekat- tengah depan	d eu x	o
œ	9	vokal bulat depan terbuka	n eu f	O
œ	9 ~	vokal bulat depan terbuka hidung	br un	O
ə	@	vokal tengah tengah	j e	@
a	a	buka vokal depan yang tidak dibulatkan	t satu ble	a

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ɑ	A ~	hidung terbuka kembali vokal yang tidak dibulatkan	cam em bert	a
e	e	vokal dekat-tengah depan yang tidak dibulatkan	mare t é	e
ɛ	E	vokal terbuka tengah depan yang tidak dibulatkan	n ei ge	E
ɛ	E ~	vokal terbuka tengah hidung depan yang tidak dibulatkan	getah di	E
saya	saya	tutup vokal depan yang tidak dibulatkan	m i lle	saya
o	o	vokal bulat dekat- tengah belakang	h ô pital	o
ɔ	O	vokal bulat punggung tengah terbuka	h o mme	O
ɔ	O ~	vokal bulat punggung tengah terbuka hidung	b pada	O
u	u	tutup kembali vokal bulat	s ou s	u
y	y	tutup vokal bulat depan	d u r	u

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Simbol Tambahan				
4	"	stress primer	Ala ba ma	
,	%	stress sekunder	Sebuah labama	
.	.	batas suku kata	A.la.ba.ma	

Prancis (Kanada) (fr-Ca)

Tabel berikut mencantumkan fonem Alfabet Fonetik Internasional (IPA), simbol Alfabet Fonetik Metode Penilaian Pidato Diperpanjang (X-SAMPA), dan viseme yang sesuai untuk suara Kanada Prancis yang didukung oleh Amazon Polly.

Tabel Fonem/Viseme

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Konsonan				
b	b	plosif bilabial bersuara	b oire	p
d	d	plosif alveolar bersuara	ma d ame	t
f	f	frikatif labiodental tanpa suara	f emme	f
g	g	plosif velar bersuara	g rand	k
ɥ	H	pendekatan labial-palatal	br u itu	u
j	j	pendekatan palatal	Mei ll eur	saya

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
k	k	plosif velar tanpa suara	q uatre	k
l	l	pendekatan lateral alveolar	ma l ade	t
m	m	hidung bilabial	m aison	p
n	n	hidung alveolar	astro n ome	t
ɲ	J	hidung palatal	bai gn er	J
ŋ	T	hidung velar	taman	k
p	p	plosif bilabial tak bersuara	p omme	p
ʁ	R	frikatif uvular bersuara	amou r eux	k
detik	detik	frikatif alveolar tak bersuara	s anté	detik
ʃ	D	frikatif postalveolar tanpa suara	ch di	D
t	t	plosif alveolar tak bersuara	t éléphone	t
v	v	frikatif labiodental bersuara	v rai	f
w	w	pendekatan labial-velar	s o ir	u
z	z	frikatif alveolar bersuara	rai s pada	detik

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ʒ	Z	frikatif postalveolar bersuara	auber g ine	D

Vokal

ø	2	vokal bulat dekat-tengah depan	d eu x	o
œ	9	vokal bulat depan terbuka tengah	n eu f	O
œ	9 ~	vokal bulat depan terbuka hidung	br un	O
ə	@	vokal tengah tengah	j e	@
a	a	buka vokal depan yang tidak dibulatkan	t ble	a
ɑ	A ~	hidung terbuka kembali vokal yang tidak dibulatkan	cam em bert	a
e	e	vokal dekat-tengah depan yang tidak dibulatkan	maret é	e
ɛ	E	vokal terbuka tengah depan yang tidak dibulatkan	n ei ge	E

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ɛ	E ~	vokal terbuka bagian depan terbuka hidung yang tidak dibulatkan	getah di	E
saya	saya	tutup vokal depan yang tidak dibulatkan	m i lle	saya
o	o	vokal bulat dekat-tengah belakang	h ô pital	o
ɔ	O	vokal bulat punggung tengah terbuka	h o mme	O
ɔ	O ~	vokal bulat punggung tengah terbuka hidung	b pada	O
u	u	tutup kembali vokal bulat	s ou s	u
y	y	tutup vokal bulat depan	d u r	u

Simbol Tambahan

'	"	stress primer	Ala ba ma	
,	%	stress sekunder	Sebuah labama	
.	.	batas suku kata	A.la.ba.ma	

Jerman (De-de)

Tabel berikut mencantumkan fonem Alfabet Fonetik Internasional (IPA), simbol Alfabet Fonetik Metode Penilaian Pidato Diperpanjang (X-SAMPA), dan viseme yang sesuai untuk suara Jerman yang didukung oleh Amazon Polly.

Tabel Fonem/Viseme

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Konsonan				
4	ʔ	berhenti glotal		
b	b	plosif bilabial bersuara	B lebih	p
d	d	plosif alveolar bersuara	D ach	t
ç	C	frikatif palatal tanpa suara	saya ch	k
ɗ	dZ	africate postalveolar bersuara	Dsch ungel	D
f	f	Frikatif labiodental tanpa suara	V ogel	f
g	g	Velar plosif bersuara	G abel	k
-h	-h	Frikatif glottal tanpa suara	H aus	k
j	j	Frikatif glottal tanpa suara	j emand	saya
k	k	Plosif velar tanpa suara	K leid	k

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
l	l	Pendekatan lateral alveolar	L dan	t
m	m	Hidung bilabial	M ilch	p
n	n	Hidung alveolar	N atur	t
ŋ	T	Velar hidung	kli	k
p	p	Plosif bilabial tanpa suara	P bahtera	p
pf	pf	Afrikat labiodental tanpa suara	Sebuah pf el	
ʀ	R	Getaran uvular	R egen	
ɖ	detik	frikatif alveolar tak bersuara	Saya ss er	detik
ʃ	D	Frikatif postalveolar tanpa suara	Fi sch er	D
t	t	Plosif alveolar tak bersuara	T opf	T
ts	Ts	Afrikat alveolar tanpa suara	Z ahl	
tʃ	tS	Afrikat postalveolar tanpa suara	deu tsch	D
v	v	Frikatif labiodental bersuara	W assen	f
x	x	Frikatif velar tanpa suara	ko ch en	k

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
z	z	Frikatif alveolar bersuara	S	detik
ʒ	Z	Frikatif postalveolar bersuara	Oran g e	D
Vokal				
ø:	2:	vokal bulat depan dekat-tengah panjang	b ö se	o
ɐ	6	vokal tengah yang hampir terbuka	bess er	a
ɐ̯	6_ ^	vokal tengah hampir terbuka non-suku kata	Kl	a
œ	9	vokal bulat depan terbuka	k ö nnen	O
ə	@	vokal tengah tengah	Merah e	@
a	a	buka vokal depan yang tidak dibulatkan	S ke lz	a
a:	a:	vokal depan terbuka panjang yang tidak dibulatkan	S ah tidak	a
aɪ	AI	diftong	n i n	a
a	aU	diftong	Au gen	a

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ɑ	A ~	hidung terbuka kembali vokal yang tidak dibulatkan	Semut restaur	a
e:	e:	vokal panjang dekat-tengah depan yang tidak dibulatkan	R e	e
ɛ	E	vokal terbuka tengah depan yang tidak dibulatkan	K e ller	E
ɛ	E ~	vokal terbuka tengah hidung depan yang tidak dibulatkan	Terr ain	E
saya:	saya:	vokal panjang dekat depan yang tidak dibulatkan	L yaitu d	saya
ɛl	l	vokal tidak bulat dekat-dekat-dekat- depan	b i tte	saya
o:	o:	vokal bulat panjang dekat-tengah belakang	K oh l	o
ɔ	O	vokal bulat punggung tengah terbuka	K o ffer	O
ɔ	O ~	vokal bulat punggung tengah terbuka hidung	Ann di ce	O

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ɔ	OY	diftong	n eu	O
u:	u:	vokal bulat panjang dekat	Br u der	u
ʊ	U	vokal bulat dekat-dekat belakang	W u nder	u
y:	y:	vokal bulat depan dekat panjang	k ü h l	u
ʏ	Y	vokal bulat dekat-dekat-depan	K ü che	u
Simbol Tambahan				
ˈ	"	stress primer	Ala ba ma	
ˌ	%	stress sekunder	Sebuah labama	
·	.	batas suku kata	A.la.ba.ma	

Jerman (Austria) (de-at)

Tabel berikut mencantumkan fonem Alfabet Fonetik Internasional (IPA), simbol Alfabet Fonetik Metode Penilaian Pidato Diperpanjang (X-SAMPA), dan viseme yang sesuai untuk suara Jerman Austria yang didukung oleh Amazon Polly.

Tabel Fonem/Viseme

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Konsonan				
ʔ	?	berhenti glotal		
b	b	plosif bilabial bersuara	B lebih	p

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
d	d	plosif alveolar bersuara	D ach	t
ç	C	frikatif palatal tanpa suara	saya ch	k
ɖ	dZ	africate postalveolar bersuara	Dsch ungel	D
f	f	Frikatif labiodental tanpa suara	V ogel	f
g	g	Velar plosif bersuara	G abel	k
-h	-h	Frikatif glottal tanpa suara	H aus	k
j	j	Frikatif glottal tanpa suara	j emand	saya
k	k	Plosif velar tanpa suara	K leid	k
l	l	Pendekatan lateral alveolar	L dan	t
m	m	Hidung bilabial	M ilch	p
n	n	Hidung alveolar	N atur	t
ŋ	T	Velar hidung	kli	k
p	p	Plosif bilabial tanpa suara	P bahtera	p
ɸ	pf	Afrikat labiodental tanpa suara	Sebuah pf el	

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
R	R	Getaran uvular	R egen	
detik	detik	frikatif alveolar tak bersuara	Saya ss er	detik
ʃ	D	Frikatif postalveolar tanpa suara	Fi sch er	D
t	t	Plosif alveolar tak bersuara	T opf	T
ʦ	Ts	Afrikat alveolar tanpa suara	Z ahl	
tʃ	tS	Afrikat postalveolar tanpa suara	deu tsch	D
v	v	Frikatif labiodental bersuara	W ass er	f
x	x	Frikatif velar tanpa suara	ko ch en	k
z	z	Frikatif alveolar bersuara	S	detik
ʒ	Z	Frikatif postalveolar bersuara	Oran g e	D
Vokal				
ø:	2:	vokal bulat depan dekat-tengah panjang	b ö se	o
e	6	vokal tengah yang hampir terbuka	bess er	a

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ɐ	6_^	vokal tengah hampir terbuka non-suku kata	Kl	a
œ	9	vokal bulat depan terbuka	k ö nnen	O
ə	@	vokal tengah tengah	Merah e	@
a	a	buka vokal depan yang tidak dibulatkan	S ke lz	a
a:	a:	vokal depan terbuka panjang yang tidak dibulatkan	S ah tidak	a
aɪ	AI	diftong	n i n	a
a	aU	diftong	Au gen	a
ɑ	A ~	hidung terbuka kembali vokal yang tidak dibulatkan	Semut restaur	a
e:	e:	vokal panjang dekat-tengah depan yang tidak dibulatkan	R e	e
ɛ	E	vokal terbuka tengah depan yang tidak dibulatkan	K e ller	E

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ɛ	E ~	vokal terbuka tengah hidung depan yang tidak dibulatkan	Terr ain	E
saya:	saya:	vokal panjang dekat depan yang tidak dibulatkan	L yaitu d	saya
ɛl	l	vokal tidak bulat dekat-dekat-dekat- depan	b i tte	saya
o:	o:	vokal bulat panjang dekat-tengah belakang	K oh l	o
ɔ	O	vokal bulat punggung tengah terbuka	K o ffer	O
ɔ	O ~	vokal bulat punggung tengah terbuka hidung	Ann di ce	O
ɔ	OY	diftong	n eu	O
u:	u:	vokal bulat panjang dekat	Br u der	u
ʊ	U	vokal bulat dekat- dekat belakang	W u nder	u
y:	y:	vokal bulat depan dekat panjang	k ü h l	u

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ʏ	Y	vokal bulat dekat-dekat-depan	K ü che	u
Simbol Tambahan				
4	"	stress primer	Ala ba ma	
,	%	stress sekunder	Sebuah labama	
.	.	batas suku kata	A.la.ba.ma	

Jerman (standar Swiss) (De-ch)

Tabel berikut mencantumkan fonem Alfabet Fonetik Internasional (IPA), simbol Alfabet Fonetik Metode Penilaian Pidato Diperpanjang (X-SAMPA), dan viseme yang sesuai untuk suara Jerman (standar Swiss) yang didukung oleh Amazon Polly.

Tabel Fonem/Viseme

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Konsonan				
4	?	berhenti glotal		
b	b	plosif bilabial bersuara	B lebih	p
d	d	plosif alveolar bersuara	D ach	t
ç	C	frikatif palatal tanpa suara	saya ch	k
ɖ	dZ	africate postalveolar bersuara	Dsch ungel	D

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
f	f	Frikatif labiodental tanpa suara	V ogel	f
g	g	Velar plosif bersuara	G abel	k
-h	-h	Frikatif glottal tanpa suara	H aus	k
j	j	Frikatif glottal tanpa suara	j emand	saya
k	k	Plosif velar tanpa suara	K leid	k
l	l	Pendekatan lateral alveolar	L dan	t
m	m	Hidung bilabial	M ilch	p
n	n	Hidung alveolar	N atur	t
ŋ	T	Velar hidung	kli	k
p	p	Plosif bilabial tanpa suara	P bahtera	p
p & #x0201d	pf	Afrikat labiodental tanpa suara	Sebuah pf el	
R	R	Getaran uvular	R egen	
detik	detik	frikatif alveolar tak bersuara	Saya ss er	detik
ʃ	D	Frikatif postalveolar tanpa suara	Fi sch er	D

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
t	t	Plosif alveolar tak bersuara	T opf	T
ts	Ts	Afrikat alveolar tanpa suara	Z ahl	
t̪	tS	Afrikat postalveolar tanpa suara	deu tsch	D
v	v	Frikatif labiodental bersuara	W assen	f
x	x	Frikatif velar tanpa suara	ko ch en	k
z	z	Frikatif alveolar bersuara	S e	detik
ʒ	Z	Frikatif postalveolar bersuara	Oran g e	D
Vokal				
ø:	2:	vokal bulat depan dekat-tengah panjang	b ö se	o
æ	6	vokal tengah yang hampir terbuka	bess er	a
æ̯	6_^	vokal tengah hampir terbuka non-suku kata	Kl ar	a
œ	9	vokal bulat depan terbuka	k ö nnen	O

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ə	@	vokal tengah tengah	Merah e	@
a	a	buka vokal depan yang tidak dibulatkan	S ke lz	a
a:	a:	vokal depan terbuka panjang yang tidak dibulatkan	S ah tidak	a
aɪ	AI	diftong	n i i n	a
a	aU	diftong	Au gen	a
ɑ	A ~	hidung terbuka kembali vokal yang tidak dibulatkan	Semut restaur	a
e:	e:	vokal panjang dekat-tengah depan yang tidak dibulatkan	R e de	e
ɛ	E	vokal terbuka tengah depan yang tidak dibulatkan	K e ller	E
ɛ	E ~	vokal terbuka tengah hidung depan yang tidak dibulatkan	Terr ain	E
saya:	saya:	vokal panjang dekat depan yang tidak dibulatkan	L yaitu d	saya

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ɛl	l	vokal tidak bulat dekat-dekat-dekat-depan	b i tte	saya
o:	o:	vokal bulat panjang dekat-tengah belakang	K oh l	o
ɔ	O	vokal bulat punggung tengah terbuka	K o ffer	O
ɔ	O ~	vokal bulat punggung tengah terbuka hidung	Ann di ce	O
ɔ	OY	diftong	n eu	O
u:	u:	vokal bulat panjang dekat	Br u der	u
ʊ	U	vokal bulat dekat- dekat belakang	W u nder	u
y:	y:	vokal bulat depan dekat panjang	k ü h l	u
ʏ	Y	vokal bulat dekat- dekat-dekat-depan	K ü che	u
Simbol Tambahan				
4	“	stress primer	Ala ba ma	
,	%	stress sekunder	Sebuah labama	
.	.	batas suku kata	A.la.ba.ma	

Hindi (Hi-in)

Tabel berikut mencantumkan fonem Alfabet Fonetik Internasional (IPA), simbol Alfabet Fonetik Metode Penilaian Pidato Diperpanjang (X-SAMPA), dan jenis suara fonem untuk suara Hindi yang didukung oleh Amazon Polly.

Untuk fonem tambahan yang digunakan bersama dengan bahasa Hindi, lihat. [Inggris \(India\) \(en-in\)](#)

Tabel Fonem/Viseme

IPA	X-SAMPA	Deskripsi	Contoh
Konsonan			
p	p_h	plosif bilabial aspirasi tanpa suara	(fool)
b	b_h	plosif bilabial aspirasi bersuara	(bhaari)
t	t_d	plosif gigi tanpa suara	(taapmaan)
t	t_d_h	plosif gigi aspirasi tanpa suara	(thoda)
d	d_d	plosif gigi bersuara	(dilli)
d	d_d_h	plosif gigi aspirasi bersuara	(dhobi)
ʈ	t`	retroflex plosif tanpa suara	(katora)
ʈʰ	t`_h	plosif retroflex aspirasi tanpa suara	(thand)
ɖ	d`	plosif retroflex bersuara	(darr)
ɖʱ	d`_h	plosif retroflex aspirasi bersuara	(dhal)

IPA	X-SAMPA	Deskripsi	Contoh
tʃ	TS_h	africate palatal aspirasi tanpa suara	(chaal)
dʒ	Dz_h	africate palatal aspirasi bersuara	(jhaal)
k	k_h	plosif velar aspirasi tanpa suara	(khan)
g	g_h	plosif velar aspirasi bersuara	(ghaan)
ŋ	n`	retroflex hidung	(kshan)
r	4	tutup alveolar	(ram)
d	r`	tutup retroflex polos	(bada)
ɽʱ	r`_h	tutup retroflex aspirasi bersuara	(barhi)
ʋ	\v	aproksiman bilabial	(wasool)
Vokal			
ə	@_o	vokal tengah tengah	(achhaa)
ə	@~	vokal tengah tengah nasalisasi	(hansnaa)
a	A_o	buka vokal depan yang tidak dibulatkan	(aag)
a	A ~	vokal depan terbuka yang tidak dibulatkan	(ghariyaan)
ɛl	l_o	vokal tidak bulat dekat-dekat-dekat-depan	(ikkees)

IPA	X-SAMPA	Deskripsi	Contoh
ɪ	SAYA~	nasalisasi dekat-dekat dekat vokal depan yang tidak dibulatkan	(sinchai)
saya	i_o	tutup vokal depan yang tidak dibulatkan	(billee)
saya	saya~	vokal depan tertutup yang tidak dibulatkan	(nahin)
ʊ	U_o	vokal bulat dekat-dekat belakang	(ullu)
ʊ	U ~	vokal bulat dekat-dekat belakang nasalisasi	(munh)
u	u_o	tutup kembali vokal bulat	(fool)
u	u ~	vokal bulat dekat yang dinasalisasi	(tidak)
ɔ	O_o	vokal bulat punggung tengah terbuka	(11/2)
ɔ	O ~	vokal bulat punggung tengah terbuka yang dinasalisasi	(bhaun)
o	o	vokal bulat dekat-tengah belakang	(sona)
o	o ~	vokal bulat dekat-tengah belakang yang dinasalis asi	(kyon)

IPA	X-SAMPA	Deskripsi	Contoh
ɛ	E_o	vokal terbuka tengah depan yang tidak dibulatkan	(paissa)
ɛ	E ~	vokal terbuka tengah depan yang tidak dibulatkan	(utama)
e	e	vokal dekat-tengah depan yang tidak dibulatkan	(ek)
e	e ~	vokal tidak bulat depan dekat-tengah yang dinasalisasi	(kitabein)

Islandia (IS-IS)

Tabel berikut mencantumkan fonem Alfabet Fonetik Internasional (IPA), simbol Alfabet Fonetik Metode Penilaian Pidato Diperpanjang (X-SAMPA), dan viseme yang sesuai untuk suara Islandia yang didukung oleh Amazon Polly.

Tabel Fonem/Viseme

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Konsonan				
b	b	plosif bilabial bersuara	gras b akkanum	0
c	c	plosif palatal tanpa suara	pak k di	k
c	c_h	plosif palatal tanpa suara yang disedot	anar k istai	k

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ç	C	frikatif palatal tanpa suara	h	k
d	d	plosif alveolar bersuara	bón d i	t
ð	D	frikatif gigi bersuara	bor ð	T
f	f	frikatif labiodental tanpa suara	du f t	f
g	g	plosif velar bersuara	hol g óma	k
ɣ	G	frikatif velar bersuara	hu g ur	k
-h	-h	frikatif glottal tanpa suara	h eili	k
j	j	aproksiman palatal	j okull	saya
k	k_h	aspirasi velar plosif tanpa suara	ós k opunum	k
l	l	pendekatan lateral alveolar	go l f	t
l	l_0	aproksiman lateral alveolar tak bersuara	fo l k	t
m	m	hidung bilabial	septe m ber	p
m	m_0	hidung bilabial tanpa suara	ko m pa	p

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
n	n	hidung alveolar	n úmer	t
n	n_0	hidung alveolar tanpa suara	pö n tun	t
ɲ	J	hidung palatal	pæli n gar	J
ŋ	T	hidung velar	sö n gvarann	k
ŋ	N_0	hidung velar tanpa suara	fræ n ka	k
p	p_h	plosif bilabial tanpa suara yang disedot	af p lánun	p
r	r	getar alveolar	afsk r ifta	r
r	r_0	getar alveolar tak bersuara	andvö r pum	r
detik	detik	frikatif alveolar tak bersuara	baðhu s	detik
t	t_h	plosif alveolar tak bersuara yang disedot	t anki	t
θ	T	frikatif gigi tanpa suara	eldökki	T
v	v	frikatif labiodental bersuara	sil f ur	f
w	w	pendekatan labial- velar		u
x	x	frikatif velar tanpa suara	samféla g s	k

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Vokal				
œ	9	vokal bulat depan terbuka tengah	þr ö skuldinum	O
œ:	9:	vokal bulat depan terbuka panjang	tv ö	O
a	a	buka vokal depan yang tidak dibulatkan	nefn	a
a:	a:	vokal depan terbuka panjang yang tidak dibulatkan	f a ra	a
au	au	diftong	á tta	a
au:	au:	diftong	á tjan	a
ɛ	E	vokal terbuka tengah depan yang tidak dibulatkan	k e nnari	E
:	E:	vokal panjang terbuka tengah depan yang tidak dibulatkan	dr e ka	E
saya	saya	tutup vokal depan yang tidak dibulatkan	Gúl í ver	saya
saya:	saya:	vokal panjang dekat depan yang tidak dibulatkan	r r	saya

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ɪ	I	vokal tidak bulat dekat-dekat-dekat- depan	samsp i l	saya
ɛl:	SAYA:	vokal panjang dekat-dekat dekat- depan yang tidak dibulatkan	st i g	saya
ɔ	O	vokal bulat punggung tengah terbuka	regndr o par	O
:	O:	vokal bulat punggung tengah terbuka panjang	ullarb o lur	O
ɔu	Ou	diftong	t o lf	O
ɔu:	Ou:	diftong	fj o rir	O
u	u	tutup kembali vokal bulat	st lkan	u
u:	u:	vokal bulat dekat belakang panjang	fr ú	u
ʏ	Y	vokal bulat dekat- dekat depan	t di u	u
:	Y	vokal bulat dekat- dekat depan yang panjang	gr u ninn	u
Simbol Tambahan				
4	"	stress primer	Ala ba ma	

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
,	%	stress sekunder	Sebuah labama	
.	.	batas suku kata	A.la.ba.ma	

Italia (IT-it)

Tabel berikut mencantumkan fonem Alfabet Fonetik Internasional (IPA), simbol Alfabet Fonetik Metode Penilaian Pidato Diperpanjang (X-SAMPA), dan viseme yang sesuai untuk suara Italia yang didukung oleh Amazon Polly.

Tabel Fonem/Viseme

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Konsonan				
b	b	plosif bilabial bersuara	b acca	p
d	d	plosif alveolar bersuara	d ama	t
ɖz	dz	afrikate alveolar bersuara	z ero	detik
ɖ	dZ	afrikate postalveolar bersuara	g iro	D
f	f	frikatif labiodental tanpa suara	f amiglia	f
g	g	plosif velar bersuara	g atto	k
-h	-h	frikatif glottal tanpa suara	h oror	k

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
j	j	aproksiman palatal	d i eci	saya
k	k	plosif velar tanpa suara	c ampo	k
l	l	aproksiman lateral alveolar	l ido	t
ʎ	L	aproksiman lateral palatal	sebuah gli o	J
m	m	hidung bilabial	m	p
n	n	hidung alveolar	n ove	t
ɲ	J	hidung palatal	rasa gn e	J
p	p	plosif bilabial tak bersuara	p izza	p
r	r	getar alveolar	r isata	r
detik	detik	frikatif alveolar tak bersuara	s ei	detik
ʃ	D	frikatif postalveolar tanpa suara	sci enza	D
t	t	plosif alveolar tak bersuara	t avola	t
ts	ts	africate alveolar tak bersuara	untuk z a	detik
t̪	tS	africate postalveolar tanpa suara	ci elo	D

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
v	v	frikatif labiodental bersuara	v enti	f
w	w	pendekatan labial-velar	q di attro	u
z	z	frikatif alveolar bersuara	bi s ogno	detik
ʒ	Z	frikatif postalveolar bersuara	bi j ou	D
Vokal				
a	a	buka vokal depan yang tidak dibulatkan	sebuah rco	a
e	e	vokal dekat-tengah depan yang tidak dibulatkan	tr e	e
ɛ	E	vokal terbuka tengah depan yang tidak dibulatkan	e ttaro	E
ɔ	ɔ	vokal bulat dekat-tengah belakang	sen o	o
ɒ	O	vokal bulat punggung tengah terbuka	o tto	O
saya	saya	tutup vokal depan yang tidak dibulatkan	saya mpero	saya

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
u	u	tutup kembali vokal bulat	kamu tidak	u

Simbol Tambahan

4	"	stress primer	Ala ba ma	
,	%	stress sekunder	Sebuah labama	
.	.	batas suku kata	A.la.ba.ma	

Jepang (Ja-jp)

Amazon Polly mendukung alfabet Pengucapan Kana dan Yomigana untuk bahasa Jepang. Untuk membuat Amazon Polly menggunakan pengucapan fonetik dengan alfabet ini, gunakan atribut fonem. `alphabet="x-amazon-phonetic standard used"`

- `x-amazon-pron-kana`— menunjukkan bahwa Pengucapan Kana digunakan. Pengucapan Kana adalah karakter Katakana khusus yang digunakan untuk transkripsi fonetik dan dapat mengkodekan aksen nada.
- `x-amazon-yomigana`— menunjukkan bahwa Yomigana digunakan. Yomigana dapat berupa huruf Katakana, Hiragana, dan Latin konvensional yang ditafsirkan sebagai romanisasi hepburn.

Contoh berikut menunjukkan bagaimana ini digunakan:

Pengucapan Kana

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

Yomigana

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

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

Tabel berikut mencantumkan fonem Alfabet Fonetik Internasional (IPA), simbol Alfabet Fonetik Metode Penilaian Pidato Diperpanjang (X-SAMPA), dan viseme yang sesuai untuk suara Jepang yang didukung oleh Amazon Polly.

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Konsonan				
r	4	tutup alveolar	, r	t
ʔ	ʔ	berhenti glotal	, atsu ʔ	
b	b	plosif bilabial bersuara	, b uyou	p
β	B	frikatif bilabial bersuara	β, v inteeji	B
c	c	plosif palatal tanpa suara	, k i ky ou	k
ç	C	frikatif palatal tanpa suara	, h ito	k
d	d	plosif alveolar bersuara	, d akuten	t
ɟ	dz\	africate alveolo-palatal bersuara	, j un	J
g	g	plosif velar bersuara	, g ohan	k
-h	-h	frikatif glottal tanpa suara	, h pada	k
j	j	aproksiman palatal	, y ane	saya

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ɲ	ɲ\	plosif palatal bersuara	, gy ou g i	J
k	k	plosif velar tanpa suara	, k anji	k
ɺ	ɺ\	flap lateral alveolar	, tsu r i	r
& # x020j	ɺ j	flap lateral alveolar, aproksiman palatal	, ry uukou	r
m	m	hidung bilabial	, m eshi	p
n	n	hidung alveolar	, n eko	t
ɲ	ɲ	hidung palatal	, n ippon	J
ɴ	ɴ\	hidung uvular	, ka n	k
p	p	plosif bilabial tak bersuara	ン, p an	p
ɸ	p\	frikatif bilabial tak bersuara	, h uku	f
detik	detik	frikatif alveolar tak bersuara	, s ou	detik
ɕ	s\	frikatif alveolo-palatal tanpa suara	, sh okan	J
t	t	plosif alveolar tak bersuara	, t egami	t
ㄸ	ts	africate alveolar tak bersuara	, ts uri	detik

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ㄸ	ts\	africate alveolo-p alatal tanpa suara	, ki ch i	J
w	w	pendekatan labial- velar	, den w a	u
z	z	frikatif alveolar bersuara	, z ashiki	detik
Vokal				
ä:	a:_”	vokal panjang terbuka tengah yang tidak dibulatkan	, h aa ri	a
ä	a_”	buka vokal pusat yang tidak dibulatkan	, k a n a	a
e:	e:_o	vokal panjang tengah depan yang tidak dibulatkan	, gakus ei	@
e	e_o	vokal tengah depan yang tidak dibulatkan	, r e ki	@
saya	saya	tutup vokal depan yang tidak dibulatkan	, k i	saya
i:	i:	vokal panjang dekat depan yang tidak dibulatkan	, sh ii ka	saya

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
u	M	tutup kembali vokal yang tidak dibulatkan	, u n	saya
:	M:	vokal panjang dekat kembali yang tidak dibulatkan	, sh uu kyou	saya
o:	o: _o	vokal bulat tengah punggung panjang	, k oo doku	o
o	o_o	vokal bulat punggung tengah	, d o kusha	o

Korea (KO-kr)

Tabel berikut mencantumkan fonem Alfabet Fonetik Internasional (IPA), simbol Alfabet Fonetik Metode Penilaian Pidato Diperpanjang (X-SAMPA), dan viseme yang sesuai untuk suara Korea yang didukung oleh Amazon Polly.

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Konsonan				
k	k	plosif velar tanpa suara	, [g] ang	k
k#	k_t	plosif velar tanpa suara yang kuat	, [kk] e	k
n	n	hidung alveolar	, [n] am	t
t	t	plosif alveolar tak bersuara	, [d] o	t

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
t#	t_t	plosif alveolar tak bersuara yang kuat	, [tt] e	t
r	4	tutup alveolar	, di [r] ang	t
l	l	pendekatan lateral alveolar	, lakukan [l]	t
m	m	hidung bilabial	, [m] u	p
p	p	plosif bilabial tak bersuara	, [b] om	p
p#	p_t	plosif bilabial tak bersuara yang kuat	, [pp] eol	p
s	s	frikatif alveolar tak bersuara	, [s] e	s
s#	s_t	frikatif alveolar tak bersuara yang kuat	, [ss] i	s
ŋ	N	hidung velar	, ba [ng]	k
ㄷ	ts\	africate alveolo-p alatal tanpa suara	, [j] o	J
ㄷ#	ts_t	africate alveolo-p alatal tanpa suara yang kuat	, [jj] i	J
ㄷ ^h	ts_h	africate alveolo-p alatal tanpa suara aspirasi	, [ch] a	J
k ^h	k_h	aspirasi velar plosif tanpa suara	, [k] o	k

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
t ^h	t_h	plosif alveolar tak bersuara yang disedot	, [t] ong	t
p ^h	p_h	plosif bilabial tanpa suara yang disedot	, [p] e	p
h	h	frikatif glottal tanpa suara	, [h] im	k
j	j	aproksiman palatal	, [y] ang	i
w	w	pendekatan labial-velar	, [w] ang	u
ɯ	M\	sekitar velar>	1, [wɯ] i	i
Vokal				
a	a	buka vokal depan yang tidak dibulatkan	, b [a] b	a
ʌ	V	vokal terbuka tengah belakang yang tidak dibulatkan	, j [eo] ng	E
ɛ	E	vokal terbuka tengah depan yang tidak dibulatkan	, b [e]	E
o	o	vokal bulat dekat-tengah belakang	, n [o]	o
u	u	tutup kembali vokal bulat	, d [u] l	u

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ʊ	M	tutup kembali vokal yang tidak dibulatkan	, [eu] n	i
i	i	tutup vokal depan yang tidak dibulatkan	, k [i] m	i

Norwegia (NB-No)

Bagan berikut mencantumkan set lengkap fonem Alfabet Fonetik Internasional (IPA) dan simbol Alfabet Fonetik Metode Penilaian Pidato Diperpanjang (X-SAMPA) serta viseme yang sesuai yang didukung oleh Amazon Polly untuk suara bahasa Norwegia.

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Konsonan				
r	4	tutup alveolar	p r øv	t
b	b	plosif bilabial bersuara	bb	p
ç	C	frikatif palatal tanpa suara	k ino	k
d	d	plosif alveolar bersuara	la dd	t
ɖ	d`	plosif retroflex bersuara	ve rd saya	t
f	f	frikatif labiodental tanpa suara	f ot	f

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
g	g	plosif velar bersuara	ta gg	k
-h	-h	frikatif glottal tanpa suara	h	k
j	j	aproksiman palatal	g i	saya
k	k	plosif velar tanpa suara	ta kk	k
l	l	aproksiman lateral alveolar	fa ll, ba ll	t
ɫ	l̥	pendekatan lateral retroflex	æ rl ig	t
m	m	hidung bilabial	la m	p
n	n	hidung alveolar	va nn	t
ŋ	n̥	retroflex hidung	ga rn	t
ŋ	T	hidung velar	sa ng	k
p	p	plosif bilabial tak bersuara	ho pp	p
detik	detik	frikatif alveolar tak bersuara	la ss	detik
ʂ	s̥	frikatif retroflex tanpa suara	å rs	D
ʃ	D	frikatif postalveolar tanpa suara	s kyt	D

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
t	t	plosif alveolar tak bersuara	la t	t
ʈ	t̚	retroflex plosif tanpa suara	ha rdt	t
ʋ	\v	aproksiman labiodental	v di	f
w	w	pendekatan labial-velar	w sakit	x
Vokal				
ø:	2:	vokal bulat depan dekat-tengah panjang	s ø t	o
œ	9	vokal bulat depan terbuka	s ø tt	O
ə	@	vokal tengah tengah	ap e	@
æ:	{:	vokal depan yang tidak dibulatkan yang panjang hampir terbuka	v v	a
ʊ	}	tutup vokal bulat tengah	l u nd	u
ɪ:	}::	vokal bulat tengah yang panjang	l u n	u

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
æ	{	vokal depan yang tidak dibulatkan hampir terbuka	v æ rt	a
ɑ	A	buka kembali vokal yang tidak dibulatkan	h ke tt	a
ɑ:	A:	vokal panjang terbuka belakang yang tidak dibulatkan	h a t	a
e:	e:	vokal panjang dekat-tengah depan yang tidak dibulatkan	s e n	e
ɛ	E	vokal terbuka tengah depan yang tidak dibulatkan	s e nd	E
i:	i:	vokal panjang dekat depan yang tidak dibulatkan	v i n	saya
ɪ	ɪ	vokal tidak bulat dekat-dekat-dekat-depan	v i nd	saya
o:	o:	vokal bulat panjang dekat-tengah belakang	v t	o
ɔ	O	vokal bulat punggung tengah terbuka	v å tt	O

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
u:	u:	vokal bulat panjang dekat	b o k	u
ʊ	U	vokal bulat dekat-dekat belakang	b u k k	u
y:	y:	vokal bulat depan dekat yang panjang	l y n	u
ɣ	Y	vokal bulat dekat-dekat-dekat-depan	l y n n e	u
Simbol Tambahan				
'	"	stress primer	Ala ba ma	
,	%	stress sekunder	Sebuah labama	
.	.	batas suku kata	A.la.ba.ma	

Polandia (PL-PL)

Tabel berikut mencantumkan fonem Alfabet Fonetik Internasional (IPA), simbol Alfabet Fonetik Metode Penilaian Pidato Diperpanjang (X-SAMPA), dan viseme yang sesuai untuk suara Polandia yang didukung oleh Amazon Polly.

Tabel Fonem/Viseme

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Konsonan				
b	b	plosif bilabial bersuara	b obas, b elka	p

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
d	d	plosif alveolar bersuara	d ar, d o	t
ɖ	dz	africate alveolar bersuara	dz menang, wi dz owie	detik
ɖ̌	dz\	africate alveolo-palatal bersuara	dź więk	J
ɖ̎	dz`	africate retroflex bersuara	dà em, dà ungla	D
f	f	frikatif labiodental tanpa suara	f urtkà, f ilm	f
g	g	plosif velar bersuara	g azeta, wa g a	k
-h	-h	frikatif glottal tanpa suara	ch leb, h andel	k
j	j	pendekatan palatal	j ak, ma j a	saya
k	k	plosif velar tanpa suara	k ura, kuda k	k
l	l	pendekatan lateral alveolar	l ipa, a l icja	t
m	m	hidung bilabial	m atka, m olo	p
n	n	hidung alveolar	n orka	t
ɲ	J	hidung palatal	ko ń, toru ń	J
p	p	plosif bilabial tak bersuara	p ora, sto p	p

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
r	r	getar alveolar	r ok, pa r k	r
detik	detik	frikatif alveolar tak bersuara	s um, pa s	detik
ʃ	s\	frikatif alveolo-p alatal tanpa suara	ś ruba, ś nieg	J
ʂ	s`	frikatif retroflex tanpa suara	sz um, ma sz	D
t	t	plosif alveolar tak bersuara	t ok, s t ół	t
ts	ts	africate alveolar tanpa suara	c ar, c o	detik
t͡	ts\	africate alveolo-p alatal tanpa suara	ć ma, mie ć	J
t͡	ts`	africate retroflex tanpa suara	cz sebagai, ra cz ej	D
v	v	frikatif labiodental bersuara	w orek, saya w a	f
w	w	pendekatan labial-velar	ł aska, ma ł o	u
z	z	frikatif alveolar bersuara	z ero	detik
ʒ	z\	frikatif alveolo-p alatal bersuara	ż rebi, bieli ż nie	J
ʐ	z`	frikatif retroflex bersuara	ar, v ona	D

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Vokal				
a	a	buka vokal depan yang tidak dibulatkan	j	a
ɛ	E	vokal terbuka tengah depan yang tidak dibulatkan	e cho	E
ɛ	E ~	vokal terbuka bagian depan terbuka hidung yang tidak dibulatkan	w ze	E
saya	saya	tutup vokal depan yang tidak dibulatkan	i le	saya
ɔ	O	vokal bulat punggung tengah terbuka	o czy	O
ɔ	O ~	vokal bulat punggung tengah terbuka hidung	w ą	O
u	u	tutup kembali vokal bulat	u czta	u
i	1	tutup vokal tengah yang tidak dibulatkan	b y k	saya
Simbol Tambahan				

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
4	"	stress primer	Ala ba ma	
,	%	stress sekunder	Sebuah labama	
.	.	batas suku kata	A.la.ba.ma	

Portugis (Pt-PT)

Tabel berikut mencantumkan fonem Alfabet Fonetik Internasional (IPA), simbol Alfabet Fonetik Metode Penilaian Pidato Diperpanjang (X-SAMPA), dan viseme yang sesuai untuk suara Portugis yang didukung oleh Amazon Polly.

Tabel Fonem/Viseme

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Konsonan				
r	4	tutup alveolar	pi r a	t
b	b	plosif bilabial bersuara	d ato	p
d	d	plosif alveolar bersuara	d ato	t
f	f	frikatif labiodental tanpa suara	f acto	f
g	g	plosif velar bersuara	g ato	k
j	j	aproksiman palatal	paragua y	saya
k	k	plosif velar tanpa suara	c acto	k

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
l	l	pendekatan lateral alveolar	ga l o	t
ʎ	L	aproksiman lateral palatal	ga lh o	J
m	m	hidung bilabial	m ato	p
n	n	hidung alveolar	n ato	t
ɲ	J	hidung palatal	pi nh	J
p	p	plosif bilabial tak bersuara	p ato	p
ʀ	R\	getar uvular	ba rr oso	k
detik	detik	frikatif alveolar tak bersuara	s aca	detik
ʃ	D	frikatif postalveolar tanpa suara	ch ato	D
t	t	plosif alveolar tak bersuara	t acto	t
v	v	frikatif labiodental bersuara	v aca	f
w	w	pendekatan labial-velar	ma u	u
z	z	frikatif alveolar bersuara	z aca	detik
ʒ	Z	frikatif postalveolar bersuara	j acto	D

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Vokal				
a	a	buka vokal depan yang tidak dibulatkan	p rto	a
a	a ~	vokal depan terbuka hidung yang tidak dibulatkan	p e ga	a
e	e	vokal dekat-tengah depan yang tidak dibulatkan	p e ga	e
e	e ~	vokal hidung dekat- tengah depan yang tidak dibulatkan	mov e m	e
ɛ	E	vokal terbuka tengah depan yang tidak dibulatkan	caf é	E
saya	saya	tutup vokal depan yang tidak dibulatkan	l saya ngueta	saya
saya	saya~	hidung dekat depan vokal yang tidak dibulatkan	c i nto	saya
o	o	vokal bulat dekat- tengah belakang	p o der	o
o	o ~	vokal bulat hidung dekat-tengah belakang	c o mpra	o

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ɔ	O	vokal bulat punggung tengah terbuka	dipan ó	O
u	u	tutup kembali vokal bulat	f u saya	u
u	u ~	hidung tutup belakang vokal bulat	s u nto	u
Simbol Tambahan				
'	"	stress primer	Ala ba ma	
,	%	stress sekunder	Sebuah labama	
.	.	batas suku kata	A.la.ba.ma	

Portugis (Brasil) (pt-BR)

Tabel berikut mencantumkan fonem Alfabet Fonetik Internasional (IPA), simbol Alfabet Fonetik Metode Penilaian Pidato Diperpanjang (X-SAMPA), dan viseme yang sesuai untuk suara Portugis Brasil yang didukung oleh Amazon Polly.

Tabel Fonem/Viseme

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Konsonan				
r	4	tutup alveolar	pi r a	t
b	b	plosif bilabial bersuara	b ato	p

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
d	d	plosif alveolar bersuara	d ato	t
ɗ	dZ	afrikate postalveolar bersuara	ida de	D
f	f	frikatif labiodental tanpa suara	f acto	f
g	g	plosif velar bersuara	g ato	k
j	j	pendekatan palatal	paragua y	saya
k	k	plosif velar tanpa suara	c acto	k
l	l	pendekatan lateral alveolar	ga l o	t
ʎ	L	aproksiman lateral palatal	ga lh o	J
m	m	hidung bilabial	m ato	p
n	n	hidung alveolar	n ato	t
ɲ	J	hidung palatal	pi nh a	J
p	p	plosif bilabial tak bersuara	p ato	p
detik	detik	frikatif alveolar tak bersuara	s aca	detik
ʃ	D	frikatif postalveolar tanpa suara	ch ato	D

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
t	t	plosif alveolar tak bersuara	t acto	t
ʈ	tS	afrikate postalveolar tanpa suara	noi te	D
v	v	frikatif labiodental bersuara	v aca	f
w	w	pendekatan labial-velar	ma u	u
χ	X	frikatif uvular tak bersuara	ca rr o	k
z	z	frikatif alveolar bersuara	z aca	detik
ʒ	Z	frikatif postalveolar bersuara	j acto	D
Vokal				
a	a	buka vokal depan yang tidak dibulatkan	p rto	a
ɑ	ɑ ~	vokal depan terbuka hidung yang tidak dibulatkan	pena sebuah mos	ɑ
e	e	vokal dekat-tengah depan yang tidak dibulatkan	p e ga	e

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
e	e ~	vokal hidung dekat-tengah depan yang tidak dibulatkan	mov e m	e
ɛ	E	vokal terbuka tengah depan yang tidak dibulatkan	caf é	E
saya	saya	tutup vokal depan yang tidak dibulatkan	l saya ngueta	saya
saya	saya~	hidung dekat depan vokal yang tidak dibulatkan	c i nto	saya
o	o	vokal bulat dekat-tengah belakang	p o der	o
o	o ~	vokal bulat hidung dekat-tengah belakang	c o mpra	o
ɔ	O	vokal bulat punggung tengah terbuka	dipan ó	O
u	u	tutup kembali vokal bulat	f u saya	u
u	u ~	hidung tutup belakang vokal bulat	s u nto	u
Simbol Tambahan				
4	"	stress primer	Ala ba ma	

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
,	%	stress sekunder	Sebuah labama	
.	.	batas suku kata	A.la.ba.ma	

Rumania (Ro-ro)

Tabel berikut mencantumkan fonem Alfabet Fonetik Internasional (IPA), simbol Alfabet Fonetik Metode Penilaian Pidato Diperpanjang (X-SAMPA), dan viseme yang sesuai untuk suara Rumania yang didukung oleh Amazon Polly.

Tabel Fonem/Viseme

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Konsonan				
b	b	plosif bilabial bersuara	b	p
d	d	plosif alveolar bersuara	d upă	t
ɖ	dZ	africate postalveolar bersuara	ge orge	D
f	f	frikatif labiodental tanpa suara	a f acere	f
g	g	plosif velar bersuara	a g riș	k
-h	-h	frikatif glottal tanpa suara	h arpa	k
j	j	aproksiman palatal	ba i e	saya

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
k	k	plosif velar tanpa suara	c oș	k
l	l	pendekatan lateral alveolar	l ampa	t
m	m	hidung bilabial	m ama	p
n	n	hidung alveolar	n atau	t
p	p	plosif bilabial tak bersuara	p ilă	p
r	r	getar alveolar	r ampa	r
detik	detik	frikatif alveolar tak bersuara	s lain	detik
ʃ	D	frikatif postalveolar tanpa suara	ma # ină	D
t	t	plosif alveolar tak bersuara	t ata	t
ts	ts	africate alveolar tak bersuara	ț ară	detik
t̚	tS	africate postalveolar tanpa suara	ce ai	D
v	v	frikatif labiodental bersuara	v negara	f
w	w	pendekatan labial-velar	bea u	u
z	z	frikatif alveolar bersuara	mo z ol	detik

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ʒ	Z	frikatif postalveolar bersuara	j oacă	D
Vokal				
ə	@	vokal tengah tengah	Bab	@
a	a	buka vokal depan yang tidak dibulatkan	c a sa	a
e	e	vokal dekat-tengah depan yang tidak dibulatkan	e lan	e
e	e_^	vokal tidak bulat dekat-tengah depan non-suku kata	b e au	e
saya	saya	tutup vokal depan yang tidak dibulatkan	m i e	saya
o	o	dekat-tengah punggung bulat vo	o ră	o
oa	o_^a	diftong	oa re	o
u	u	tutup kembali vokal bulat	u nde	u
ɨ	1	tutup vokal tengah yang tidak dibulatkan	Rom â nia	saya

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Simbol Tambahan				
'	"	stress primer	Ala ba ma	
,	%	stress sekunder	Sebuah labama	
.	.	batas suku kata	A.la.ba.ma	

Rusia (Ru-ru)

Tabel berikut mencantumkan fonem Alfabet Fonetik Internasional (IPA), simbol Alfabet Fonetik Metode Penilaian Pidato Diperpanjang (X-SAMPA), dan viseme yang sesuai untuk suara Rusia yang didukung oleh Amazon Polly.

Tabel Fonem/Viseme

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Konsonan				
b	b	plosif bilabial bersuara	в оро	p
b	b'	plosif bilabial bersuara palatalis asi	po	p
d	d	plosif alveolar bersuara	в ом	t
d	d'	plosif alveolar bersuara palatalis asi	(с) я	t
f	f	frikatif labiodental tanpa suara	аr	f

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
f	f'	frikatif labiodental tanpa suara palatalisasi	р евраль	f
g	g	plosif velar bersuara	но г а	k
g	g'	plosif velar bersuara palatalisasi	г ерос	k
j	j	aproksiman palatal	дина на н, я иц	saya
k	k	plosif velar tanpa suara	К оо	k
k	k'	plosif velar tanpa suara yang dipalatalisasi	К ино	k
l	l	aproksiman lateral alveolar	р амта	t
l	l'	aproksiman lateral alveolar palatalisasi	х ес	t
m	m	hidung bilabial	м ама	p
m	m'	hidung bilabial palatalisasi		p
n	n	hidung alveolar	н ос	t
n	n'	nasal alveolar palatalisasi	н я н	t

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
p	p	plosif bilabial tak bersuara	р ата	p
p	p'	plosif bilabial tanpa suara yang dipalatalisasi	р еро	p
r	r	getar alveolar	р она	r
r	r'	getar alveolar palatalisasi	р мка	r
detik	detik	frikatif alveolar tak bersuara	с ыр	detik
s	s'	frikatif alveolar tanpa suara palatalisasi	с ердце, ру съ	detik
:	s\:	frikatif alveolo-p alatal tanpa suara panjang	ека	J
ʂ	s`	frikatif retroflex tanpa suara	(i)	D
t	t	plosif alveolar tak bersuara	о оскa	t
t	t'	plosif alveolar tak bersuara palatalisasi	т ё а я	t
ʦ	ts	africate alveolar tak bersuara	аръ	detik

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ʈ	ts\	africate alveolo-palatal tanpa suara	ас	J
v	v	frikatif labiodental bersuara	в op	f
v	v'	frikatif labiodental bersuara palatalisasi	в ерФь	f
x	x	frikatif velar tanpa suara	х op	k
x	x'	frikatif velar tanpa suara yang dipalatalisasi	х имия	k
z	z	frikatif alveolar bersuara	(i)	detik
z	z'	frikatif alveolar bersuara palatalisasi	х има	detik
:	z\:	frikatif alveolo-palatal bersuara panjang	уе бaть	J
ʐ	z`	frikatif retroflex bersuara	ена	D
Vokal				
ə	@	vokal tengah tengah	канарейца	@

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
a	a	buka vokal depan yang tidak dibulatkan	дв а, я бооко	a
e	e	vokal dekat-tengah depan yang tidak dibulatkan	р е съ	e
ɛ	E	vokal terbuka tengah depan yang tidak dibulatkan	то	E
saya	saya	tutup vokal depan yang tidak dibulatkan	ов и н, е тыре	saya
o	o	vokal bulat dekat- tengah belakang	К о т	o
u	u	tutup kembali vokal bulat	т у, въГ а	u
i	1	tutup vokal tengah yang tidak dibulatkan	т ы уь	saya

Spanyol (ES-es)

Tabel berikut mencantumkan fonem Alfabet Fonetik Internasional (IPA), simbol Alfabet Fonetik Metode Penilaian Pidato Diperpanjang (X-SAMPA), dan viseme yang sesuai untuk suara Spanyol yang didukung oleh Amazon Polly.

Tabel Fonem/Viseme

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Konsonan				
r	4	tutup alveolar	pe r o, b r avo, amo r, e r no	t
b	b	plosif bilabial bersuara	b e stia	p
β	B	frikatif bilabial bersuara	menjadi b é	B
d	d	plosif alveolar bersuara	cuan d o	t
ð	D	frikatif gigi bersuara	ar d er	T
f	f	frikatif labiodental tanpa suara	f ase, ca f é	f
g	g	plosif velar bersuara	g ato, len g ua, g uerra	k
ɣ	G	frikatif velar bersuara	tri g o, Ar g os	k
j	j	aproksiman palatal	hac i a, t i erra, rad i o, v i uda	saya
ɲ	j\	frikatif palatal bersuara	enh i elar, sa y o, di y ectado, des y erba	J
k	k	plosif velar tanpa suara	c aña, la c a, q uisimos	k

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
l	l	pendekatan lateral alveolar	l ino, ca l atau, principa l	t
ʎ	L	aproksiman lateral palatal	Akan ave, po ll o	J
m	m	hidung bilabial	m adre, co m er, a n fibio	p
n	n	hidung alveolar	n ido, n illo, si n	t
ɲ	J	hidung palatal	caba ñ a, ñ oquis	J
ŋ	T	hidung velar	ci n co, ve n ga	k
p	p	plosif bilabial tak bersuara	p ozo, ke p o	p
r	r	getar alveolar	pe rr o, en r achado	r
detik	detik	frikatif alveolar tak bersuara	s aco, ca s a, puerta s	detik
t	t	plosif alveolar tak bersuara	t amiz, á t omo	t
ʈ	tS	africate postalveolar tanpa suara	ch ubasco	D
θ	T	frikatif gigi tanpa suara	cere z a, z orro, la c ero, pa z	T
w	w	pendekatan labial-velar	f u ego, f u imos, c u ota, c u adro	u
x	x	frikatif velar tanpa suara	j amon, g eneral, su j e, relo j	k

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
z	z	frikatif alveolar bersuara	ra s pergi, mi s mo	detik
Vokal				
a	a	buka vokal depan yang tidak dibulatkan	t a nque	a
e	e	vokal dekat-tengah depan yang tidak dibulatkan	p e jadi	e
saya	saya	tutup vokal depan yang tidak dibulatkan	c i nco	saya
o	o	vokal bulat dekat- tengah belakang	b o sque	o
u	u	vokal dekat-tengah depan yang tidak dibulatkan	p u blicar	u
Simbol Tambahan				
4	"	stress primer	Ala ba ma	
,	%	stress sekunder	Sebuah labama	
.	.	batas suku kata	A.la.ba.ma	

Spanyol (Meksiko) (es-Mx)

Tabel berikut mencantumkan fonem Alfabet Fonetik Internasional (IPA), simbol Alfabet Fonetik Metode Penilaian Pidato Diperpanjang (X-SAMPA), dan viseme yang sesuai untuk suara Spanyol Meksiko yang didukung oleh Amazon Polly.

Tabel Fonem/Viseme

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Konsonan				
r	4	tutup alveolar	pe r o, b r avo, amo r, e r no	t
b	b	plosif bilabial bersuara	b estia	p
β	B	frikatif bilabial bersuara	menjadi b é	B
d	d	plosif alveolar bersuara	cuan d o	t
ð	D	frikatif gigi bersuara	ar d er	T
f	f	frikatif labiodental tanpa suara	f ase, ca f é	f
g	g	plosif velar bersuara	g ato, len g ua, g uerra	k
ɣ	G	frikatif velar bersuara	tri g o, Ar g os	k
j	j	aproksiman palatal	hac i a, t i erra, rad i o, v i uda	saya
ɲ	j\	frikatif palatal bersuara	enh i elar, sa y o, di y ectado, des y erba	J
k	k	plosif velar tanpa suara	c aña, la c a, q uisimos	k

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
l	l	pendekatan alveolar lateral	lino, ca l atau, principa l	t
m	m	hidung bilabial	m adre, co m er, a n fibio	p
n	n	hidung alveolar	n ido, n illo, si n	t
ɲ	J	hidung palatal	caba ñ a, ñ oquis	J
ŋ	T	hidung velar	a n gosto, i n creíble	k
p	p	plosif bilabial tak bersuara	p ozo, ke p o	p
r	r	getar alveolar	pe rr o, en r achado	r
detik	detik	frikatif alveolar tak bersuara	s aco, ca s a, puerta s	detik
ʃ	D	frikatif postalveolar tanpa suara	sh ow, fla sh	D
t	t	plosif alveolar tak bersuara	t amiz, á t omo	t
ʈ	tS	africate postalveolar tanpa suara	ch ubasco	D
w	w	pendekatan labial-velar	f u ego, f u imos, c u ota, c u adro	u
x	x	frikatif velar tanpa suara	j amon, g eneral, kacang j e, relo j	k

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
z	z	frikatif alveolar bersuara	ra s pergi, mi s mo	detik

Vokal

a	a	vokal pusat terbuka yang tidak dibulatkan	t a nque	a
e	e	vokal dekat-tengah depan yang tidak dibulatkan	p e jadi	e
saya	saya	tutup vokal depan yang tidak dibulatkan	c i nco	saya
o	o	vokal bulat dekat-tengah belakang	b o sque	o
u	u	tutup kembali vokal bulat	p u blicar	u

Simbol Tambahan

4	"	stress primer	Ala ba ma	
,	%	stress sekunder	Sebuah labama	
.	.	batas suku kata	A.la.ba.ma	

Spanyol (AS) (es-AS)

Tabel berikut mencantumkan fonem Alfabet Fonetik Internasional (IPA), simbol Alfabet Fonetik Metode Penilaian Pidato Diperpanjang (X-SAMPA), dan viseme yang sesuai untuk suara Spanyol AS yang didukung oleh Amazon Polly.

Tabel Fonem/Viseme

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Konsonan				
r	4	tutup alveolar	pe r o, b r avo, amo r, e r no	t
b	b	plosif bilabial bersuara	b estia	p
β	B	frikatif bilabial bersuara	menjadi b é	B
d	d	plosif alveolar bersuara	cuan d o	t
ð	D	frikatif gigi bersuara	ar d er	T
f	f	frikatif labiodental tanpa suara	f ase, ca f é	f
g	g	plosif velar bersuara	g ato, len g ua, g uerra	k
ɣ	G	frikatif velar bersuara	tri g o, Ar g os	k
j	j	aproksiman palatal	hac i a, t i erra, rad i o, v i uda	saya
ɲ	j\	frikatif palatal bersuara	enh i elar, sa y o, di y ectado, des y erba	J
k	k	plosif velar tanpa suara	c aña, la c a, q uisimos	k

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
l	l	pendekatan alveolar lateral	l ino, ca l atau, principa l	t
m	m	hidung bilabial	m adre, co m er, a n fibio	p
n	n	hidung alveolar	n ido, a n illo, si n	t
ɲ	J	hidung palatal	caba ñ a, ñ oquis	J
ŋ	T	hidung velar	a n gosto, i n creíble	k
p	p	plosif bilabial tak bersuara	p ozo, ke p o	p
r	r	getar alveolar	pe rr o, en r achado	r
detik	detik	frikatif alveolar tak bersuara	s aco, ca s a, puerta s	detik
ʃ	D	frikatif postalveolar tanpa suara	sh ow, fla sh	D
t	t	plosif alveolar tak bersuara	t amiz, á t omo	t
ʈ	tS	africate postalveolar tanpa suara	ch ubasco	D
w	w	pendekatan labial-velar	f u ego, f u imos, c u ota, c u adro	u
x	x	frikatif velar tanpa suara	j amon, g eneral, kacang j e, relo j	k

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
z	z	frikatif alveolar bersuara	ra s pergi, mi s mo	detik

Vokal

a	a	vokal pusat terbuka yang tidak dibulatkan	t a nque	a
e	e	vokal dekat-tengah depan yang tidak dibulatkan	p e jadi	e
saya	saya	tutup vokal depan yang tidak dibulatkan	c i nco	saya
o	o	vokal bulat dekat-tengah belakang	b o sque	o
u	u	tutup kembali vokal bulat	p u blicar	u

Simbol Tambahan

'	"	stress primer	Ala ba ma	
,	%	stress sekunder	Sebuah labama	
.	.	batas suku kata	A.la.ba.ma	

Swedia (SV-se)

Tabel berikut mencantumkan fonem Alfabet Fonetik Internasional (IPA), simbol Alfabet Fonetik Metode Penilaian Pidato Diperpanjang (X-SAMPA), dan viseme yang sesuai untuk suara Swedia yang didukung oleh Amazon Polly.

Tabel Fonem/Viseme

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Konsonan				
b	b	plosif bilabial bersuara	b	p
d	d	plosif alveolar bersuara	d al	t
ɖ	d̪	plosif retroflex bersuara	bo rd	t
f	f	frikatif labiodental tanpa suara	f	f
g	g	plosif velar bersuara	g ås	k
-h	-h	frikatif glottal tanpa suara	h al	k
j	j	aproksiman palatal	j ag	saya
k	k	plosif velar tanpa suara	k al	k
l	l	pendekatan lateral alveolar	l ös	t
ɭ	l̪	pendekatan lateral retroflex	ke rl ig	t
m	m	hidung bilabial	m il	p
n	n	hidung alveolar	n tahun	t
ɳ	n̪	retroflex hidung	ba rn	t

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ŋ	T	hidung velar	ri ng	k
p	p	plosif bilabial tak bersuara	p il	p
r	r	getar alveolar	r adalah	r
detik	detik	frikatif alveolar tak bersuara	s il	detik
ç	s\	frikatif alveolo-p alatal tanpa suara	tj ock	J
ʃ	s`	frikatif retroflex tanpa suara	fo rs, sch lager	D
t	t	plosif alveolar tak bersuara	t al	t
ʈ	t`	retroflex plosif tanpa suara	hjo rt	t
v	v	frikatif labiodental bersuara	v tahun	f
w	w	pendekatan labial-velar	a u la, udara w ays	u
ɣ	x\	frikatif palatal-velar tanpa suara	sj uk	k
Vokal				
ø	2	vokal bulat dekat-tengah depan	f ö ll, f ö rr	o

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ø	2:	vokal bulat depan dekat-tengah panjang	f ö l, n ö t, f ö r	o
ə	8	vokal bulat tengah dekat-tengah	b u ss, f u ll	o
ə	@	vokal tengah tengah	pojk e n	@
:	}::	vokal bulat tengah yang panjang	h u s, f u l	u
a	a	buka vokal depan yang tidak dibulatkan	h ll, m a tt	a
æ	{	vokal depan yang tidak dibulatkan hampir terbuka	h e rr	a
ɑ:	A:	vokal panjang terbuka belakang yang tidak dibulatkan	h a l, m a t	a
e:	e:	vokal panjang dekat-tengah depan yang tidak dibulatkan	v e t, h e l	e
ɛ	E	vokal terbuka tengah depan yang tidak dibulatkan	v e tt, r ä tt, h e tta, h ä ll	E

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ɛ:	E:	vokal panjang terbuka tengah depan yang tidak dibulatkan	s ä l, h ä l, h ä r	E:
i:	i:	vokal panjang dekat depan yang tidak dibulatkan	v i t, s i l	i:
ɪ	ɪ	vokal tidak bulat dekat-dekat-dekat- depan	v i t t, s i l l	saya
o:	o:	vokal bulat panjang dekat-tengah belakang	h å l, m å l	o
ɔ	O	vokal bulat punggung tengah terbuka	h ll, m o ll	O
u:	u:	vokal bulat dekat belakang panjang	s o l, b o t	u
ʊ	U	vokal bulat dekat- dekat belakang	b o t t	u
y	y	tutup vokal bulat depan	b y t t	u
y:	y:	vokal bulat depan dekat panjang	s y l, s y l	u
Simbol Tambahan				
'	"	stress primer	Ala ba ma	

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
,	%	stress sekunder	Sebuah labama	
.	.	batas suku kata	A.la.ba.ma	

Turki (Tr-tr)

Tabel berikut mencantumkan fonem Alfabet Fonetik Internasional (IPA), simbol Alfabet Fonetik Metode Penilaian Pidato Diperpanjang (X-SAMPA), dan viseme yang sesuai untuk suara Turki yang didukung oleh Amazon Polly.

Tabel Fonem/Viseme

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Konsonan				
r	4	tutup alveolar	du r um	t
ɾ	4_0_r	flap alveolar gesekan tanpa suara	bi r	t
ɹ	4_r	flap alveolar gesekan	r	t
b	b	plosif bilabial bersuara	r	p
c	c	plosif palatal tanpa suara	k edi	k
d	d	plosif alveolar bersuara	d e e	t
ɟ	dZ	africate postalveo lar bersuara	c saya	D

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
f	f	frikatif labiodental tanpa suara	f adalah	f
g	g	velar plosiv bersuara	g alibi	k
-h	-h	frikatif glottal tanpa suara	h asta	k
j	j	aproksiman palatal	y di	saya
ɟ	J\	plosif palatal bersuara	g enç	J
k	k	plosif velar tanpa suara	k il	k
l	l	aproksiman lateral alveolar	l l e	t
ɭ	5	aproksiman lateral alveolar velarisasi	l abirent	t
m	m	hidung bilabial	m aaş	p
n	n	hidung alveolar	a n ı	t
p	p	plosif bilabial tak bersuara	i p	p
detik	detik	frikatif alveolar tak bersuara	s es	detik
ʃ	D	frikatif postalveolar tanpa suara	a ş ı	D
t	t	plosif alveolar tak bersuara	ü t ü	t

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ʈ	tʂ	africate postalveolar tanpa suara	ç aba	D
v	v	frikatif labiodental bersuara	ek v ator, kah v eci, ak v aryum, adalah v eçli, teş v iki, cet v el	f
z	z	frikatif alveolar bersuara	v er	detik
ʒ	ʒ	frikatif postalveolar bersuara	a z ık	D
Vokal				
ø	2	vokal bulat dekat-tengah depan	g ö l	0
œ	9	vokal bulat depan terbuka	banliy ö	O
a	a	buka vokal depan yang tidak dibulatkan	k a l	a
a:	a:	vokal depan terbuka panjang yang tidak dibulatkan	d a v a cı	a
æ	{	vokal depan yang tidak dibulatkan hampir terbuka	özl e m, güv e nlik, gür e l, som e rsault	a

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
e	e	vokal dekat-tengah depan yang tidak dibulatkan	k e apa	e
ɛ	E	vokal terbuka tengah depan yang tidak dibulatkan	ded e	E
saya	saya	tutup vokal depan yang tidak dibulatkan	b i r	saya
i:	i:	vokal panjang dekat depan yang tidak dibulatkan	saya zah	saya
ɪ	ɪ	vokal tidak bulat dekat-dekat-dekat- depan	keç i	saya
ʊ	ʊ	tutup kembali vokal yang tidak dibulatkan	k i l	saya
o	o	vokal bulat dekat- tengah belakang	k o l	o
o:	o:	vokal bulat panjang dekat-tengah belakang	d o lar	o
u	u	tutup kembali vokal bulat	d u r u m	u
u:	u:	vokal bulat panjang dekat	r u bersenandung	u

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ʊ	U	vokal bulat dekat-dekat belakang	dol kamu	u
y	y	tutup vokal bulat depan	g ü venlik	u
ɤ	Y	vokal bulat dekat-dekat-depan	aş ı	u
Simbol Tambahan				
'	"	stress primer	Ala ba ma	
,	%	stress sekunder	Sebuah labama	
.	.	batas suku kata	A.la.ba.ma	

Welsh (cy-GB)

Tabel berikut mencantumkan fonem Alfabet Fonetik Internasional (IPA), simbol Alfabet Fonetik Metode Penilaian Pidato Diperpanjang (X-SAMPA), dan viseme yang sesuai untuk suara Welsh yang didukung oleh Amazon Polly.

Tabel Fonem/Viseme

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Konsonan				
b	b	plosif bilabial bersuara	b aban	p
d	d	plosif alveolar bersuara	d misalnya	t
ɖ	dZ	africate postalveolar bersuara	gare j	D

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ð	D	frikatif gigi bersuara	deu dd misalnya	T
f	f	frikatif labiodental tanpa suara	ff acs	f
g	g	plosif velar bersuara	g adael	k
-h	-h	frikatif glottal tanpa suara	h aearn	k
j	j	aproksiman palatal	cerdik i o	saya
k	k	plosif velar tanpa suara	c semut	k
l	l	aproksiman lateral alveolar	l	t
ɬ	K	frikatif lateral alveolar tak bersuara	akan	t
m	m	hidung bilabial	m ae	p
m	m_0	hidung bilabial tanpa suara	y mh id	p
n	n	hidung alveolar	n aw	t
n	n_0	hidung alveolar tanpa suara	sebuah nh awster	t
ŋ	T	hidung velar	argyfw	k
ŋ	N_0	hidung velar tanpa suara	henion	k

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
p	p	plosif bilabial tak bersuara	p ump	p
r	r	getar alveolar	rh oi	r
r	r_0	getar alveolar tak bersuara	ga r w	r
detik	detik	frikatif alveolar tak bersuara	s aith	detik
ʃ	D	frikatif postalveolar tanpa suara	tenda si	D
t	t	plosif alveolar tak bersuara	t egan	t
t͡ʃ	tS	afrikate postalveolar tanpa suara	cy ts ain	D
θ	T	frikatif gigi tanpa suara	aber th	T
v	v	frikatif labiodental bersuara	cakar f	f
w	w	pendekatan labial-velar	rhag w eld	u
χ	X	frikatif uvular tak bersuara	ch kita ch	k
z	z	frikatif alveolar bersuara	bantuan s	detik
ʒ	Z	frikatif postalveolar bersuara	rou ge	D

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Vokal				
ə	@	vokal tengah tengah	y chwaneg	@
a	a	buka vokal depan yang tidak dibulatkan	sebuah cen	a
ai	ai	diftong	d au	a
au	au	diftong	aw dur	a
ɑ:	A:	vokal panjang terbuka belakang yang tidak dibulatkan	m a b	a
ɑ:	A:1	diftong	ae lod	a
e:	e:	vokal panjang dekat-tengah depan yang tidak dibulatkan	p e th	e
ɛ	E	vokal terbuka tengah depan yang tidak dibulatkan	p e dwar	E
ɛi	Ei	diftong	b i c	E
saya:	saya:	vokal panjang dekat depan yang tidak dibulatkan	tr saya	saya

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ɪ	l	vokal tidak bulat dekat-dekat-dekat- depan	m saya liwn	saya
u	1u	diftong	unigr yw	saya
o:	o:	vokal bulat panjang dekat-tengah belakang	o ddi	o
ɔ	O	vokal bulat punggung tengah terbuka	o ddieithr	O
ɔi	Oi	diftong	tr oi	O
ɔu	Ou	diftong	r ow nd	O
u:	u:	vokal bulat panjang dekat	c w ch	u
ʊ	U	vokal bulat dekat- dekat belakang	ac w stig	u
ʊi	Ui	diftong	wy th	u
Simbol Tambahan				
4	"	stress primer	Ala ba ma	
,	%	stress sekunder	Sebuah labama	
.	.	batas suku kata	A.la.ba.ma	

Mesin suara Amazon Polly

Amazon Polly memiliki empat mesin suara yang mengubah teks input menjadi ucapan yang hidup. Ini termasuk: Generatif, Long-form, Neural, dan Standar. Untuk menggunakan suara Amazon Polly, pilih mesin dan operasi API sintesis ucapan. Kemudian berikan teks input untuk mesin untuk disintesis, dan pilih format output audio. Dengan masukan ini, Amazon Polly mensintesis teks yang disediakan ke dalam aliran audio ucapan berkualitas tinggi.

Bagian berikut mencakup detail tentang mesin suara yang ditawarkan oleh Amazon Polly.

Topik

- [Suara generatif](#)
- [Long-form suara](#)
- [Suara saraf](#)
- [Suara standar](#)
- [Memilih mesin suara](#)

Suara generatif

Mesin text-to-speech (TTS) generatif Amazon Polly menawarkan suara percakapan yang paling mirip manusia, terlibat secara emosional, dan adaptif yang tersedia untuk digunakan melalui konsol Amazon Polly.

Mesin Generatif adalah model Amazon Polly TTS terbesar hingga saat ini. Ini menyebarkan transformator miliaran parameter yang mengubah teks mentah menjadi kode ucapan, diikuti oleh dekoder berbasis konvolusi yang mengubah kode ucapan ini menjadi bentuk gelombang secara bertahap dan ramping. Metode ini menunjukkan kemampuan muncul Model Bahasa Besar (LLM) yang dilaporkan secara luas ketika dilatih tentang peningkatan volume data yang tersedia untuk umum dan kepemilikan yang terdiri dari berbagai suara, bahasa, dan gaya.

Mesin Generatif menciptakan ucapan sintetis yang terlibat secara emosional, tegas, dan sangat sehari-hari dengan cara yang sangat mirip dengan suara manusia. Anda dapat menggunakan suara-suara ini sebagai asisten pelanggan yang berpengetahuan luas, pelatih virtual, atau pengiklan dengan pidato sintetis yang hampir manusiawi.

Note

Teknologi canggih yang mendasari suara-suara ini termasuk dalam paradigma AI generatif untuk pemodelan bahasa dan suara. Efek samping dari teknologi ini adalah bahwa setiap pembaruan pada data pelatihan dan model dapat menghasilkan sedikit variasi pada cara suara terdengar, bahkan jika kualitas keseluruhannya meningkat dengan pembaruan model. Ini dapat berdampak pada kasus penggunaan dengan bagian konten berbeda yang disintesis dalam jangka waktu yang lama — misalnya, musim podcast.

Suara generatif yang tersedia

Amazon Polly saat ini menawarkan 43 suara dalam varian generatif.

	Bahasa	Kode bahasa	Name/ID	Gender
1	Inggris (Australia)	En-au	Olivia	Perempuan
2	Inggris (Inggris)	id - GB	Amy	Perempuan
			Brian	Laki-laki
3	Inggris (India)	En-in	Kajal	Perempuan
4	Inggris (Irlandia)	l-ie	Niamh	Perempuan
5	Inggris (Selandia Baru)	en-NZ	Aria	Perempuan
6	English (Singaporean)	En-SG	Melati	Perempuan
7	Inggris (Afrika Selatan)	En-za	Ayanda	Perempuan
8	Inggris (US)	en-US	Danielle	Perempuan
			Joanna	Perempuan

	Bahasa	Kode bahasa	Name/ID	Gender
			Matius	Laki-laki
			Ruth	Perempuan
			Salli	Perempuan
			Stephen	Laki-laki
			Tiffany	Perempuan
9	Belanda (Belgia)	NI-be	Lisa	Perempuan
10	Belanda (Belanda)	NI-NL	Laura	Perempuan
11	Perancis (Belgia)	FR-Be	Isabelle	Perempuan
12	Perancis (Kanada)	FR-Ca	Gabrielle	Perempuan
			Liam	Laki-laki
13	Prancis (Prancis)	FR-fr	Ambre	Perempuan
			Céline	Perempuan
			Florian	Laki-laki
			Léa	Perempuan
			Remi	Laki-laki
14	Jerman (Austria)	De-at	Hannah	Perempuan
15	Jerman (Jerman)	De-de	Daniel	Laki-laki
			Lennart	Laki-laki
			Vicki	Perempuan
16	Jerman (Swiss)	De-ch	Sabrina	Perempuan

	Bahasa	Kode bahasa	Name/ID	Gender
17	Italia (Italia)	It-itu	Beatrice	Perempuan
			Bianca	Perempuan
			Lorenzo	Laki-laki
18	Korea (Korea)	Ko-kr	Seoyeon	Perempuan
19	Polandia (Polandia)	PL-PL	Ewa	Perempuan
			Ola	Perempuan
20	Portugis (Brasil)	Pt-BR	Camila	Perempuan
21	Spanyol (Meksiko)	ES-mx	Andrés	Laki-laki
			Mia	Perempuan
22	Spanyol (Spanyol)	ES-es	Lucia	Perempuan
			Sergio	Laki-laki
23	Spanyol (AS)	es-AS	Lupe	Perempuan
			Pedro	Laki-laki

Note

Biaya suara generatif ditentukan pada halaman informasi [harga Amazon Polly](#).

Kompatibilitas fitur dan wilayah

Suara generatif Amazon Polly tersedia di wilayah berikut:

- AS Timur (Virginia N.): us-east-1
- Eropa (Frankfurt am Main): eu-central-1
- AS Barat (Oregon): us-west-2

- Asia Pasifik (Tokyo): ap-northeast-1
- Asia Pasifik (Seoul): ap-northeast-2
- Asia Pasifik (Singapura): ap-southeast-1
- Eropa (London): eu-west-2
- Kanada (Tengah): ca-central-1
- Eropa (Zürich): eu-central-2
- Wilayah lain tidak tersedia

Fitur berikut ini didukung untuk suara generatif:

- API Streaming Bidirectional sekarang ditawarkan di Generative Engine dan memungkinkan untuk streaming input dan output pada saat yang bersamaan. API ini tersedia di AWS wilayah berikut: AS Timur (Virginia N.), Eropa (Frankfurt), AS Barat (Oregon), Asia Pasifik (Singapura), Eropa (London), Kanada (Tengah), dan Eropa (Zurich). Kunjungi [dokumentasi](#) untuk mempelajari lebih lanjut tentang cara menggunakannya.
- Real-time dan operasi sintesis ucapan asinkron.
- Gaya berbicara penyiar berita tidak didukung di mesin Generatif.
- Banyak (tetapi tidak semua) tag SSML didukung oleh Amazon Polly. Untuk informasi selengkapnya tentang tag NTTS-supported SSML, lihat Tag SSML yang [didukung](#)
- Seperti halnya suara standar, Anda dapat memilih dari berbagai laju pengambilan sampel untuk mengoptimalkan bandwidth dan kualitas audio untuk aplikasi Anda. Tingkat pengambilan sampel yang valid untuk suara standar dan saraf adalah 8 kHz, 16 kHz, 22 kHz, atau 24 kHz. Default untuk suara standar adalah 22 kHz. Default untuk suara generatif adalah 24 kHz. Amazon Polly mendukung format streaming audio MP3, OGG (Vorbis), dan PCM mentah.

Support untuk menghasilkan tanda ucapan saat ini tidak tersedia.

Note

Saat ini wilayah Eropa (London), Kanada (Tengah) dan Eropa (Zurich) hanya mendukung suara Generatif berikut: Joanna (en-US), Ruth (en-US), Salli (en-US), Stephen (en-US), Tiffany (en-US), Amy (en-GB), Brian (en-GB), Olivia (en-AU), Florian (fr-FR), Ambre (fr-FR), Lorenzo (IT-IT), Beatrice (IT-IT), Melati (en-SG), Aria (en-NZ), Lennart (De-DE), Vicki (De-

de), Sabrina (de-CH), Hannah (De-AT), Niamh (en-le), Camila (Pt-BR), Lisa (NI-be), dan Seoyeon (Ko-kr)

Note

Dalam hal halusinasi model yang tidak mungkin terjadi, (dan dengan perilaku model mesin Generatif untuk merender token ucapan dengan token) mekanisme penghentian darurat yang diberlakukan sudah ada. Mekanisme bawaan menghentikan model dari rendering ucapan lebih jauh. Fitur keamanan ini didasarkan pada analisis data di mana model memiliki potensi untuk berhalusinasi, biasanya di akhir kalimat.

Mungkin ada kasus di mana model berpikir itu akan berhalusinasi dan kemudian mungkin berakhir memotong kata selama langkah generasi, sehingga merender setengah kata. Ini berpotensi menghasilkan hasil yang tidak pantas.

Long-form suara

Amazon Polly memiliki Long-form mesin yang menghasilkan suara seperti manusia, sangat ekspresif, dan mahir secara emosional. Long-form Suara dirancang untuk memikat perhatian pendengar untuk konten yang lebih panjang, seperti artikel berita, materi pelatihan, atau video pemasaran.

Long-form Suara Amazon Polly dikembangkan dengan teknologi TTS pembelajaran mendalam mutakhir. Model ini belajar untuk mereplikasi fonem, prosodi, intonasi, dan aspek fonetik dan akustik lainnya dari bahasa manusia, menghasilkan output ucapan yang sangat alami.

Long-form Mesin menggunakan embeddings teks untuk menafsirkan makna teks. Menggunakan penyematan teks, Long-form mesin dapat menghasilkan penekanan, jeda, dan nada suara alami yang benar. Hasilnya adalah suara yang menggabungkan berbagai elemen emosional yang ada dalam komunikasi manusia. Ini termasuk meniru dialog yang mengejutkan atau membedakan dari narasi. Bersama-sama, ini menciptakan produk pidato premium yang terdengar seperti manusia hidup.

Note

Teknologi canggih yang mendasari suara-suara ini termasuk dalam paradigma AI generatif untuk pemodelan bahasa dan suara. Efek samping dari teknologi ini adalah bahwa setiap

pembaruan pada data pelatihan dan model dapat menghasilkan sedikit variasi pada cara suara terdengar, bahkan jika kualitas keseluruhannya meningkat dengan pembaruan model. Ini dapat berdampak pada kasus penggunaan dengan bagian konten berbeda yang disintesis dalam jangka waktu yang lama — misalnya, musim podcast.

Tersedia suara bentuk panjang

Amazon Polly saat ini menawarkan empat en-US dan dua ES-es suara bentuk panjang. Kedua bahasa memiliki suara perempuan dan laki-laki yang tersedia. Suara bentuk panjang bahasa Inggris Daniel, Gregory, dan Ruth juga tersedia dalam varian NTTS percakapan.

	Bahasa	Kode bahasa	Name/ID	Gender
1	Inggris (US)	en-US	Danielle	Perempuan
			Gregory	Laki-laki
			Ruth	Perempuan
			Patrick	Laki-laki
2	Spanyol (Spanyol)	ES-es	Alba	Perempuan
			Raul	Laki-laki

Kompatibilitas fitur dan wilayah

Suara bentuk panjang Amazon Polly tersedia di wilayah berikut:

- AS Timur (Virginia N.): us-east-1
- Wilayah lain tidak tersedia

Long-form Mesin Amazon Polly mendukung fitur-fitur berikut:

- Real-time dan operasi sintesis ucapan asinkron.
- Semua [tanda bicara](#).

- Banyak (tetapi tidak semua) tag SSML didukung oleh Amazon Polly. Untuk informasi selengkapnya tentang tag NTTS-supported SSML, lihat Tag SSML yang [didukung](#)
- Seperti halnya suara standar, Anda dapat memilih dari berbagai laju pengambilan sampel untuk mengoptimalkan bandwidth dan kualitas audio untuk aplikasi Anda. Tingkat pengambilan sampel yang valid untuk suara standar, bentuk panjang, dan saraf adalah: 8 kHz, 16 kHz, 22kHz, atau 24 kHz. Default untuk suara standar adalah 22 kHz. Default untuk suara bentuk panjang dan saraf adalah 24 kHz. Amazon Polly mendukung format streaming audio MP3, OGG (Vorbis), dan PCM mentah.

Note

Long-form biaya suara ditentukan pada halaman informasi [harga Amazon Polly](#).

Suara saraf

Amazon Polly memiliki mesin Neural text-to-speech (NTTS) yang dapat menghasilkan suara berkualitas lebih tinggi daripada suara standarnya. Suara TTS standar menggunakan sintesis konkatenatif. Mesin standar menggabungkan fonem ucapan yang direkam, menghasilkan ucapan sintesis yang terdengar sangat alami. Namun, variasi bicara yang tak terhindarkan dan teknik yang digunakan untuk mengelompokkan bentuk gelombang membatasi kualitas bicara. Mesin Amazon Polly NTTS tidak menggunakan sintesis concatenative standar untuk menghasilkan ucapan. Ini memiliki dua bagian:

- Jaringan saraf — yang mengubah urutan fonem (unit bahasa paling dasar) menjadi urutan spektrogram. (Spektrogram adalah snapshot dari tingkat energi dalam pita frekuensi yang berbeda.)
- Vocoder — yang mengubah spektrogram menjadi sinyal audio yang hampir terus menerus.

Komponen pertama dari sistem TTS saraf adalah model urutan-ke-urutan. Model ini tidak membuat hasilnya hanya dari input yang sesuai tetapi juga mempertimbangkan bagaimana urutan elemen input bekerja sama. Model memilih spektrogram yang dikeluarkannya sehingga pita frekuensinya menekankan fitur akustik yang digunakan otak manusia saat memproses ucapan.

Output dari model ini kemudian diteruskan ke vocoder saraf. Ini mengubah spektrogram menjadi bentuk gelombang bicara. Ketika dilatih pada kumpulan data besar yang digunakan untuk

membangun sistem sintesis gabungan tujuan umum, pendekatan urutan-ke-urutan ini akan menghasilkan suara berkualitas lebih tinggi dan terdengar lebih alami.

Suara saraf yang tersedia

Suara saraf tersedia dalam 36 bahasa dan varian bahasa. Tabel berikut mencantumkan suara-suara.

	Varian bahasa dan bahasa	Kode bahasa	Name/ID	Gender
1	Arab (Teluk)	AR-ae	Hala	Perempuan
			Zayd	Laki-laki
2	Belanda Belgia (Flemish)	NI-be	Lisa	Perempuan
3	Catalan	CA-es	Arlet	Perempuan
4	Ceko	cs-cz	Jitka	Perempuan
5	Tionghoa (Kanton)	Yue-CN	Hiujin	Perempuan
6	Mandarin (Mandarin)	Cmn-cn	Zhiyu	Perempuan
7	Denmark	Da-dk	Sofie	Perempuan
8	Belanda	NI-NL	Laura	Perempuan
9	Inggris (Australi a)	En-au	Olivia	Perempuan
10	Inggris (Inggris)	id - GB	Amy*	Perempuan
			Emma	Perempuan
			Brian	Laki-laki
			Arthur	Laki-laki

	Varian bahasa dan bahasa	Kode bahasa	Name/ID	Gender
11	Inggris (India)	En-in	Kajal	Perempuan
12	Inggris (Irlandia)	En-ie	Niamh	Perempuan
13	Inggris (Selandia Baru)	en-NZ	Aria	Perempuan
14	English (Singaporean)	En-SG	Melati	Perempuan
15	Inggris (Afrika Selatan)	En-za	Ayanda	Perempuan
16	Inggris (US)	en-US	Danielle	Perempuan
			Gregory	Laki-laki
			Ivy	Perempuan (anak)
			Joanna*	Perempuan
			Kendra	Perempuan
			Kimberly	Perempuan
			Salli	Perempuan
			Joey	Laki-laki
			Justin	Pria (anak)
			Kevin	Pria (anak)
			Matthew*	Laki-laki
			Ruth	Perempuan
			Stephen	Laki-laki

	Varian bahasa dan bahasa	Kode bahasa	Name/ID	Gender
17	Finlandia	Fi-Fi	Suvi	Perempuan
18	Perancis (Belgia)	FR-Be	Isabelle	Perempuan
19	Perancis (Kanada)	FR-Ca	Gabrielle	Perempuan
			Liam	Laki-laki
20	Perancis	FR-fr	Léa	Perempuan
			Remi	Laki-laki
21	Jerman	De-de	Vicki	Perempuan
			Daniel	Laki-laki
22	Jerman (Austria)	De-at	Hannah	Perempuan
23	Jerman (Swiss)	De-ch	Sabrina	Perempuan
24	Hindi	Hi-in	Kajal	Perempuan
25	Italia	It-itu	Bianca	Perempuan
			Adriano	Laki-laki
26	Jepang	Ja-JP	Takumi	Laki-laki
			Kazuha	Perempuan
			Tomoko	Perempuan
27	Korea	Ko-kr	Seoyeon	Perempuan
			Jihye	Perempuan
28	Norwegia	NB-Tidak	Ida	Perempuan
29	Polandia	PL-PL	Ola	Perempuan

	Varian bahasa dan bahasa	Kode bahasa	Name/ID	Gender
30	Portugis (Brasil)	Pt-BR	Camila	Perempuan
			Vitória/Vitoria	Perempuan
			Thiago	Laki-laki
31	Portugis (Eropa)	Pt-PT	Inês/Ines	Perempuan
32	Spanyol (Spanyol)	ES-es	Lucia	Perempuan
			Sergio	Laki-laki
33	Spanyol (Meksiko)	ES-mx	Mia	Perempuan
			Andrés	Laki-laki
34	Spanyol (AS)	es-AS	Lupe*	Perempuan
			Pedro	Laki-laki
35	Swedia	SV-SE	Elin	Perempuan
36	Turki	TR-TR	Burcu	Perempuan

* Suara Amy, Joanna, Lupe, dan Matthew dapat digunakan dengan gaya berbicara penyiar berita. Untuk informasi selengkapnya, lihat [Menerapkan suara penyiar berita](#).

Kompatibilitas fitur dan wilayah

Suara saraf tidak tersedia di semua AWS Wilayah, juga tidak mendukung semua fitur Amazon Polly.

Suara saraf didukung di wilayah berikut:

- AS Timur (Virginia N.): us-east-1
- AS Barat (Oregon): us-west-2
- Afrika (Cape Town): af-south-1
- Asia Pasifik (Tokyo): ap-northeast-1

- Asia Pasifik (Seoul): ap-northeast-2
- Asia Pasifik (Osaka): ap-northeast-3
- Asia Pasifik (Mumbai): ap-south-1
- Asia Pasifik (Singapura): ap-southeast-1
- Asia Pasifik (Sydney): ap-southeast-2
- Asia Pasifik (Malaysia): ap-tenggara - 5
- Asia Pasifik (Thailand): ap-tenggara - 7
- Kanada (Tengah): ca-central-1
- Eropa (Frankfurt am Main): eu-central-1
- Eropa (Irlandia): eu-west-1
- Eropa (London): eu-west-2
- Eropa (Paris): eu-west-3
- Eropa (Spanyol): eu-south-2
- Eropa (Zürich): eu-central-2
- AWS GovCloud (US-West): us-gov-west-1

Titik akhir dan protokol untuk Wilayah ini identik dengan yang digunakan untuk suara standar. Untuk informasi selengkapnya, lihat [titik akhir dan kuota Amazon Polly](#).

Fitur-fitur berikut didukung untuk suara saraf:

- Real-time dan operasi sintesis ucapan asinkron.
- Gaya berbicara penyiar berita. Untuk informasi lebih lanjut tentang gaya berbicara, lihat [Menerapkan suara penyiar berita](#).
- Semua tanda bicara.
- Banyak (tetapi tidak semua) tag SSML yang didukung oleh Amazon Polly. Untuk informasi selengkapnya tentang tag TTSS-supported SSML, lihat Tag yang Didukung.

Seperti halnya suara standar, Anda dapat memilih dari berbagai laju pengambilan sampel untuk mengoptimalkan bandwidth dan kualitas audio untuk aplikasi Anda. Tingkat pengambilan sampel yang valid untuk suara standar dan saraf adalah 8 kHz, 16 kHz, 22 kHz, atau 24 kHz. Default untuk suara standar adalah 22 kHz. Default untuk suara saraf adalah 24 kHz. Amazon Polly mendukung format streaming audio MP3, OGG (Vorbis), dan PCM mentah.

Suara standar

Amazon Polly memiliki mesin standar yang menggunakan sintesis concatenative. Mesin standar menggabungkan fonem ucapan yang direkam, menghasilkan ucapan sintesis yang terdengar sangat alami.

Tersedia suara Standar

Amazon Polly saat ini menawarkan 40 suara standar wanita dan 20 pria dalam 29 varian bahasa dan bahasa.

	Bahasa	Kode bahasa	Name/ID	Gender
1	Arab	arb	Zeina	Perempuan
2	Mandarin (Mandarin)	Cmn-cn	Zhiyu	Perempuan
3	Denmark	Da-dk	Naja	Perempuan
			Ibu	Laki-laki
4	Belanda	NI-NL	Lotte	Perempuan
			Ruben	Laki-laki
5	Inggris (Australia)	En-au	Nicole	Perempuan
			Russell	Laki-laki
6	Inggris (Inggris)	id - GB	Amy	Perempuan
			Emma	Perempuan
			Brian	Laki-laki
7	Inggris (India)	En-in	Aditi	Perempuan
			Raveena	Perempuan
8	Inggris (US)	en-US	Ivy	Perempuan

	Bahasa	Kode bahasa	Name/ID	Gender
			Joanna	Perempuan
			Kendra	Perempuan
			Kimberly	Perempuan
			Salli	Perempuan
			Joey	Laki-laki
			Kevin	Laki-laki
9	Inggris (Welsh)	id- GB-WLS	Geraint	Laki-laki
10	Perancis	FR-fr	Cé line/Celine	Perempuan
			Léa	Perempuan
			Mathieu	Laki-laki
11	Perancis (Kanada)	FR-Ca	Chantal	Perempuan
12	Jerman	De-de	Marlene	Perempuan
			Vicki	Perempuan
			Hans	Laki-laki
13	Hindi	Hi-in	Aditi	Perempuan
14	Islandia	As-adalah	Dó ra/Dora	Perempuan
			Karl	Laki-laki
15	Italia	It-itu	Carla	Perempuan
			Bianca	Perempuan
			Giorgio	Laki-laki

	Bahasa	Kode bahasa	Name/ID	Gender
16	Jepang	Ja-JP	Mizuki	Perempuan
			Takumi	Laki-laki
17	Korea	Ko-kr	Seoyeon	Perempuan
18	Norwegia	NB-Tidak	Liv	Perempuan
19	Polandia	PL-PL	Ewa	Perempuan
			Maja	Perempuan
			Jacek	Laki-laki
			Jan	Laki-laki
20	Portugis (Brasil)	Pt-BR	Camila	Perempuan
			Vitória/Vitoria	Perempuan
			Ricardo	Laki-laki
21	Portugis (Eropa)	Pt-PT	Inês/Ines	Perempuan
			Cristiano	Laki-laki
22	Rumania	Ro-ro	Carmen	Perempuan
23	Rusia	Ru-ru	Tatiana	Perempuan
			Maxim	Laki-laki
24	Spanyol (Spanyol)	ES-es	Conchita	Perempuan
			Lucia	Perempuan
			Enrique	Laki-laki
25	Spanyol (Meksiko)	ES-mx	Mia	Perempuan

	Bahasa	Kode bahasa	Name/ID	Gender
26	Spanyol (AS)	es-AS	Lupe	Perempuan
			Pene lope/Pene lope	Perempuan
			Miguel	Laki-laki
27	Swedia	SV-SE	Astrid	Perempuan
28	Turki	TR-TR	Filiz	Laki-laki
29	Welsh	Cy-GB	Gwyneth	Perempuan

Kompatibilitas fitur dan wilayah

Suara standar Amazon Polly tersedia di wilayah Amazon Polly berikut:

- AS Timur (Virginia N.): us-east-1
- AS Timur (Ohio): us-east-2
- AS Barat (California N.): us-west-1
- AS Barat (Oregon): us-west-2
- Afrika (Cape Town): af-south-1
- Asia Pasifik (Hong Kong): ap-east-1
- Asia Pasifik (Tokyo): ap-northeast-1
- Asia Pasifik (Seoul): ap-northeast-2
- Asia Pasifik (Osaka): ap-northeast-3
- Asia Pasifik (Mumbai): ap-south-1
- Asia Pasifik (Singapura): ap-southeast-1
- Asia Pasifik (Sydney): ap-southeast-2
- Asia Pasifik (Malaysia): ap-tenggara - 5
- Tiongkok (Ningxia): cn-barat laut-1;
- Kanada (Tengah): ca-central-1
- Eropa (Frankfurt am Main): eu-central-1

- Eropa (Irlandia): eu-west-1
- Eropa (London): eu-west-2
- Eropa (Paris): eu-west-3
- Eropa (Spanyol): eu-south-2
- Eropa (Stockholm): eu-north-1
- Timur Tengah (Bahrain): me-south-1
- Amerika Selatan (Sao Paulo): sa-east-1
- AWS GovCloud (US-West): kami-gov-barat-1

Titik akhir dan protokol untuk Wilayah ini identik dengan yang digunakan untuk suara Neural. Untuk informasi selengkapnya, lihat [titik akhir dan kuota Amazon Polly](#).

Mesin standar Amazon Polly mendukung fitur-fitur berikut (TBD):

- Real-time dan operasi sintesis ucapan asinkron.
- Semua [tanda bicara](#).
- Banyak (tetapi tidak semua) tag SSML didukung oleh Amazon Polly. Untuk informasi selengkapnya tentang tag NTTS-supported SSML, lihat Tag [SSML yang didukung](#).
- Anda dapat memilih dari berbagai tingkat pengambilan sampel untuk mengoptimalkan bandwidth dan kualitas audio untuk aplikasi Anda. Tingkat pengambilan sampel default untuk suara standar adalah 22 kHz. Amazon Polly mendukung format streaming audio MP3, OGG (Vorbis), dan PCM mentah.

Note

Biaya suara standar ditentukan pada halaman informasi [harga Amazon Polly](#).

Memilih mesin suara

Anda dapat mengakses suara Amazon Polly melalui konsol Amazon Polly atau. AWS CLI

Untuk memilih mesin suara di konsol

1. Buka konsol Amazon Polly di. <https://console.aws.amazon.com/polly/>

2. Dari konsol Amazon Polly, pilih mesin suara yang diinginkan.
3. Pilih suara yang diinginkan dari menu drop-down suara.
4. Hasilkan audio TTS dengan teks pilihan Anda.

Untuk memilih mesin suara di AWS CLI, tentukan Engine dan VoiceId dalam operasi `SynthesizeSpeech` atau `StartSpeechSynthesisTask` API. [Untuk beberapa contoh, lihat contoh kode mulai cepat dan contoh Python.](#)

Tanda bicara

Tanda ucapan adalah metadata yang menggambarkan ucapan yang Anda sintesis, seperti di mana kalimat atau kata dimulai dan diakhiri dalam aliran audio. Saat Anda meminta tanda ucapan untuk teks Anda, Amazon Polly mengembalikan metadata ini alih-alih ucapan yang disintesis. Dengan menggunakan tanda ucapan bersama dengan aliran audio ucapan yang disintesis, Anda dapat memberikan pengalaman visual yang disempurnakan kepada aplikasi Anda.

Misalnya, menggabungkan metadata dengan aliran audio dari teks Anda dapat memungkinkan Anda untuk menyinkronkan ucapan dengan animasi wajah (sinkronisasi bibir) atau untuk menyorot kata-kata tertulis saat diucapkan.

Tanda pidato tersedia saat menggunakan mesin saraf, bentuk panjang, atau standar. text-to-speech

Topik

- [Jenis tanda bicara](#)
- [Visemes dan Amazon Polly](#)
- [Output tanda ucapan](#)
- [Meminta tanda pidato](#)
- [Tanda bicara tanpa contoh SSML](#)
- [Tanda ucapan dengan contoh SSML](#)

Jenis tanda bicara

Anda meminta tanda ucapan menggunakan [SpeechMarkTypes](#) opsi untuk [StartSpeechSynthesisTask](#) perintah [SynthesizeSpeech](#) atau. Anda menentukan elemen metadata yang ingin Anda kembalikan dari teks masukan Anda. Anda dapat meminta sebanyak empat jenis metadata tetapi Anda harus menentukan setidaknya satu per permintaan. Tidak ada output audio yang dihasilkan dengan permintaan.

Dalam AWS CLI, misalnya:

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

Amazon Polly menghasilkan tanda bicara menggunakan elemen-elemen berikut:

- kalimat - Menunjukkan elemen kalimat dalam teks input.
- kata - Menunjukkan elemen kata dalam teks.
- viseme — Menjelaskan gerakan wajah dan mulut yang sesuai dengan setiap fonem yang diucapkan. Untuk informasi selengkapnya, lihat [Visemes dan Amazon Polly](#).
- ssml - Menjelaskan <mark>elemen dari teks masukan SSML. Untuk informasi selengkapnya, lihat [Menghasilkan pidato dari dokumen SSML](#).

Visemes dan Amazon Polly

Viseme mewakili posisi wajah dan mulut saat mengucapkan sepatah kata pun. Ini adalah padanan visual dari fonem, yang merupakan unit akustik dasar dari mana sebuah kata terbentuk. Visemes adalah blok pembangun visual dasar bicara.

Setiap bahasa memiliki seperangkat viseme yang sesuai dengan fonem spesifiknya. Dalam suatu bahasa, setiap fonem memiliki viseme yang sesuai yang mewakili bentuk yang dibuat mulut saat membentuk suara. Namun, tidak semua viseme dapat dipetakan ke fonem tertentu karena banyak fonem tampak sama ketika diucapkan, meskipun kedengarannya berbeda. Misalnya, dalam bahasa Inggris, kata “pet” dan “bet” berbeda secara akustik. Namun, ketika diamati secara visual (tanpa suara), mereka terlihat persis sama.

Bagan berikut menunjukkan sebagian daftar fonem Alfabet Fonetik Internasional (IPA) dan Metode Penilaian Pidato Diperpanjang Alfabet Fonetik (X-SAMPA) serta viseme yang sesuai untuk suara bahasa Inggris AS.

Untuk tabel lengkap dan tabel untuk semua bahasa yang tersedia, lihat [Bahasa di Amazon Polly](#).

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
Konsonan				
b	b	Plosif bilabial bersuara	b ed	p
d	d	Plosif alveolar bersuara	d ig	t
ɹ̥	dʒ	Afrikat postalveolar bersuara	j ump	D

IPA	X-SAMPA	Deskripsi	Contoh	Viseme
ð	D	Frikatif gigi bersuara	th en	T
f	f	Frikatif labiodental tanpa suara	f saya	f
g	g	Velar plosif bersuara	g ame	k
-h	-h	Frikatif glottal tanpa suara	h rumah	k
...	...	...	...	...

Output tanda ucapan

Amazon Polly mengembalikan objek tanda ucapan dalam aliran JSON yang dibatasi garis. Objek tanda ucapan berisi bidang-bidang berikut:

- waktu — stempel waktu dalam milidetik dari awal aliran audio yang sesuai
- jenis — jenis tanda bicara (kalimat, kata, viseme, atau ssml)
- start — offset dalam byte (bukan karakter) dari awal objek dalam teks input (tidak termasuk tanda viseme)
- akhir — offset dalam byte (bukan karakter) dari ujung objek dalam teks input (tidak termasuk tanda viseme)
- nilai — ini bervariasi tergantung pada jenis tanda bicara
 - SSML: Tag SSML <mark>
 - viseme: nama viseme
 - kata atau kalimat: substring dari teks input, seperti yang dibatasi oleh bidang awal dan akhir

Misalnya, Amazon Polly menghasilkan objek tanda word ucapan berikut dari teks “Mary had a little lamb”:

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

Kata yang dijelaskan ("had") dimulai 373 milidetik setelah aliran audio dimulai, dan dimulai pada byte 5 dan berakhir pada byte 8 dari teks input.

Note

Metadata ini untuk voice-id. Joanna Jika Anda menggunakan suara lain dengan teks input yang sama, metadata mungkin berbeda.

Meminta tanda pidato

Anda dapat menggunakan konsol atau `synthesize-speech` perintah untuk meminta tanda ucapan dari Amazon Polly. Anda kemudian dapat melihat metadata atau menyimpannya ke file.

Console

Untuk menghasilkan tanda bicara di konsol

1. Masuk ke AWS Management Console dan buka konsol Amazon Polly di <https://console.aws.amazon.com/polly/>
2. Pilih tab Text-to-Speech.
3. Nyalakan SSML untuk menggunakan SSML.
4. Ketik atau tempel teks Anda ke dalam kotak input.
5. Untuk Bahasa, pilih bahasa teks Anda.
6. Untuk Suara, pilih suara yang ingin Anda gunakan.
7. Untuk mengubah pengucapan teks, perluas Pengaturan tambahan, aktifkan Sesuaikan pengucapan, dan untuk Terapkan leksikon, pilih leksikon yang diinginkan.
8. Untuk memverifikasi pidato, pilih Dengarkan.
9. Aktifkan pengaturan format file Ucapan.

Note

Format pengunduhan MP3, OGG, PCM, MU-law, atau A-law tidak akan menghasilkan tanda bicara.

10. Untuk Format File, pilih Tanda ucapan.
11. Untuk jenis tanda Pidato, pilih jenis tanda bicara yang akan dihasilkan. Opsi untuk memilih metadata SSML hanya tersedia saat SSML aktif. Untuk informasi lebih lanjut tentang menggunakan SSML dengan Amazon Polly lihat. [Menghasilkan pidato dari dokumen SSML](#)
12. Pilih Unduh.

AWS CLI

Selain teks input, elemen berikut diperlukan untuk mengembalikan metadata ini:

- `output-format`

Amazon Polly hanya mendukung format JSON saat mengembalikan tanda ucapan.

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

Jika Anda menggunakan format keluaran yang tidak didukung, Amazon Polly melempar pengecualian.

- `voice-id`

Untuk memastikan bahwa metadata cocok dengan aliran audio terkait, tentukan suara yang sama yang digunakan untuk menghasilkan aliran audio ucapan yang disintesis. Suara yang tersedia tidak memiliki tingkat bicara yang identik. Jika Anda menggunakan suara selain yang digunakan untuk menghasilkan ucapan, metadata tidak akan cocok dengan aliran audio.

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

- `speech-mark-types`

Tentukan jenis atau jenis tanda bicara yang Anda inginkan. Anda dapat meminta salah satu atau semua jenis tanda ucapan, tetapi harus menentukan setidaknya satu jenis.

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

- `text-type`

Teks biasa adalah teks input default untuk Amazon Polly, jadi Anda harus menggunakan `text-type ssml` jika Anda ingin mengembalikan tanda ucapan SSML.

- outfile

Tentukan file output yang metadata ditulis.

```
MaryLamb.txt
```

AWS CLI Contoh berikut diformat untuk Unix, Linux, dan macOS. Untuk Windows, ganti karakter kelanjutan backslash (\) Unix di akhir setiap baris dengan tanda sisipan (^) dan gunakan tanda kutip penuh (") di sekitar teks input dengan tanda kutip tunggal (') untuk tag interior.

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

Tanda bicara tanpa contoh SSML

Contoh berikut menunjukkan kepada Anda seperti apa metadata yang diminta di layar Anda untuk kalimat sederhana: "Mary had a little lamb." Untuk mempermudah, kami tidak menyertakan tanda ucapan SSML dalam contoh ini.

AWS CLI Contoh berikut diformat untuk Unix, Linux, dan macOS. Untuk Windows, ganti karakter kelanjutan backslash (\) Unix di akhir setiap baris dengan tanda sisipan (^) dan gunakan tanda kutip penuh (") di sekitar teks input dengan tanda kutip tunggal (') untuk tag interior.

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

Saat Anda membuat permintaan ini, Amazon Polly mengembalikan yang berikut ini di file.txt:

```
{"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"
}
```

Dalam output ini, setiap bagian teks dipecah dalam hal tanda bicara:

- Kalimat “Mary punya anak domba kecil.”
- Setiap kata dalam teks: “Maria”, “memiliki”, “a”, “kecil”, dan “domba.”
- Viseme untuk setiap suara dalam aliran audio yang sesuai: “p”, “E”, “r”, “i”, dan sebagainya. Untuk informasi lebih lanjut tentang visemes lihat [Visemes dan Amazon Polly](#)

Tanda ucapan dengan contoh SSML

Proses menghasilkan tanda ucapan dari teks yang disempurnakan SSML mirip dengan proses ketika SSML tidak ada. Gunakan `synthesize-speech` perintah, dan tentukan teks yang disempurnakan SSML dan jenis tanda ucapan yang Anda inginkan, seperti yang ditunjukkan pada contoh berikut. Untuk membuat contoh lebih mudah dibaca, kami tidak menyertakan tanda ucapan viseme, tetapi ini dapat disertakan juga.

AWS CLI Contoh berikut diformat untuk Unix, Linux, dan macOS. Untuk Windows, ganti karakter kelanjutan backslash (\) Unix di akhir setiap baris dengan tanda sisipan (^) dan gunakan tanda kutip penuh (") di sekitar teks input dengan tanda kutip tunggal (') untuk tag interior.

```
aws polly synthesize-speech \
```

```
--output-format json \  
--voice-id Joanna \  
--text-type ssm1 \  
--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
```

Saat Anda membuat permintaan ini, Amazon Polly mengembalikan yang berikut ini di file.txt:

```
{"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"}
```

Menghasilkan pidato dari dokumen SSML

Anda dapat menggunakan Amazon Polly untuk menghasilkan ucapan dari teks biasa atau dari dokumen yang ditandai dengan Speech Synthesis Markup Language (SSML). Menggunakan teks yang disempurnakan SSML memberi Anda kontrol tambahan atas cara Amazon Polly menghasilkan ucapan dari teks yang Anda berikan.

Dengan tag SSML, Anda dapat menyesuaikan dan mengontrol aspek ucapan seperti pengucapan, volume, dan kecepatan bicara. Dalam AWS Management Console, teks yang disempurnakan SSML yang ingin Anda konversi ke audio dimasukkan pada tab SSML halaman. Text-to-Speech Meskipun teks yang dimasukkan dalam teks biasa bergantung pada pengaturan default untuk bahasa dan suara yang Anda pilih, teks yang disempurnakan dengan SSML memberi tahu Amazon Polly tidak hanya apa yang ingin Anda katakan, tetapi bagaimana Anda ingin mengatakannya. Kecuali untuk tag SSML yang ditambahkan, Amazon Polly mensintesis teks yang disempurnakan SSML dengan cara yang sama seperti mensintesis teks biasa. Untuk informasi selengkapnya, lihat [Mensintesis pidato dengan contoh Amazon Polly](#).

Saat menggunakan SSML, Anda melampirkan seluruh teks dalam `< speak >` tag untuk memberi tahu Amazon Polly bahwa Anda menggunakan SSML. Contoh:

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

Anda kemudian menggunakan tag SSML tertentu pada teks di dalam `< speak >` tag untuk menyesuaikan cara Anda ingin teks berbunyi. Anda dapat menambahkan jeda, mengubah kecepatan bicara, menurunkan atau menaikkan volume suara, atau menambahkan banyak penyesuaian lain sehingga teks terdengar tepat untuk Anda. Untuk daftar lengkap tag SSML yang dapat Anda gunakan, lihat. [Tag SSML yang didukung](#)

Misalnya, Anda dapat menyertakan jeda panjang dalam teks Anda, atau mengubah kecepatan bicara atau nada. Pilihan lain termasuk:

- menekankan kata atau frasa tertentu
- menggunakan pengucapan fonetik
- termasuk suara pernapasan
- berbisik
- menggunakan gaya berbicara penyiar berita.

Untuk detail selengkapnya tentang tag SSML yang didukung oleh Amazon Polly dan cara menggunakannya, lihat [Tag SSML yang didukung](#)

Saat menggunakan SSML, ada beberapa karakter yang dipesan yang memerlukan perlakuan khusus. Ini karena SSML menggunakan karakter ini sebagai bagian dari kodenya. Untuk menggunakannya, Anda menggunakan entitas tertentu untuk menghindarinya. Untuk informasi selengkapnya, lihat [Karakter yang dipesan di SSML](#)

Amazon Polly menyediakan jenis kontrol ini dengan subset tag markup SSML yang ditentukan oleh [Speech Synthesis Markup Language \(SSML\) Versi 1.1, Rekomendasi W3C](#).

Anda dapat menggunakan SSML dalam konsol Amazon Polly atau dengan menggunakan AWS CLI. Topik berikut menunjukkan kepada Anda bagaimana Anda dapat menggunakan SSML untuk menghasilkan ucapan dan mengontrol output sehingga sesuai dengan kebutuhan Anda.

Topik

- [Karakter yang dipesan di SSML](#)
- [Menggunakan SSML di konsol](#)
- [Menggunakan SSML dengan perintah Synthesize-Speech](#)
- [Mensintesis dokumen yang disempurnakan SSML](#)
- [Tag SSML yang didukung](#)

Karakter yang dipesan di SSML

Ada lima karakter standar yang biasanya tidak dapat digunakan dalam pernyataan SSML. Entitas ini dicadangkan oleh spesifikasi bahasa. Karakter-karakter ini adalah sebagai berikut:

Karakter
melarikan
diri

tanda
kutip
(tanda
kutip
ganda)

Karena

melarikan

diri

&mpersand

tanda

kutip

atau

tanda

kutip

tunggal

kurang

dari

tanda

lebih

besar

dari

tanda

Karena SSML menggunakan karakter ini sebagai bagian dari kodenya, untuk menggunakan simbol-simbol ini di SSML, Anda harus melarikan diri dari karakter saat Anda menggunakannya. Anda menggunakan kode escape alih-alih karakter yang sebenarnya sehingga ditampilkan dengan benar sambil tetap membuat dokumen SSML yang valid. Misalnya, kalimat berikut

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

akan dirender di SSML sebagai

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

Dalam hal ini, karakter khusus untuk apostrof dan ampersand diloloskan sehingga dokumen SSML tetap valid.

Untuk simbol &, <, dan>, kode escape selalu diperlukan saat Anda menggunakan SSML. Selain itu, ketika Anda menggunakan tanda kutip apostrophe/tunggal (') sebagai tanda kutip, Anda juga harus menggunakan kode escape.

Namun, ketika Anda menggunakan tanda kutip ganda ("), atau tanda apostrophe/single kutip (') sebagai tanda kutip, maka apakah Anda menggunakan kode escape tergantung pada konteks atau tidak.

Tanda kutip ganda

- Harus diloloskan ketika dalam nilai atribut dibatasi oleh tanda kutip ganda. Misalnya, dalam AWS CLI kode berikut

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

- Tidak perlu melarikan diri ketika dalam konteks tekstual. Misalnya, dalam hal berikut

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

- Tidak perlu diloloskan ketika dalam nilai atribut dibatasi oleh tanda kutip tunggal. Misalnya, dalam AWS CLI kode berikut

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

Tanda kutip tunggal

- Harus melarikan diri ketika digunakan sebagai tanda kutip. Misalnya, dalam hal berikut

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

- Tidak perlu melarikan diri ketika dalam konteks tekstual. Misalnya, dalam hal berikut

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

- Tidak perlu diloloskan ketika dalam atribut kode dibatasi oleh tanda kutip ganda. Misalnya, dalam AWS CLI kode berikut

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

Menggunakan SSML di konsol

Dalam contoh berikut, Anda menggunakan tag SSML untuk memberi tahu Amazon Polly untuk mengganti “World Wide Web Consortium” dengan “W3C” ketika berbicara paragraf pendek. Anda juga menggunakan tag untuk memperkenalkan jeda dan membisikkan kata. Bandingkan hasil latihan ini dengan hasil latihan [Menerapkan leksikon \(Synthesizing Speech\)](#).

Untuk informasi lebih lanjut tentang SSML, dengan contoh, lihat. [Tag SSML yang didukung](#)

Untuk mensintesis ucapan dari teks yang disempurnakan SSML (konsol)

1. Masuk ke AWS Management Console dan buka konsol Amazon Polly di. <https://console.aws.amazon.com/polly/>
2. Jika belum ditampilkan, pilih tab Text-to-Speech.
3. Nyalakan SSML.
4. Ketik atau tempel teks berikut di kotak teks:

```
<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>
```

Tag SSML memberi tahu Amazon Polly cara merender teks:

- `<break time="1s"/>` memberitahu Amazon Polly untuk berhenti sejenak 1 detik antara dua kalimat pertama.
- `<sub alias="World Wide Web Consortium">W3C</sub>` memberitahu Amazon Polly untuk menggantikan World Wide Web Consortium untuk akronim W3C.
- `<amazon:effect name="whispered">Nice job</amazon:effect>` memberitahu Amazon Polly untuk membisikkan contoh kedua “Kerja bagus.”.

 Note

Saat Anda menggunakan AWS CLI, Anda melampirkan teks input dalam tanda kutip untuk membedakannya dari kode sekitarnya. Konsol Amazon Polly tidak menampilkan kode kepada Anda, jadi Anda tidak menyertakan teks input dalam tanda kutip saat menggunakannya.

5. Untuk Bahasa, pilih Inggris, AS, lalu pilih suara.
6. Untuk mendengarkan pidato, pilih Dengarkan.
7. Untuk menyimpan file ucapan, pilih Unduh. Jika Anda ingin menyimpannya dalam format yang berbeda, perluas Pengaturan tambahan, aktifkan Pengaturan format file ucapan dan pilih format yang Anda inginkan, lalu pilih Unduh.

Menggunakan SSML dengan perintah Synthesize-Speech

Contoh ini menunjukkan cara menggunakan `synthesize-speech` perintah dengan string SSML. Saat Anda menggunakan `synthesize-speech` perintah, Anda biasanya memberikan yang berikut:

- Teks masukan (wajib)
- Membuka dan menutup tag (wajib)
- Format output
- Sebuah suara

Dalam contoh ini, Anda menentukan string teks sederhana dalam tanda kutip bersama dengan `<speack></speack>` tag pembuka dan penutup yang diperlukan.

 Important

Meskipun Anda tidak menggunakan tanda kutip di sekitar teks input di konsol Amazon Polly, Anda harus menggunakannya dalam menggunakan AWS CLI. Penting juga bagi Anda untuk membedakan antara tanda kutip di sekitar teks input dan kutipan yang diperlukan untuk tag individual.

Misalnya, Anda dapat menggunakan tanda kutip standar (") untuk melampirkan teks input, dan tanda kutip tunggal (') untuk tag interior, atau sebaliknya. Opsi mana pun berfungsi untuk

Unix, Linux, dan macOS. Namun, dengan Windows Anda harus melampirkan teks input dalam tanda kutip standar dan menggunakan tanda kutip tunggal untuk tag.

Untuk semua sistem operasi, Anda dapat menggunakan tanda kutip standar (") untuk melampirkan teks input, dan tanda kutip tunggal (') untuk tag interior. Contoh:

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

Untuk Unix, Linux, dan macOS, Anda juga dapat menggunakan kebalikannya, dengan tanda kutip tunggal (') yang melampirkan teks input dan tanda kutip standar (") untuk tag interior:

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

AWS CLI Contoh berikut diformat untuk Unix, Linux, dan macOS. Untuk Windows, ganti karakter kelanjutan backslash (\) Unix di akhir setiap baris dengan tanda sisipan (^) dan gunakan tanda kutip penuh (") di sekitar teks input dengan tanda kutip tunggal (') untuk tag interior.

```
aws polly synthesize-speech \
--text-type ssm1 \
--text '<speak>Hello world</speak>' \
--output-format mp3 \
--voice-id Joanna \
speech.mp3
```

Untuk mendengar pidato yang disintesis, putar speech.mp3 file yang dihasilkan menggunakan pemutar audio apa pun.

Mensintesis dokumen yang disempurnakan SSML

Untuk teks input yang lebih panjang, Anda mungkin merasa lebih mudah untuk menyimpan konten SSML Anda ke file dan cukup tentukan nama file dalam perintah. `synthesize-speech` Misalnya Anda dapat menyimpan yang berikut ini ke file bernama `example.xml`:

```
<?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` atribut menentukan en-US (US English) sebagai bahasa teks input. Untuk informasi tentang bagaimana bahasa teks input dan bahasa suara yang dipilih memengaruhi `SynthesizeSpeech` operasi, lihat [Menentukan bahasa lain untuk kata-kata tertentu](#).

Untuk menjalankan file yang disempurnakan SSML

1. Simpan SSML ke file (misalnya, `example.xml`).
2. Jalankan `synthesize-speech` perintah berikut dari jalur tempat file XHTML disimpan dan tentukan file SSML sebagai input dengan mengganti teks `file:\\example.xml` input. Karena perintah ini menunjuk ke file alih-alih berisi teks input yang sebenarnya, Anda tidak menggunakan tanda kutip.

Note

AWS CLI Contoh berikut diformat untuk Unix, Linux, dan macOS. Untuk Windows, ganti karakter kelanjutan backslash (\) Unix di akhir setiap baris dengan tanda sisipan (^).

```
aws polly synthesize-speech \
--text-type ssm1 \
--text file://example.xml \
--output-format mp3 \
--voice-id Joanna \
speech.mp3
```

3. Untuk mendengar pidato yang disintesis, putar `speech.mp3` file yang dihasilkan menggunakan pemutar audio apa pun.

Tag SSML yang didukung

Semua tag kecuali `<amazon:domain name="news">` didukung untuk suara Standar. Ketersediaan tag untuk suara lain disediakan dalam tabel berikut.

Amazon Polly mendukung tag SSML berikut:

Tindakan	Tag SSML	Ketersedi aan suara saraf	Ketersedi aan suara bentuk panjang	Ketersedi aan suara generatif
Menambahkan jeda	<break>	Ketersedi aan penuh	Ketersedi aan penuh	Ketersedi aan penuh
Menekankan kata-kata	<emphasis>	Tidak tersedia	Tidak tersedia	Tidak tersedia
Menentukan bahasa lain untuk kata-kata tertentu	<lang>	Ketersedi aan penuh	Ketersedi aan penuh	Ketersedi aan penuh
Menempatkan tag khusus dalam teks Anda	<mark>	Ketersedi aan penuh	Ketersedi aan penuh	Ketersedi aan sebagian
Menambahkan jeda antar paragraf	<p>	Ketersedi aan penuh	Ketersedi aan penuh	Ketersedi aan penuh
Menggunakan pengucapa n fonetik	<phoneme>	Ketersedi aan penuh	Ketersedi aan penuh	Ketersedi aan sebagian
Mengontrol volume, kecepatan berbicara, dan nada	<prosody>	Ketersedi aan sebagian	Ketersedi aan sebagian	Availiabi litas sebagian
Menetapkan durasi maksimum untuk ucapan yang disintesis	<prosody amazon:max- duration>	Tidak tersedia	Tidak tersedia	Tidak tersedia
Menambahkan jeda di antara kalimat	<s>	Ketersedi aan penuh	Ketersedi aan penuh	Ketersedi aan penuh
Mengontrol bagaimana jenis kata khusus diucapkan	<say-as>	Ketersedi aan sebagian	Ketersedi aan penuh	Ketersedi aan penuh

Tindakan	Tag SSML	Ketersedi aan suara saraf	Ketersedi aan suara bentuk panjang	Ketersedi aan suara generatif
Mengidentifikasi teks yang disempurnakan SSML	< speak >	Ketersedi aan penuh	Ketersedi aan penuh	Ketersedi aan penuh
Mengucapkan akronim dan singkatan	< sub >	Ketersedi aan penuh	Ketersedi aan penuh	Ketersedi aan penuh
Meningkatkan pengucapan dengan menentukan bagian pidato	< w >	Ketersedi aan penuh	Ketersedi aan penuh	Ketersedi aan penuh
Menambahkan suara pernapasan	< amazon:auto-breaths >	Tidak tersedia	Tidak tersedia	Tidak tersedia
Gaya berbicara penyiar berita	< amazon:domain name="news" >	Pilih suara saraf saja	Tidak tersedia	Tidak tersedia
Menambahkan kompresi rentang dinamis	< amazon:effect name="drc" >	Ketersedi aan penuh	Ketersedi aan penuh	Tidak tersedia
Berbicara dengan lembut	< amazon:effect phonation ="soft" >	Tidak tersedia	Tidak tersedia	Tidak tersedia
Mengontrol timbre	< amazon:efek> vocal-tra ct-length	Tidak tersedia	Tidak tersedia	Tidak tersedia
Berbisik	< amazon:effect name="whispered" >	Tidak tersedia	Tidak tersedia	Tidak tersedia

Note

Jika Anda menggunakan tag SSML yang tidak didukung dalam format standar, neural, atau bentuk panjang, Anda akan mendapatkan kesalahan.

Mengidentifikasi teks yang disempurnakan SSML

< speak >

Tag ini didukung oleh format TTS generatif, bentuk panjang, saraf, dan standar.

< speak >Tag adalah elemen root dari semua teks Amazon Polly SSML. < speak >Semua teks yang disempurnakan SSML harus diapit dalam sepasang tag.

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

Menambahkan jeda

< break >

Tag ini didukung oleh format TTS generatif, bentuk panjang, saraf, dan standar.

Untuk menambahkan jeda ke teks Anda, gunakan < break >tag. Anda dapat mengatur jeda berdasarkan kekuatan (setara dengan jeda setelah koma, kalimat, atau paragraf), atau Anda dapat mengaturnya ke jangka waktu tertentu dalam hitungan detik atau milidetik. Jika Anda tidak menentukan atribut untuk menentukan panjang jeda, Amazon Polly menggunakan default, < break strength="medium"/> yaitu, yang menambahkan jeda panjang jeda setelah koma.

strengthnilai atribut:

- none: Tidak ada jeda. Gunakan none untuk menghapus jeda yang biasanya terjadi, seperti setelah periode tertentu.
- x-weak: Memiliki kekuatan yang sama dengannone, tidak ada jeda.
- weak: Menetapkan jeda dengan durasi yang sama dengan jeda setelah koma.
- medium: Memiliki kekuatan yang sama denganweak.
- strong: Menetapkan jeda dengan durasi yang sama dengan jeda setelah kalimat.
- x-strong: Menetapkan jeda dengan durasi yang sama dengan jeda setelah paragraf.

timenilai atribut:

- **[number]**s: Durasi jeda, dalam hitungan detik. Durasi maksimum adalah10s.

- **[number]**ms: Durasi jeda, dalam milidetik. Durasi maksimum adalah 10000ms.

Contoh:

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

Jika Anda tidak menggunakan atribut dengan break tag, hasilnya bervariasi tergantung pada teks:

- Jika tidak ada tanda baca lain di samping break tag, itu akan membuat `< break strength="medium" />` (jeda panjang koma).
- Jika tag berada di sebelah koma, tag akan ditingkatkan menjadi `< break strength="strong" />` (jeda panjang kalimat).
- Jika tag berada di sebelah titik, tag akan ditingkatkan menjadi `< break strength="x-strong" />` (jeda sepanjang paragraf).

Menekankan kata-kata

`< emphasis>`

Tag ini hanya didukung oleh format TTS standar.

Untuk menekankan kata-kata, gunakan `< emphasis>` tag. Menekankan kata-kata mengubah kecepatan dan volume berbicara. Penekanan yang lebih besar membuat Amazon Polly berbicara teks lebih keras dan lebih lambat. Kurang penekanan membuatnya berbicara lebih tenang dan lebih cepat. Untuk menentukan tingkat penekanan, gunakan `level` atribut.

`level` nilai atribut:

- **Strong:** Meningkatkan volume dan memperlambat kecepatan berbicara sehingga pidato lebih keras dan lebih lambat.
- **Moderate:** Meningkatkan volume dan memperlambat kecepatan berbicara, tetapi kurang dari **strong**. **Moderate** adalah default.
- **Reduced:** Mengurangi volume dan mempercepat kecepatan berbicara. Pidato lebih lembut dan lebih cepat.

 Note

Tingkat dan volume berbicara normal untuk suara jatuh di antara reduced level moderate dan.

Contoh:

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

Menentukan bahasa lain untuk kata-kata tertentu

<lang>

Tag ini didukung oleh format TTS generatif, bentuk panjang, saraf, dan standar. Untuk suara generatif, <lang> tag hanya dapat digunakan di sekitar kalimat lengkap.

Tentukan bahasa lain untuk kata, frasa, atau kalimat tertentu dengan <lang>tag. Kata-kata dan frasa bahasa asing umumnya diucapkan lebih baik ketika mereka tertutup dalam sepasang <lang> tag. Untuk menentukan bahasa, gunakan `xml:lang` atribut. Untuk daftar lengkap bahasa yang tersedia, lihat [Bahasa di Amazon Polly](#).

Kecuali Anda menerapkan <lang> tag, semua kata dalam teks input diucapkan dalam bahasa suara yang ditentukan dalam `voice-id`. Jika Anda menerapkan <lang> tag, kata-kata diucapkan dalam bahasa itu.

Misalnya, jika `voice-id` adalah Joanna (yang berbicara bahasa Inggris AS), Amazon Polly berbicara yang berikut dengan suara Joanna tanpa aksen Prancis:

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

Jika Anda menggunakan suara Joanna dengan <lang> tag, Amazon Polly mengucapkan kalimat dalam suara Joanna dalam bahasa Prancis beraksen Amerika:

```
<speak>
```

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

Karena Joanna bukan suara asli Prancis, pengucapan didasarkan pada bahasa ibunya, Inggris AS. Misalnya, meskipun pengucapan bahasa Prancis yang sempurna menampilkan getaran uvular /R/ dalam kata français, suara Inggris AS Joanna mengucapkan fonem ini sebagai suara yang sesuai /r/.

Jika Anda menggunakan Giorgio, yang berbicara bahasa Italia, dengan teks berikut, Amazon Polly mengucapkan kalimat dalam suara Giorgio dengan pengucapan Italia: `voice-id`

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

Jika Anda menggunakan suara yang sama dengan `<lang>` tag berikut, Amazon Polly mengucapkan Bruce Springsteen dalam bahasa Inggris beraksen Italia:

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

Tag ini juga dapat digunakan sebagai pengganti [DefaultLangCode](#) opsi opsional saat mensintesis ucapan. Namun, melakukan hal itu mengharuskan Anda memformat teks Anda menggunakan SSML.

Menempatkan tag khusus dalam teks Anda

`<mark>`

Tag ini didukung oleh format TTS bentuk panjang, saraf, dan standar. Tag ini tidak melakukan apa pun untuk suara generatif karena tanda pidato tidak tersedia untuk suara generatif.

Untuk menempatkan tag kustom dalam teks, gunakan `<mark>` tag. Amazon Polly tidak mengambil tindakan pada tag, tetapi mengembalikan lokasi tag dalam metadata SSML. Tag ini dapat berupa apa saja yang ingin Anda panggil, asalkan mempertahankan format berikut:

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

Misalnya, misalkan nama tag adalah “hewan” dan teks inputnya adalah:

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

Amazon Polly mungkin mengembalikan metadata SSML berikut:

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

Menambahkan jeda antar paragraf

<p>

Tag ini didukung oleh format TTS generatif, bentuk panjang, saraf, dan standar.

<p>Untuk menambahkan jeda di antara paragraf dalam teks Anda, gunakan tag. Menggunakan tag ini memberikan jeda yang lebih lama daripada penutur asli biasanya menempatkan pada koma atau akhir kalimat. Gunakan <p>tag untuk melampirkan paragraf:

```
< 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>
```

Ini setara dengan menentukan jeda menggunakan. < break strength="x-strong"/>

Menggunakan pengucapan fonetik

<phoneme>

Tag ini didukung oleh format TTS bentuk panjang, saraf, dan standar.

<phoneme>Untuk membuat Amazon Polly menggunakan pengucapan fonetik untuk teks tertentu, gunakan tag.

Dua atribut diperlukan dengan <phoneme> tag. Mereka menunjukkan alfabet fonetik yang digunakan Amazon Polly dan simbol fonetik dari pengucapan yang dikoreksi:

- alphabet

- **ipa**— Menunjukkan bahwa Alfabet Fonetik Internasional (IPA) akan digunakan.
- **x-sampa**— Menunjukkan bahwa Metode Penilaian Pidato Diperpanjang Alfabet Fonetik (X-SAMPA) akan digunakan.
- **ph**
 - Menentukan simbol fonetik untuk pengucapan. Untuk informasi selengkapnya, lihat [Bahasa di Amazon Polly](#)

Dengan `<phoneme>` tag, Amazon Polly menggunakan pengucapan yang ditentukan oleh `ph` atribut, bukan pengucapan standar yang dikaitkan secara default dengan bahasa yang digunakan oleh suara yang dipilih.

Misalnya, kata “kemiri” dapat diucapkan dengan dua cara. Dalam contoh berikut, “kemiri” diberikan pengucapan yang berbeda di setiap baris. Amazon Polly mengucapkan kemiri seperti yang ditentukan dalam `ph` atribut, alih-alih menggunakan pengucapan default.

Alfabet Fonetik Internasional (IPA)

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

Metode Penilaian Pidato Diperpanjang Alfabet Fonetik (X-SAMPA)

```
<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>
```

Mandarin Mandarin menggunakan Pinyin untuk pengucapan fonetik..

Pinyin

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

Bahasa Jepang menggunakan Yomigana dan Pengucapan Kana.

Yomigana

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

Pengucapan Kana

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

Mengontrol volume, kecepatan berbicara, dan nada

<prosody>

Atribut tag prosody sepenuhnya didukung oleh suara TTS standar. Suara generatif, Neural, dan Bentuk Panjang mendukung rate atribut volume dan, tetapi tidak mendukung atribut. pitch Untuk suara Generatif, tag prosodi hanya dapat digunakan di sekitar kalimat lengkap.

Untuk mengontrol volume, kecepatan, atau nada suara yang Anda pilih, gunakan prosody tag.

Volume, kecepatan bicara, dan nada tergantung pada suara tertentu yang dipilih. Selain perbedaan antara suara untuk bahasa yang berbeda, ada perbedaan antara suara individu yang berbicara dalam bahasa yang sama. Karena itu, meskipun atribut serupa di semua bahasa, ada variasi yang jelas dari bahasa ke bahasa dan tidak ada nilai absolut yang tersedia.

prosodyTag memiliki tiga atribut, masing-masing memiliki beberapa nilai yang tersedia untuk mengatur atribut. Setiap atribut menggunakan sintaks yang sama:

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

- volume
 - default: Mengatur ulang volume ke level default untuk suara saat ini.
 - silent,x-soft,soft, mediumloud,x-loud: Mengatur volume ke nilai yang telah ditentukan untuk suara saat ini.

- +ndB,-ndB: Mengubah volume relatif terhadap level saat ini. Nilai +0dB berarti tidak ada perubahan, +6dB berarti kira-kira dua kali volume saat ini, dan -6dB berarti sekitar setengah volume saat ini.

Misalnya, Anda dapat mengatur volume untuk suatu bagian sebagai berikut:

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

Atau Anda bisa mengaturnya dengan cara ini:

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

- rate
- x-slow,slow,medium,fast,x-fast. Menetapkan pitch ke nilai yang telah ditentukan untuk suara yang dipilih.
- n%: Persentase perubahan non-negatif dalam tingkat berbicara. Misalnya, nilai 100% berarti tidak ada perubahan dalam tingkat berbicara, nilai 200% berarti tingkat berbicara dua kali tingkat default, dan nilai 50% berarti tingkat berbicara setengah dari tingkat default. Nilai ini memiliki kisaran 20-200%.

Misalnya, Anda dapat mengatur kecepatan bicara untuk suatu bagian sebagai berikut:

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

Atau Anda bisa mengaturnya dengan cara ini:

```
<say>
  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**: Mengatur ulang pitch ke level default untuk suara saat ini.
- **x-low,low,medium,high,x-high**: Mengatur pitch ke nilai yang telah ditentukan untuk suara saat ini.
- **+n%atau-n%**: Menyesuaikan pitch dengan persentase relatif. Misalnya, nilai **+0%** berarti tidak ada perubahan nada dasar, **+5%** memberikan nada dasar yang sedikit lebih tinggi, dan **-5%** menghasilkan nada dasar yang sedikit lebih rendah.

Misalnya, Anda dapat mengatur pitch untuk suatu bagian sebagai berikut:

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

Atau Anda bisa mengaturnya dengan cara ini:

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

<prosody> Tag harus berisi setidaknya satu atribut, tetapi dapat menyertakan lebih banyak dalam tag yang sama.

```
<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>
```

Itu juga dapat dikombinasikan dengan tag bersarang, sebagai berikut:

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

```
</speak>
```

Note

Saat `<prosody>` ini sebagian tersedia untuk suara Generatif.

Menetapkan durasi maksimum untuk ucapan yang disintesis

```
<prosody amazon:max-duration>
```

Tag ini saat ini hanya didukung oleh format TTS standar.

Untuk mengontrol berapa lama Anda ingin pidato untuk mengambil ketika itu disintesis, gunakan `<prosody>` tag dengan `amazon:max-duration` atribut.

Durasi ucapan yang disintesis sedikit berbeda, tergantung pada suara yang Anda pilih. Hal ini dapat membuat sulit untuk mencocokkan ucapan yang disintesis dengan visual atau aktivitas lain yang membutuhkan waktu yang tepat. Masalah ini diperbesar untuk aplikasi terjemahan karena waktu yang diperlukan untuk mengucapkan frasa tertentu dapat sangat bervariasi dengan bahasa yang berbeda.

`<prosody amazon:max-duration>` Tag mencocokkan ucapan yang disintesis dengan jumlah waktu yang Anda inginkan (durasi).

Tag ini menggunakan sintaks berikut:

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

Dengan `<prosody amazon:max-duration>` tag, Anda dapat menentukan durasi dalam detik atau milidetik:

- *ns*: durasi maksimum dalam hitungan detik
- *nms*: durasi maksimum dalam milidetik

Misalnya, teks lisan berikut memiliki durasi maksimum 2 detik:

```
<speak>
```

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

Teks ditempatkan di dalam tag, itu tidak melebihi durasi yang ditentukan. Jika suara atau bahasa yang dipilih biasanya memakan waktu lebih lama dari durasi itu, Amazon Polly mempercepat pembicaraan sehingga sesuai dengan durasi yang ditentukan.

Jika durasi yang ditentukan lebih lama dari yang diperlukan untuk membaca teks dengan kecepatan normal, Amazon Polly membaca pidato secara normal. Itu tidak memperlambat pidato atau menambah keheningan, sehingga audio yang dihasilkan lebih pendek dari yang diminta.

Note

Amazon Polly meningkatkan kecepatan tidak lebih dari 5 kali tingkat normal. Jika teks diucapkan lebih cepat dari ini, biasanya tidak masuk akal. Jika pidato tidak sesuai dengan durasi yang Anda tentukan bahkan ketika dipercepat hingga maksimum, audio akan dipercepat tetapi akan bertahan lebih lama dari durasi yang ditentukan.

Anda dapat menyertakan satu kalimat atau beberapa kalimat dalam `<prosody amazon:max-duration>` tag, dan Anda dapat menggunakan beberapa `<prosody amazon:max-duration>` tag dalam teks Anda.

Contoh:

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

```
</prosody>
</speak>
```

Menggunakan `<prosody amazon:max-duration>` tag dapat meningkatkan latensi saat Amazon Polly mengembalikan ucapan yang disintesis. Tingkat latensi tergantung pada bagian dan panjangnya. Sebaiknya gunakan teks yang terdiri dari bagian teks yang relatif pendek.

Batasan

Ada batasan baik dalam cara Anda menggunakan `<prosody amazon:max-duration>` tag dan cara kerjanya dengan tag SSML lainnya:

- Teks di dalam `<prosody amazon:max-duration>` tag tidak boleh lebih dari 1500 karakter.
- Anda tidak dapat membuat `<prosody amazon:max-duration>` tag sarang. Jika Anda menempatkan satu `<prosody amazon:max-duration>` tag di dalam tag lain, Amazon Polly mengabaikan tag batin.

Misalnya, berikut ini, `<prosody amazon:max-duration="5s">` tag diabaikan:

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

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

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

- Anda tidak dapat menggunakan `<prosody>` tag dengan `rate` atribut dalam `<prosody amazon:max-duration>` tag. Ini karena keduanya mempengaruhi kecepatan di mana teks diucapkan.

Dalam contoh berikut, Amazon Polly mengabaikan tag: `<prosody rate="2">`

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

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

Jeda dan **max-duration**

Saat menggunakan `max-duration` tag, Anda masih dapat menyisipkan jeda dalam teks Anda. Namun, Amazon Polly menyertakan panjang jeda saat menghitung durasi maksimum untuk berbicara. Selain itu, Amazon Polly mempertahankan jeda singkat yang terjadi di mana koma dan titik ditempatkan dalam suatu bagian dan termasuk dalam durasi maksimum.

Misalnya, di blok berikut, jeda 600 milidetik dan jeda yang disebabkan oleh koma dan periode terjadi dalam pidato 8 detik:

```
<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>
```

Menambahkan jeda di antara kalimat

`<s>`

Tag ini didukung oleh format TTS generatif, bentuk panjang, saraf, dan standar.

Untuk menambahkan jeda di antara baris atau kalimat dalam teks Anda, gunakan `<s>` tag. Menggunakan tag ini memiliki efek yang sama seperti:

- Mengakhiri kalimat dengan titik (.)
- Menentukan jeda dengan `<break strength="strong"/>`

Berbeda `<break>` dengan tag, `<s>` tag melampirkan kalimat. Ini berguna untuk mensintesis pidato yang diatur dalam baris, bukan kalimat, seperti puisi.

Dalam contoh berikut, `<s>` tag membuat jeda singkat setelah kalimat pertama dan kedua. Kalimat terakhir tidak memiliki `<s>` tag, tetapi juga diikuti oleh jeda singkat karena berakhir dengan titik.

```
<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>
```

Mengontrol bagaimana jenis kata khusus diucapkan

`<say-as>`

`<say-as>` Tag ini didukung oleh mesin TTS generatif, bentuk panjang, saraf, dan standar. Namun, perhatikan bahwa jika Amazon Polly menggunakan suara saraf dan menemukan `<say-as>` tag dengan `characters` opsi saat runtime, kalimat yang terpengaruh akan disintesis menggunakan suara standar terkait. Namun, kalimat yang terpengaruh akan tetap ditagih seolah-olah menggunakan suara saraf.

Gunakan `<say-as>` tag dengan `interpret-as` atribut untuk memberi tahu Amazon Polly cara mengucapkan karakter, kata, dan angka tertentu. Ini memungkinkan Anda memberikan konteks tambahan untuk menghilangkan ambiguitas apa pun tentang bagaimana Amazon Polly harus merender teks.

`<say-as>` Tag menggunakan satu atribut `interpret-as`, yang menggunakan sejumlah kemungkinan nilai yang tersedia. Masing-masing menggunakan sintaks yang sama:

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

Nilai-nilai berikut tersedia dengan `interpret-as`:

- `characters` atau `spell-out`: Mengeja setiap huruf teks, seperti dalam a-b-c.

 Note

Opsi ini saat ini tidak didukung untuk suara saraf. Jika Anda menggunakan suara saraf dan kode SSML ini ditemui oleh Amazon Polly saat run-time, kalimat yang terpengaruh akan disintesis menggunakan suara standar terkait. Harap dicatat, bagaimanapun, bahwa kalimat ini masih akan ditagih seolah-olah menggunakan suara saraf.

- `cardinalataunumber`: Menafsirkan teks numerik sebagai angka kardinal, seperti pada 1.234.
- `ordinal`: Menafsirkan teks numerik sebagai nomor urut, seperti pada 1.234.
- `digits`: Jabarkan setiap digit secara individual, seperti pada 1-2-3-4.
- `fraction`: Menafsirkan teks numerik sebagai pecahan. Ini bekerja untuk kedua fraksi umum seperti 3/20, dan pecahan campuran, seperti 2 ½. Lihat di bawah untuk informasi lebih lanjut.
- `unit`: Menafsirkan teks numerik sebagai pengukuran. Nilai harus berupa angka atau pecahan diikuti oleh unit tanpa spasi di antara seperti di1/2inch, atau hanya dengan unit, seperti dalam1meter.
- `date`: Menafsirkan teks sebagai tanggal. Format tanggal harus ditentukan dengan atribut format. Lihat di bawah untuk informasi lebih lanjut.
- `time`: Menafsirkan teks numerik sebagai durasi, dalam menit dan detik, seperti dalam. 1 ' 21 "
- `address`: Menafsirkan teks sebagai bagian dari alamat jalan.
- `expletive`: "Bunyi bip" konten yang disertakan dalam tag.
- `telephone`: Menafsirkan teks numerik sebagai nomor telepon 7 digit atau 10 digit, seperti dalam. 2025551212 Anda juga dapat menggunakan nilai ini untuk menangani ekstensi telepon, seperti dalam2025551212x345. Lihat di bawah untuk informasi lebih lanjut.

 Note

Saat ini `telephone` opsi tidak tersedia untuk semua bahasa. Namun, tersedia untuk suara yang berbicara varian bahasa Inggris (en-AU, en-GB, en-in, en-US, dan), varian bahasa Spanyol (es-ES, es-MX, dan es-US en-GB-WLS), varian bahasa Prancis (fr-fr dan fr-CA), dan varian Portugis (Pt-BR dan Pt-PT), serta Jerman (de-de), Italia (IT-it), Jepang (Ja-jp), dan Rusia (Ru-ru). Perlu juga dicatat bahwa dalam beberapa kasus, bahasa seperti Arab (arb) secara otomatis menangani nomor yang ditetapkan sebagai nomor telepon sehingga tidak benar-benar menerapkan tag `telephone` SSML.

Pecahan

Amazon Polly menafsirkan nilai dalam `say-as` tag yang memiliki `interpret-as="fraction"` atribut sebagai pecahan umum. Berikut ini adalah sintaks untuk pecahan:

- Pecahan

Sintaks: *cardinal number*/*cardinal number*, seperti 2/9.

Misalnya: `<say-as interpret-as="fraction">2/9</say-as>` diucapkan “dua kesembilan.”

- Nomor Campuran Non-Negatif

Sintaks: *cardinal number*+*cardinal number*/*cardinal number*, seperti 3+1/2.

Misalnya, `<say-as interpret-as="fraction">3+1/2</say-as>` diucapkan “tiga setengah.”

Note

Harus ada + antara “3” dan “1/2”. Amazon Polly tidak mendukung angka campuran tanpa+, seperti “3 1/2”.

Tanggal

Kapan `interpret-as` diatur ke `date`, Anda juga perlu menunjukkan format tanggal.

Ini menggunakan sintaks berikut:

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

Contoh:

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

Format berikut dapat digunakan dengan `date` atribut.

- `mdy`: Month-day-year.

- dmy: Day-month-year.
- ymd: Year-month-day.
- md: Bulan-hari.
- dm: Hari-bulan.
- ym: Tahun-bulan.
- my: Bulan-tahun.
- d: Hari.
- m: Bulan.
- y: Tahun.
- yyymmdd: Year-month-day. Jika Anda menggunakan format ini, Anda dapat membuat Amazon Polly melewati sebagian tanggal menggunakan tanda tanya.

Misalnya, Amazon Polly merender yang berikut ini sebagai “22 September”:

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

Format tidak diperlukan.

Telepon

Amazon Polly mencoba menafsirkan teks yang Anda berikan dengan benar berdasarkan pemformatan teks bahkan tanpa tag. `<say-as>` Misalnya, jika teks Anda menyertakan “202-555-1212,” Amazon Polly menafsirkannya sebagai nomor telepon 10 digit dan mengatakan setiap digit satu per satu, dengan jeda singkat untuk setiap tanda hubung. Dalam hal ini, Anda tidak perlu menggunakannya `<say-as interpret-as="telephone">`. Namun, jika Anda memberikan teks “2025551212” dan ingin Amazon Polly mengatakannya sebagai nomor telepon, Anda akan menentukan. `<say-as interpret-as="telephone">`

Logika untuk menafsirkan setiap elemen adalah spesifik bahasa. Misalnya, bahasa Inggris AS dan Inggris berbeda dalam cara nomor telepon diucapkan (dalam bahasa Inggris Inggris, urutan digit yang sama dikelompokkan bersama, seperti dalam “lima ganda” atau “tiga empat”). Untuk melihat perbedaannya, uji contoh berikut dengan suara AS dan dengan suara Inggris:

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

Mengucapkan akronim dan singkatan

<sub>

Tag ini didukung oleh format TTS generatif, bentuk panjang, saraf, dan standar.

Gunakan <sub> tag dengan alias atribut untuk menggantikan kata yang berbeda (atau pengucapan) untuk teks yang dipilih seperti akronim atau singkatan.

Ini menggunakan sintaks:

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

Dalam contoh berikut, nama “Merkurius” diganti dengan simbol kimia elemen untuk membuat konten audio lebih jelas.

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

Meningkatkan pengucapan dengan menentukan bagian pidato

<w>

Tag ini didukung oleh format TTS generatif, bentuk panjang, saraf, dan standar.

Anda dapat menggunakan <w>tag untuk menyesuaikan pengucapan kata-kata dengan menentukan bagian kata dari ucapan atau makna alternatif. Ini dilakukan dengan menggunakan role atribut.

Tag ini menggunakan sintaks berikut:

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

Nilai-nilai berikut dapat digunakan untuk role atribut:

Untuk menentukan bagian pidato:

- amazon:VB: menafsirkan kata sebagai kata kerja (present simple).

- `amazon:VBD`: menafsirkan kata sebagai kata kerja past tense.
- `amazon:DT`: menafsirkan kata sebagai penentu.
- `amazon:IN`: menafsirkan kata sebagai preposisi.
- `amazon:JJ`: menafsirkan kata sebagai kata sifat.
- `amazon:NN`: menafsirkan kata sebagai kata benda.

Misalnya, tergantung pada bagian pidatonya, pengucapan bahasa Inggris AS dari kata “baca” bervariasi berdasarkan tag:

```
<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>.
</say-as>
```

Untuk menentukan arti tertentu:

- `amazon:DEFAULT`: menggunakan arti kata default.
- `amazon:SENSE_1`: menggunakan arti kata non-default saat hadir. Misalnya, kata benda “bass” diucapkan secara berbeda tergantung pada artinya. Arti default adalah bagian terendah dari rentang musik. Arti alternatifnya adalah spesies ikan air tawar, juga disebut “bass” tetapi diucapkan berbeda. Menggunakan `<w role="amazon:SENSE_1">bass</w>` render pengucapan non-default (freshwater fish) untuk teks audio.

Perbedaan pengucapan dan makna ini dapat didengar jika Anda mensintesis yang berikut:

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

Note

Beberapa bahasa mungkin memiliki pilihan yang berbeda dari bagian pidato yang didukung.

Menambahkan suara pernapasan

dan `<amazon:breath><amazon:auto-breaths>`

Tag ini hanya didukung oleh format TTS standar.

Pidato yang terdengar alami mencakup kata-kata yang diucapkan dengan benar dan suara pernapasan. Dengan menambahkan suara pernapasan ke ucapan yang disintesis, Anda dapat membuatnya terdengar lebih alami. `<amazon:auto-breaths>` Tag `<amazon:breath>` dan memberikan napas. Anda memiliki opsi berikut:

- Mode manual: Anda mengatur lokasi, panjang, dan volume suara napas dalam teks
- Mode otomatis: Amazon Polly secara otomatis memasukkan suara pernapasan ke output ucapan
- Mode campuran: Anda dan Amazon Polly menambahkan suara pernapasan

Mode Manual

Dalam mode manual, Anda menempatkan `<amazon:breath/>` tag di teks input di mana Anda ingin mencari nafas. Anda dapat menyesuaikan panjang dan volume napas dengan `volume` atribut `duration` dan masing-masing:

- `duration`: Mengontrol panjang nafas. Nilai yang valid adalah: `default,x-short,short,medium,long,x-long`. Nilai default-nya adalah `medium`.
- `volume`: Mengontrol seberapa keras suara pernapasan. Nilai yang valid adalah: `default,x-soft,soft,medium,loud,x-loud`. Nilai default-nya adalah `medium`.

Note

Panjang dan volume yang tepat dari setiap nilai atribut bergantung pada suara Amazon Polly tertentu yang digunakan.

Untuk mengatur suara napas menggunakan default, gunakan `<amazon:breath/>` tanpa atribut.

Misalnya, untuk menggunakan atribut untuk mengatur durasi dan volume untuk napas ke medium, Anda akan mengatur atribut sebagai berikut:

```
<speak>
```

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

Untuk menggunakan default, Anda hanya akan menggunakan tag:

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

Anda dapat menambahkan suara pernapasan individu dalam suatu bagian, sebagai berikut:

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

Mode Otomatis

Dalam mode otomatis, Anda menggunakan `<amazon:auto-breaths>` tag untuk memberi tahu Amazon Polly agar secara otomatis membuat suara pernapasan pada interval yang sesuai. Anda dapat mengatur frekuensi interval, volumenya, dan durasinya. Tempatkan `</amazon:auto-breaths>` tag di awal teks yang ingin Anda terapkan pernapasan otomatis dan kemudian tutup tag di bagian akhir.

Note

Berbeda dengan tag mode manual `<amazon:breath/>`, `<amazon:auto-breaths>` tag memerlukan tag penutup (`</amazon:auto-breaths>`).

Anda dapat menggunakan atribut opsional berikut dengan `<amazon:auto-breaths>` tag:

- **volume:** Mengontrol seberapa keras suara pernapasan. Nilai yang valid adalah: `default,x-soft,soft,medium,loud,x-loud`. Nilai default-nya adalah `medium`.

- **frequency:** Mengontrol seberapa sering suara pernapasan terjadi dalam teks. Nilai yang valid adalah: default, x-low, low, medium, high, x-high. Nilai default-nya adalah medium.
- **duration:** Mengontrol panjang nafas. Nilai yang valid adalah: default, x-short, short, medium, long, x-long. Nilai default-nya adalah medium.

Secara default, frekuensi suara pernapasan tergantung pada teks input. Namun, suara pernapasan sering terjadi setelah koma dan menstruasi.

Contoh berikut menunjukkan cara menggunakan `<amazon:auto-breaths>` tag. Untuk memutuskan opsi mana yang akan digunakan untuk konten Anda, salin contoh yang berlaku ke konsol Amazon Polly dan dengarkan perbedaannya.

- Menggunakan mode otomatis tanpa parameter opsional.

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

- Menggunakan mode otomatis dengan kontrol volume. Parameter yang tidak ditentukan (`duration` dan `frequency`) diatur ke nilai default (medium).

```
<speak>
  <amazon:auto-breaths volume="x-soft">Amazon Polly is a service that turns text
into lifelike
  speech, allowing you to create applications that talk and build entirely new
categories of
  speech-enabled products. Amazon Polly is a text-to-speech service, that uses
advanced deep
  learning technologies to synthesize speech that sounds like a human voice. With
dozens of
```

```

    lifelike voices across a variety of languages, you can select the ideal voice
    and build speech-
    enabled applications that work in many different countries.</amazon:auto-
    breaths>
</speak>

```

- Menggunakan mode otomatis dengan kontrol frekuensi. Parameter yang tidak ditentukan (`duration` dan `volume`) diatur ke nilai default (`medium`).

```

<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>

```

- Menggunakan mode otomatis dengan beberapa parameter. Untuk `Duration` parameter yang tidak ditentukan, Amazon Polly menggunakan nilai `medium` default ().

```

<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>

```

Gaya berbicara penyiar berita

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

Gaya penyiar berita hanya tersedia untuk suara Matthew atau Joanna, yang hanya tersedia dalam bahasa Inggris Amerika (en-US), Lupe, dalam bahasa Spanyol AS (es-AS) dan Amy, dalam bahasa Inggris Inggris (en-GB). Ini hanya didukung saat menggunakan Neural format.

Untuk menggunakan gaya penyiar berita, Anda menggunakan tag SSML dan sintaks berikut:

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

Misalnya, Anda dapat menggunakan gaya penyiar berita dengan suara Amy sebagai berikut:

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

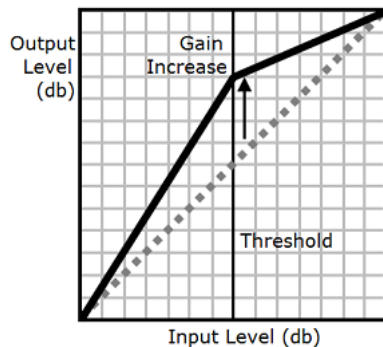
Menambahkan kompresi rentang dinamis

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

Tag ini didukung oleh format TTS bentuk panjang, saraf, dan standar.

Tergantung pada teks, bahasa, dan suara yang digunakan dalam file audio, suara berkisar dari lembut hingga keras. Suara lingkungan, seperti suara kendaraan yang bergerak, seringkali dapat menutupi suara yang lebih lembut, yang membuat trek audio sulit didengar dengan jelas. Untuk meningkatkan volume suara tertentu dalam file audio Anda, gunakan tag kompresi rentang dinamis (drc).

drcTag menetapkan ambang batas “kenyaringan” midrange untuk audio Anda, dan meningkatkan volume (penguatan) suara di sekitar ambang batas itu. Ini menerapkan kenaikan keuntungan terbesar yang paling dekat dengan ambang batas, dan kenaikan kenaikan berkurang lebih jauh dari ambang batas.



Ini membuat suara kelas menengah lebih mudah didengar di lingkungan yang bising, yang membuat seluruh file audio lebih jelas.

drcTag adalah parameter Boolean (baik ada atau tidak). Ini menggunakan sintaks: `<amazon:effect name="drc">` dan ditutup dengan `</amazon:effect>`.

Anda dapat menggunakan **drc** tag dengan suara atau bahasa apa pun yang didukung oleh Amazon Polly. Anda dapat menerapkannya ke seluruh bagian rekaman, atau hanya untuk beberapa kata.

Contoh:

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

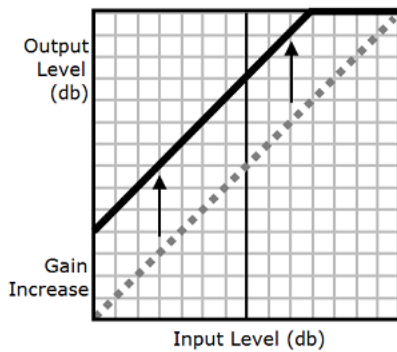
Note

Ketika Anda menggunakan "drc" dalam `amazon:effect` sintaks, itu peka huruf besar/kecil.

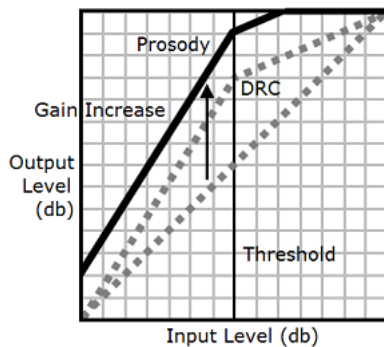
Menggunakan **drc** dengan **prosody volume** Tag

Seperti yang ditunjukkan grafik berikut, **prosody volume** tag secara merata meningkatkan volume seluruh file audio dari level asli (garis putus-putus) ke tingkat yang disesuaikan (garis padat). Untuk

lebih meningkatkan volume bagian tertentu dari file, gunakan `drc` tag dengan `prosody volume` tag. Menggabungkan tag tidak memengaruhi pengaturan `prosody volume` tag.



Saat Anda menggunakan `prosody volume` tag `drc` dan bersama-sama, Amazon Polly menerapkan `drc` tag terlebih dahulu, meningkatkan suara kisaran menengah (yang mendekati ambang batas). Kemudian menerapkan `prosody volume` tag dan selanjutnya meningkatkan volume seluruh trek audio secara merata.



Untuk menggunakan tag bersama-sama, sarang satu di dalam yang lain. Contoh:

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

Dalam teks ini, `prosody volume` tag meningkatkan volume seluruh bagian menjadi “keras.” `drc` tag meningkatkan volume nilai kisaran menengah di kalimat kedua.

 Note

Saat menggunakan `prosody volume` tag `drc` dan bersama-sama, gunakan praktik XHTML standar untuk tag bersarang.

Berbicara dengan lembut

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

Tag ini saat ini hanya didukung oleh format TTS standar.

Untuk menentukan bahwa teks input harus diucapkan dengan `softer-than-normal` suara, gunakan `<amazon:effect phonation="soft">`tag.

Ini menggunakan sintaks:

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

Misalnya, Anda dapat menggunakan tag ini dengan suara Matius sebagai berikut:

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

Mengontrol timbre

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

Tag ini saat ini hanya didukung oleh format TTS standar.

Timbre adalah kualitas nada suara yang membantu Anda membedakan antara suara, bahkan ketika mereka memiliki nada dan kenyaringan yang sama. Salah satu fitur fisiologis terpenting yang berkontribusi pada timbre bicara adalah panjang saluran vokal. Saluran vokal adalah rongga udara yang membentang dari atas pita suara hingga ke tepi bibir.

Untuk mengontrol timbre ucapan keluaran di Amazon Polly, gunakan `vocal-tract-length` tag. Tag ini memiliki efek mengubah panjang saluran vokal pembicara, yang terdengar seperti perubahan ukuran speaker. Ketika Anda meningkatkan `vocal-tract-length`, speaker terdengar lebih besar secara fisik. Saat Anda mengurangnya, speaker terdengar lebih kecil. Anda dapat menggunakan tag ini dengan salah satu suara dalam portofolio Amazon Text-to-Speech Polly.

Untuk mengubah timbre, gunakan nilai berikut:

- `+n%` atau `-n%`: Menyesuaikan panjang saluran vokal dengan persentase perubahan relatif pada suara saat ini. Misalnya, `+4%` atau `-2%`. Nilai yang valid berkisar dari `+100%` hingga `-50%`. Nilai di luar rentang ini terpotong. Misalnya, `+111%` terdengar seperti `+100%` dan `-60%` terdengar seperti `-50%`.
- `n%`: Mengubah panjang saluran vokal menjadi persentase absolut dari panjang saluran suara saat ini. Misalnya, `110%` atau `75%`. Nilai absolut `110%` setara dengan nilai relatif `+10%`. Nilai absolut `100%` sama dengan nilai default untuk suara saat ini.

Contoh berikut menunjukkan cara mengubah panjang saluran vokal untuk mengubah timbre:

```
<speak>
  This is my original voice, without any modifications. <amazon:effect vocal-tract-
length="+15%">
  Now, imagine that I am much bigger. </amazon:effect> <amazon:effect vocal-tract-
length="-15%">
  Or, perhaps you prefer my voice when I'm very small. </amazon:effect> You can also
control the
  timbre of my voice by making minor adjustments. <amazon:effect vocal-tract-
length="+10%">
  For example, by making me sound just a little bigger. </
amazon:effect><amazon:effect
  vocal-tract-length="-10%"> Or, making me sound only somewhat smaller. </
amazon:effect>
</speak>
```

Menggabungkan Beberapa Tag

Anda dapat menggabungkan `vocal-tract-length` tag dengan tag SSML lain yang didukung oleh Amazon Polly. Karena timbre (panjang saluran vokal) dan nada terhubung erat, Anda mungkin mendapatkan hasil terbaik dengan menggunakan tag `vocal-tract-length` dan tag `<prosody pitch>`. Untuk menghasilkan suara yang paling realistis, kami sarankan Anda menggunakan

persentase perubahan yang berbeda untuk kedua tag. Bereksperimenlah dengan berbagai kombinasi untuk mendapatkan hasil yang Anda inginkan.

Contoh berikut menunjukkan bagaimana menggabungkan tag.

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

Berbisik

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

Tag ini saat ini hanya didukung oleh format TTS standar.

Tag ini menunjukkan bahwa teks input harus diucapkan dengan suara berbisik daripada sebagai ucapan normal. Ini dapat digunakan dengan salah satu suara dalam portofolio Amazon Text-to-Speech Polly.

Ini menggunakan sintaks berikut:

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

Contoh:

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

Dalam hal ini, pidato yang disintesis yang diucapkan oleh karakter dibisikkan, tetapi frasa “katanya” diucapkan dalam pidato sintesis normal dari suara Amazon Polly yang dipilih.

Anda dapat meningkatkan efek “berbisik” dengan memperlambat tingkat prosodi hingga 10%, tergantung pada efek yang Anda inginkan.

Contoh:

```
<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>
```

Saat membuat tanda ucapan untuk suara berbisik, aliran audio juga harus menyertakan suara yang dibisikkan untuk memastikan bahwa tanda ucapan cocok dengan aliran audio.

Mengelola leksikon

Leksikon pengucapan memungkinkan Anda untuk menyesuaikan pengucapan kata-kata. Amazon Polly menyediakan operasi API yang dapat Anda gunakan untuk menyimpan leksikon di suatu wilayah. AWS Leksikon tersebut kemudian khusus untuk wilayah tertentu. Anda dapat menggunakan satu atau lebih leksikon dari wilayah itu saat mensintesis teks dengan menggunakan operasi `SynthesizeSpeech`. Ini menerapkan leksikon yang ditentukan ke teks input sebelum sintesis dimulai. Untuk informasi selengkapnya, lihat [SynthesizeSpeech](#).

Note

Leksikon ini harus sesuai dengan rekomendasi W3C Spesifikasi Leksikon Pengucapan (PLS). Untuk informasi lebih lanjut, lihat [Spesifikasi Leksikon Pengucapan \(PLS\) Versi 1.0 di situs web W3C](#).

Berikut ini adalah contoh cara menggunakan leksikon dengan mesin sintesis ucapan:

- Kata-kata umum kadang-kadang bergaya dengan angka menggantikan huruf, seperti dengan “g3t sm4rt” (menjadi pintar). Manusia dapat membaca kata-kata ini dengan benar. Namun, mesin Text-to-Speech (TTS) membaca teks secara harfiah, mengucapkan nama persis seperti yang dieja. Di sinilah Anda dapat memanfaatkan leksikon untuk menyesuaikan ucapan yang disintesis dengan menggunakan Amazon Polly. Dalam contoh ini, Anda dapat menentukan alias (get smart) untuk kata “g3t sm4rt” dalam leksikon.
- Teks Anda mungkin termasuk akronim, seperti W3C. Anda dapat menggunakan leksikon untuk mendefinisikan alias untuk kata W3C sehingga dibaca dalam bentuk lengkap yang diperluas (World Wide Web Consortium).

Leksikon memberi Anda kontrol tambahan atas bagaimana Amazon Polly mengucapkan kata-kata yang tidak umum untuk bahasa yang dipilih. Misalnya, Anda dapat menentukan pengucapan menggunakan alfabet fonetik. Untuk informasi lebih lanjut, lihat [Spesifikasi Leksikon Pengucapan \(PLS\) Versi 1.0 di situs web W3C](#).

Topik

- [Menggunakan beberapa leksikon](#)
- [Mengunggah leksikon](#)

- [Menerapkan leksikon \(Synthesizing Speech\)](#)
- [Memfilter daftar leksikon di konsol](#)
- [Mengunduh leksikon di konsol](#)
- [Menghapus leksikon](#)

Menggunakan beberapa leksikon

Anda dapat menerapkan hingga lima leksikon ke teks Anda. Jika grafem yang sama muncul di lebih dari satu leksikon yang Anda terapkan pada teks Anda, urutan penerapannya dapat membuat perbedaan dalam pidato yang dihasilkan. Misalnya, diberikan teks berikut, “Halo, nama saya Bob.” dan dua leksem dalam leksikon berbeda yang keduanya menggunakan grafem. Bob

LeXA

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

LexB

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

Jika leksikon tercantum dalam urutan LexA dan kemudian LexB, pidato yang disintesis adalah “Halo, nama saya Robert.” Jika mereka terdaftar dalam urutan LexB dan kemudian LexA, pidato yang disintesis adalah “Halo, nama saya Bobby.”

Example— Menerapkan LexA Sebelum LexB

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

Output pidato: “Halo, nama saya Robert.”

Example— Menerapkan LexB sebelum LexA

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

Output pidato: “Halo, nama saya Bobby.”

Untuk informasi tentang menerapkan leksikon menggunakan konsol Amazon Polly, lihat [Menerapkan leksikon \(Synthesizing Speech\)](#)

Mengunggah leksikon

Leksikon yang Anda gunakan harus sesuai dengan rekomendasi W3C Spesifikasi Leksikon Pengucapan (PLS). Untuk informasi lebih lanjut, lihat [Spesifikasi Leksikon Pengucapan \(PLS\) Versi 1.0 di situs web](#) W3C.

Console - Lexicons tab

Untuk menggunakan leksikon pengucapan, Anda harus mengunggahnya terlebih dahulu. Ada dua lokasi di konsol tempat Anda dapat mengunggah leksikon, tab Text-to-Speech dan tab Lexicons.

Proses berikut menjelaskan cara menambahkan leksikon yang dapat Anda gunakan untuk menyesuaikan bagaimana kata dan frasa yang tidak umum untuk bahasa yang dipilih diucapkan.

Untuk menambahkan leksikon dari tab Lexicons

1. Masuk ke AWS Management Console dan buka konsol Amazon Polly di <https://console.aws.amazon.com/polly/>
2. Pilih tab Lexicons.
3. Pilih Unggah leksikon.
4. Berikan nama untuk leksikon dan kemudian gunakan Pilih file leksikon untuk menemukan leksikon yang akan diunggah. Anda hanya dapat mengunggah file PLS dengan ekstensi.pls atau.xl.

5. Pilih Unggah leksikon. Jika leksikon dengan nama yang sama (apakah file.pls atau.xl) sudah ada, mengunggah leksikon akan menimpa leksikon yang ada.

Console - TTS tab

Untuk menambahkan leksikon dari tab text-to-Speech

1. Masuk ke AWS Management Console dan buka konsol Amazon Polly di <https://console.aws.amazon.com/polly/>
2. Pilih tab Text-to-Speech.
3. Perluas Pengaturan tambahan, aktifkan Sesuaikan pengucapan, lalu pilih Unggah leksikon.
4. Berikan nama untuk leksikon dan kemudian gunakan Pilih file leksikon untuk menemukan leksikon yang akan diunggah. Anda hanya dapat menggunakan file PLS dengan ekstensi.pls atau.xl.
5. Pilih Unggah leksikon. Jika leksikon dengan nama yang sama (apakah file.pls atau.xl) sudah ada, mengunggah leksikon akan menimpa leksikon yang ada.

AWS CLI - one lexeme

Dengan Amazon Polly, Anda dapat menggunakan [PutLexicon](#) untuk menyimpan leksikon pengucapan di AWS Wilayah tertentu untuk akun Anda. Kemudian, Anda dapat menentukan satu atau lebih leksikon yang disimpan ini dalam [SynthesizeSpeech](#) permintaan Anda yang ingin Anda terapkan sebelum layanan mulai mensintesis teks. Untuk informasi selengkapnya, lihat [Mengelola leksikon](#).

Pertimbangkan leksikon yang sesuai dengan PLS W3C berikut.

```
<?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>
```

Perhatikan hal berikut:

- Dua atribut yang ditentukan dalam `<lexicon>` elemen:
 - `xml:langAtribut` menentukan kode bahasa, `en-US`, yang leksikon berlaku. Amazon Polly dapat menggunakan leksikon contoh ini jika suara yang Anda tentukan dalam `SynthesizeSpeech` panggilan memiliki kode bahasa yang sama (`en-US`).

 Note

Anda dapat menggunakan `DescribeVoices` operasi untuk menemukan kode bahasa yang terkait dengan suara.

- `alphabetAtribut` menentukan IPA, yang berarti bahwa alfabet Alfabet Fonetik Internasional (IPA) digunakan untuk pengucapan. IPA adalah salah satu huruf untuk menulis pengucapan. Amazon Polly juga mendukung Extended Speech Assessment Methods Phonetic Alphabet (X-SAMPA).
- `<lexeme>` Elemen menggambarkan pemetaan antara `<grapheme>` (yaitu, representasi tekstual dari kata) dan `<alias>`

Untuk menguji leksikon ini, lakukan hal berikut:

1. Simpan leksikon sebagai `example.pls`.
2. Jalankan `put-lexicon` AWS CLI perintah untuk menyimpan leksikon (dengan nama `w3c`), di wilayah `us-east-2`.

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

3. Jalankan `synthesize-speech` perintah untuk mensintesis teks sampel ke aliran audio (`speech.mp3`), dan tentukan `lexicon-name` parameter opsional.

```
aws polly synthesize-speech \
--text 'W3C is a Consortium' \
--voice-id Joanna \
--output-format mp3 \
--lexicon-names="w3c" \
speech.mp3
```

4. Mainkan hasilnya `speech.mp3`, dan perhatikan bahwa kata W3C dalam teks digantikan oleh World Wide Web Consortium.

Contoh leksikon sebelumnya menggunakan alias. Alfabet IPA yang disebutkan dalam leksikon tidak digunakan. Leksikon berikut menentukan pengucapan fonetik menggunakan `<phoneme>` elemen dengan alfabet IPA.

```
<?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>
```

Ikuti langkah yang sama untuk menguji leksikon ini. Pastikan Anda menentukan teks input yang memiliki kata “pecan” (misalnya, “Pecan pie is delicious”).

Lihat sumber daya berikut untuk contoh kode tambahan untuk operasi PutLexicon API:

- Contoh Java: [PutLexicon](#)
- Python (Boto3) Contoh: [PutLexicon](#)

AWS CLI - multiple lexemes

Dengan Amazon Polly, Anda dapat menggunakan [PutLexicon](#) untuk menyimpan leksikon pengucapan di AWS Wilayah tertentu untuk akun Anda. Kemudian, Anda dapat menentukan satu atau lebih leksikon yang disimpan ini dalam [SynthesizeSpeech](#) permintaan Anda yang ingin Anda terapkan sebelum layanan mulai mensintesis teks. Untuk informasi selengkapnya, lihat [Mengelola leksikon](#).

Dalam contoh ini, leksem yang Anda tentukan dalam leksikon berlaku secara eksklusif untuk teks input untuk sintesis. Pertimbangkan leksikon berikut:

```
<?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>
```

Leksikon menentukan tiga leksem, dua di antaranya mendefinisikan alias untuk grafem W3C sebagai berikut:

- Elemen `<lexeme>` pertama mendefinisikan alias (World Wide Web Consortium).
- Yang kedua `<lexeme>` mendefinisikan alias alternatif (WWW Consortium).

Amazon Polly menggunakan pengganti pertama untuk setiap grafem tertentu dalam leksikon.

Yang ketiga <lexeme> mendefinisikan pengganti (Komunitas) untuk kata Konsorsium.

Pertama, mari kita uji leksikon ini. Misalkan Anda ingin mensintesis teks sampel berikut ke file audio (speech.mp3), dan Anda menentukan leksikon dalam panggilan ke. SynthesizeSpeech

```
The W3C is a Consortium
```

SynthesizeSpeech pertama menerapkan leksikon sebagai berikut:

- Sesuai leksem pertama, kata W3C direvisi menjadi World Wide Web Consortium. Teks yang direvisi muncul sebagai berikut:

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

- Alias yang didefinisikan dalam leksem ketiga hanya berlaku untuk kata Konsorsium yang merupakan bagian dari teks asli, menghasilkan teks berikut:

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

Anda dapat menguji ini menggunakan AWS CLI sebagai berikut:

1. Simpan leksikon sebaga `example.pls`.
2. Jalankan `put-lexicon` perintah untuk menyimpan leksikon dengan nama `w3c` di wilayah `us-east-2`.

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

3. Jalankan `list-lexicons` perintah untuk memverifikasi bahwa leksikon `w3c` ada dalam daftar leksikon yang dikembalikan.

```
aws polly list-lexicons
```

4. Jalankan `synthesize-speech` perintah untuk mensintesis teks sampel ke file audio (speech.mp3), dan tentukan `lexicon-name` parameter opsional.

```
aws polly synthesize-speech \  
--text 'W3C is a Consortium' \  
--lexicon-name w3c
```

```
--voice-id Joanna \  
--output-format mp3 \  
--lexicon-names="w3c" \  
speech.mp3
```

5. Putar `speech.mp3` file yang dihasilkan untuk memverifikasi bahwa ucapan yang disintesis mencerminkan perubahan teks.

Lihat sumber daya berikut untuk contoh kode tambahan untuk operasi PutLexicon API:

- Contoh Java: [PutLexicon](#)
- Python (Boto3) Contoh: [PutLexicon](#)

Menerapkan leksikon (Synthesizing Speech)

Leksikon yang Anda gunakan harus sesuai dengan rekomendasi W3C Spesifikasi Leksikon Pengucapan (PLS). Untuk informasi lebih lanjut, lihat [Spesifikasi Leksikon Pengucapan \(PLS\) Versi 1.0 di situs web](#) W3C.

Console

Prosedur berikut menunjukkan bagaimana menerapkan leksikon ke teks input Anda dengan menerapkan leksikon untuk menggantikan “World Wide Web Consortium” untuk “W3C”. `w3c.pls` Jika Anda menerapkan beberapa leksikon ke teks Anda, mereka diterapkan dalam urutan top-down dengan kecocokan pertama lebih diutamakan daripada kecocokan selanjutnya. Leksikon diterapkan ke teks hanya jika bahasa yang ditentukan dalam leksikon sama dengan bahasa yang dipilih.

Anda dapat menerapkan leksikon ke teks biasa atau input SSML.

Example— Menerapkan Leksikon W3C.pls

Untuk membuat leksikon yang Anda perlukan untuk latihan ini, lihat [Mengunggah leksikon](#). Gunakan editor teks biasa untuk membuat leksikon `W3C.pls` yang ditampilkan di bagian atas topik. Ingat di mana Anda menyimpan file ini.

Untuk menerapkan leksikon W3C.pls ke input Anda

Dalam contoh ini kami memperkenalkan leksikon untuk menggantikan “World Wide Web Consortium” untuk “W3C”. Bandingkan hasil latihan ini dengan [Menggunakan SSML di konsol](#) untuk bahasa Inggris AS dan bahasa lain.

1. Masuk ke AWS Management Console dan buka konsol Amazon Polly di. <https://console.aws.amazon.com/polly/>
2. Lakukan salah satu tindakan berikut:
 - Matikan SSML lalu ketik atau tempel teks ini ke dalam kotak input teks.

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

- Nyalakan SSML lalu ketik atau tempel teks ini ke dalam kotak input teks.

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

3. Dari daftar Bahasa, pilih Bahasa Inggris, AS, lalu pilih suara yang ingin Anda gunakan untuk teks ini.
4. Perluas Pengaturan tambahan dan nyalakan Sesuaikan pengucapan.
5. Dari daftar leksikon, pilih W3C (English, US).

Jika W3C (English, US) leksikon tidak terdaftar, pilih Unggah leksikon dan unggah, lalu pilih dari daftar. Untuk membuat leksikon ini, lihat [Mengunggah leksikon](#).

6. Untuk segera mendengarkan pidato, pilih Dengarkan.
7. Untuk menyimpan pidato ke file,
 - a. Pilih Unduh.
 - b. Untuk mengubah ke format file yang berbeda, aktifkan Pengaturan format file ucapan, pilih format file yang Anda inginkan, lalu pilih Unduh.

Ulangi langkah sebelumnya, tetapi pilih bahasa yang berbeda dan perhatikan perbedaan dalam output.

AWS CLI

Dalam panggilan ke `SynthesizeSpeech`, Anda dapat menentukan beberapa leksikon. Dalam hal ini, leksikon pertama yang ditentukan (dalam urutan dari kiri ke kanan) mengesampingkan leksikon sebelumnya.

Pertimbangkan dua leksikon berikut. Perhatikan bahwa setiap leksikon menjelaskan alias yang berbeda untuk grafem `W3C` yang sama.

- Leksikon 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>
```

- Leksikon 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>
```

Misalkan Anda menyimpan leksikon ini sebagai `w3c` dan `w3cAlternate` masing-masing. Jika Anda menentukan leksikon secara berurutan (`w3c` diikuti oleh `w3cAlternate`) dalam `SynthesizeSpeech` panggilan, alias untuk W3C yang ditentukan dalam leksikon pertama lebih diutamakan daripada yang kedua. Untuk menguji leksikon, lakukan hal berikut:

1. Simpan leksikon secara lokal dalam file yang dipanggil `w3c.pls` dan `w3cAlternate.pls`
2. Unggah leksikon ini menggunakan `put-lexicon` AWS CLI perintah.

- Unggah `w3c.pls` leksikon dan simpan sebagai `w3c`.

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

- Unggah `w3cAlternate.pls` leksikon pada layanan sebagai `w3cAlternate`.

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

3. Jalankan `synthesize-speech` perintah untuk mensintesis teks sampel ke aliran audio (`speech.mp3`), dan tentukan kedua leksikon menggunakan parameter `lexicon-name`

```
aws polly synthesize-speech \  
--text 'PLS is a W3C recommendation' \  
--voice-id Joanna \  
--output-format mp3 \  
--lexicon-names '["w3c","w3cAlternative"]' \  
speech.mp3
```

4. Uji hasilnya `speech.mp3`. Itu harus dibaca sebagai berikut:

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

Memfilter daftar leksikon di konsol

Prosedur berikut menjelaskan cara memfilter daftar leksikon sehingga hanya leksikon dari bahasa yang dipilih yang ditampilkan.

Console

Untuk memfilter leksikon yang terdaftar berdasarkan bahasa

1. Masuk ke AWS Management Console dan buka konsol Amazon Polly di. <https://console.aws.amazon.com/polly/>
2. Pilih tab Lexicons.
3. Pilih bahasa apa saja.
4. Dari daftar bahasa, pilih bahasa yang ingin Anda filter.

Daftar ini hanya menampilkan leksikon untuk bahasa yang dipilih.

AWS CLI

Amazon Polly menyediakan operasi [ListLexicons](#) API yang dapat Anda gunakan untuk mendapatkan daftar leksikon pengucapan di akun Anda di Wilayah tertentu. AWS CLI Panggilan berikut mencantumkan leksikon di akun Anda di wilayah us-east-2.

```
aws polly list-lexicons
```

Berikut ini adalah contoh respon, menunjukkan dua leksikon bernama w3c dan tomato. Untuk setiap leksikon, respons mengembalikan metadata seperti kode bahasa yang diterapkan leksikon, jumlah leksem yang ditentukan dalam leksikon, ukuran dalam byte, dan sebagainya. Kode bahasa menjelaskan bahasa dan lokal di mana leksem yang didefinisikan dalam leksikon berlaku.

```
{
  "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"
}
]
```

Sumber daya berikut berisi informasi tambahan untuk ListLexicons operasi:

- Contoh Java: [ListLexicons](#)
- Python (Boto3) Contoh: [ListLexicon](#)

Mengunduh leksikon di konsol

Proses berikut menjelaskan cara mengunduh satu atau lebih leksikon. Anda dapat menambahkan, menghapus, atau memodifikasi entri leksikon dalam file dan kemudian mengunggahnya lagi untuk menjaga leksikon Anda. up-to-date

Console

Untuk mengunduh satu atau lebih leksikon

1. Masuk ke AWS Management Console dan buka konsol Amazon Polly di. <https://console.aws.amazon.com/polly/>
2. Pilih tab Lexicons.
3. Pilih leksikon atau leksikon yang ingin Anda unduh.
 - a. Untuk mengunduh satu leksikon, pilih namanya dari daftar.
 - b. Untuk mengunduh beberapa leksikon sebagai satu file arsip terkompresi, pilih kotak centang di samping setiap entri dalam daftar yang ingin Anda unduh.
4. Pilih Unduh.
5. Buka folder tempat Anda ingin mengunduh leksikon.

6. Pilih Simpan.

AWS CLI

Amazon Polly menyediakan operasi [GetLexicon](#) API untuk mengambil konten leksikon pengucapan yang Anda simpan di akun Anda di wilayah tertentu.

`get-lexicon` AWS CLI Perintah berikut mengambil konten `example` leksikon.

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

Jika Anda belum memiliki leksikon yang disimpan di akun Anda, Anda dapat menggunakan `PutLexicon` operasi untuk menyimpannya. Untuk informasi selengkapnya, lihat [Mengunggah leksikon](#).

Berikut ini adalah contoh respon. Selain konten leksikon, respons mengembalikan metadata, seperti kode bahasa yang diterapkan leksikon, jumlah leksem yang ditentukan dalam leksikon, Nama Sumber Daya Amazon (ARN) sumber daya, dan ukuran leksikon dalam byte. `LastModifiedNilainya` adalah stempel waktu Unix.

```
{  
  "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  
  }  
}
```

Sumber daya berikut berisi contoh kode tambahan untuk `GetLexicon` operasi:

- Contoh Java: [GetLexicon](#)
- Python (Boto3) Contoh: [GetLexicon](#)

Menghapus leksikon

Proses berikut menjelaskan cara menghapus leksikon. Setelah menghapus leksikon, Anda harus menambahkannya kembali sebelum Anda dapat menggunakannya lagi. Anda dapat menghapus satu atau lebih leksikon pada saat yang sama dengan memilih kotak centang di sebelah leksikon individual.

Console

Untuk menghapus leksikon

1. Masuk ke AWS Management Console dan buka konsol Amazon Polly di. <https://console.aws.amazon.com/polly/>
2. Pilih tab Lexicons.
3. Pilih satu atau beberapa leksikon yang ingin Anda hapus dari daftar.
4. Pilih Hapus.
5. Masukkan teks konfirmasi lalu pilih Hapus untuk menghapus leksikon dari Wilayah atau Batal untuk menyimpannya.

AWS CLI

Amazon Polly menyediakan operasi [DeleteLexicon](#) API untuk menghapus leksikon pengucapan dari AWS Wilayah tertentu di akun Anda. Berikut ini AWS CLI menghapus leksikon yang ditentukan.

AWS CLI Contoh berikut diformat untuk Unix, Linux, dan macOS. Untuk Windows, ganti karakter kelanjutan backslash (\) Unix di akhir setiap baris dengan tanda sisipan (^) dan gunakan tanda kutip penuh (") di sekitar teks input dengan tanda kutip tunggal (') untuk tag interior.

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

Sumber daya berikut berisi informasi tambahan untuk DeleteLexicon operasi:

- Contoh Java: [DeleteLexicon](#)
- Python (Boto3) Contoh: [DeleteLexicon](#)

File audio panjang

Untuk membuat file TTS untuk bagian teks yang besar, gunakan fungsionalitas sintesis asinkron Amazon Polly. Ini menggunakan tiga `SpeechSynthesisTask` APIs:

- `StartSpeechSynthesisTask`: memulai tugas sintesis baru.
- `GetSpeechSynthesisTask`: mengembalikan rincian tentang tugas sintesis yang dikirimkan sebelumnya.
- `ListSpeechSynthesisTasks`: mencantumkan semua tugas sintesis yang dikirimkan.

`SynthesizeSpeech` Operasi ini menghasilkan audio dalam waktu nyaris nyata, dengan latensi yang relatif sedikit dalam banyak kasus. Untuk melakukan ini, operasi hanya dapat mensintesis 3000 karakter.

Fitur Sintesis Asinkron Amazon Polly mengatasi tantangan memproses dokumen teks yang lebih besar dengan mengubah cara dokumen disintesis dan dikembalikan. Ketika permintaan sintesis dibuat dengan mengirimkan teks input menggunakan, Amazon Polly mengantri permintaan `StartSpeechSynthesisTask`, dan kemudian memprosesnya secara asinkron di latar belakang segera setelah sumber daya sistem tersedia. Amazon Polly kemudian mengunggah aliran ucapan atau tanda ucapan yang dihasilkan langsung ke bucket Amazon Simple Storage Service (Amazon S3) Anda (wajib), dan memberi tahu Anda tentang ketersediaan file yang telah diselesaikan melalui topik SNS (opsional) Anda.

Dengan cara ini, semua fungsi kecuali pemrosesan waktu nyaris nyata tersedia untuk teks dengan panjang hingga 100.000 karakter yang dapat ditagih (atau 200.000 karakter total).

Untuk mensintesis dokumen menggunakan metode ini, Anda harus memiliki bucket Amazon S3 yang dapat ditulis tempat file audio dapat disimpan. Anda dapat diberi tahu saat audio yang disintesis siap dengan menyediakan pengenal Topik SNS opsional. Ketika tugas sintesis selesai, Amazon Polly akan mempublikasikan pesan tentang topik itu. Pesan ini mungkin juga berisi informasi kesalahan yang berguna dalam kasus di mana tugas sintesis tidak berhasil. Untuk melakukan ini, pastikan bahwa pengguna yang membuat tugas sintesis juga dapat mempublikasikan ke Topik SNS. Lihat [dokumentasi Amazon SNS](#) untuk informasi selengkapnya tentang cara membuat dan berlangganan Topik SNS.

Enkripsi

Anda dapat menyimpan file output dalam bentuk terenkripsi di bucket S3 Anda jika diinginkan. Untuk melakukan ini, Anda mengaktifkan [enkripsi bucket Amazon S3](#), yang menggunakan salah satu cipher blok terkuat yang tersedia, Standar Enkripsi Lanjutan 256-bit (AES-256).

Topik

- [Menyiapkan kebijakan IAM untuk sintesis asinkron](#)
- [Membuat file audio yang panjang](#)

Menyiapkan kebijakan IAM untuk sintesis asinkron

Untuk menggunakan fungsionalitas sintesis asinkron, Anda memerlukan kebijakan IAM yang memungkinkan hal berikut:

- penggunaan operasi Amazon Polly baru
- menulis ke ember output S3
- menerbitkan ke topik SNS status [opsional]

Kebijakan berikut hanya memberikan izin yang diperlukan untuk sintesis asinkron dan dapat dilampirkan ke pengguna IAM.

Membuat file audio yang panjang

Anda dapat menggunakan konsol Amazon Polly untuk membuat pidato panjang menggunakan sintesis asinkron dengan fungsionalitas yang sama seperti yang dapat Anda gunakan dengan AWS CLI. Ini dilakukan dengan menggunakan tab Text-to-Speech seperti sintesis lainnya.

Console

Fungsi sintesis asinkron lainnya juga tersedia melalui konsol. Tab tugas sintesis S3 mencerminkan `ListSpeechSynthesisTasks` fungsionalitas, menampilkan semua tugas yang disimpan ke bucket S3 dan memungkinkan Anda untuk memfilternya jika Anda mau. Mengklik pada satu tugas tertentu menunjukkan detailnya, mencerminkan `GetSpeechSynthesisTask` fungsionalitas.

Untuk mensintesis teks besar menggunakan konsol Amazon Polly

1. Masuk ke AWS Management Console dan buka konsol Amazon Polly di. <https://console.aws.amazon.com/polly/>
2. Pilih tab Text-to-Speech. Pilih Formulir Panjang sebagai mesin jika sesuai.
3. Dengan SSML aktif atau mati, ketik atau tempel teks Anda ke dalam kotak input.
4. Pilih bahasa, wilayah, dan suara untuk teks Anda.
5. Pilih Simpan ke S3.

 Note

Opsi Unduh dan Dengarkan berwarna abu-abu jika panjang teks di atas batas 3.000 karakter untuk SynthesizeSpeech operasi waktu nyata.

6. Konsol membuka formulir sehingga Anda dapat memilih tempat menyimpan file output.
 - a. Isi nama bucket Amazon S3 tujuan.
 - b. Secara opsional, isi kunci awalan output.

 Note

Bucket output S3 harus dapat ditulis.

- c. Jika Anda ingin diberi tahu saat tugas sintesis selesai, berikan pengenalan topik SNS opsional.

 Note

SNS harus terbuka untuk publikasi oleh pengguna konsol saat ini untuk menggunakan opsi ini. Untuk informasi selengkapnya, lihat [Amazon Simple Notification Service \(SNS\)](#)

- d. Pilih Simpan ke S3.

Untuk mengambil informasi tentang tugas sintesis ucapan Anda

1. Di konsol, pilih tab S3 Synthesis Tasks.

2. Tugas ditampilkan dalam urutan tanggal. Untuk memfilter tugas, berdasarkan status, pilih Semua status dan kemudian pilih status yang akan digunakan.
3. Untuk melihat detail tugas tertentu, pilih ID Tugas yang ditautkan.

AWS CLI

Fungsionalitas sintesis asinkron Amazon Polly menggunakan tiga `SpeechSynthesisTask` APIs untuk bekerja dengan sejumlah besar teks:

- `StartSpeechSynthesisTask`: memulai tugas sintesis baru.
- `GetSpeechSynthesisTask`: mengembalikan rincian tentang tugas sintesis yang dikirimkan sebelumnya.
- `ListSpeechSynthesisTasks`: mencantumkan semua tugas sintesis yang dikirimkan.

Mensintesis sejumlah besar teks () **StartSpeechSynthesisTask**

Saat Anda ingin membuat file audio yang lebih besar dari yang dapat Anda buat dengan waktu nyata `SynthesizeSpeech`, gunakan `StartSpeechSynthesisTask` operasi. Selain argumen yang diperlukan untuk `SynthesizeSpeech` operasi, `StartSpeechSynthesisTask` juga memerlukan nama bucket Amazon S3. Dua argumen opsional lainnya juga tersedia: key prefix untuk file output dan ARN untuk Topik SNS jika Anda ingin menerima pemberitahuan status tentang tugas tersebut.

- `OutputS3BucketName`: Nama bucket Amazon S3 tempat sintesis harus diunggah. Bucket ini harus berada di wilayah yang sama dengan layanan Amazon Polly. Selain itu, pengguna IAM yang digunakan untuk melakukan panggilan harus memiliki akses ke bucket. [Diperlukan]
- `OutputS3KeyPrefix`: Awalan kunci untuk file output. Gunakan parameter ini jika Anda ingin menyimpan file ucapan keluaran dalam kunci seperti direktori khusus di bucket Anda. [Opsional]
- `SnsTopicArn`: Topik SNS ARN untuk digunakan jika Anda ingin menerima pemberitahuan tentang status tugas. Topik SNS ini harus berada di wilayah yang sama dengan layanan Amazon Polly. Selain itu, pengguna IAM yang digunakan untuk melakukan panggilan harus memiliki akses ke topik. [Opsional]

Misalnya, contoh berikut dapat digunakan untuk menjalankan `start-speech-synthesis-task` AWS CLI perintah di wilayah AS Timur (Ohio):

AWS CLI Contoh berikut diformat untuk Unix, Linux, dan macOS. Untuk Windows, ganti karakter kelanjutan backslash (\) Unix di akhir setiap baris dengan tanda sisipan (^) dan gunakan tanda kutip penuh (") di sekitar teks input dengan tanda kutip tunggal (') untuk tag interior.

```
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
```

Ini akan menghasilkan respons yang terlihat mirip dengan ini:

```
"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` Operasi mengembalikan beberapa bidang baru:

- `OutputUri`: lokasi file ucapan keluaran Anda.
- `TaskId`: pengidentifikasi unik untuk tugas sintesis ucapan yang dihasilkan oleh Amazon Polly.
- `CreationTime`: stempel waktu saat tugas awalnya dikirimkan.
- `RequestCharacters`: jumlah karakter yang dapat ditagih dalam tugas.
- `TaskStatus`: memberikan informasi tentang status tugas yang diajukan.

Ketika tugas Anda dikirimkan, status awal akan ditampilkan `scheduled`. Saat Amazon Polly mulai memproses tugas, status akan berubah menjadi `inProgress` dan lebih baru, menjadi `completed` atau `failed` Jika tugas gagal, pesan kesalahan akan dikembalikan saat memanggil `GetSpeechSynthesisTask` atau `ListSpeechSynthesisTasks` operasi.

Ketika tugas selesai, file ucapan tersedia di lokasi yang ditentukan dalam `OutputUri`.

Mengambil informasi tentang tugas sintesis ucapan Anda

Anda bisa mendapatkan informasi tentang tugas, seperti kesalahan, status, dan sebagainya, menggunakan `GetSpeechSynthesisTask` operasi. Untuk melakukan ini, Anda perlu `task-id` dikembalikan oleh `StartSpeechSynthesisTask`.

Misalnya, contoh berikut dapat digunakan untuk menjalankan `get-speech-synthesis-task` AWS CLI perintah:

```
aws polly get-speech-synthesis-task \  
--region us-east-2 \  
--endpoint-url "https:// polly.us-east-2.amazonaws.com/" \  
--task-id task identifier
```

Anda juga dapat mencantumkan semua tugas sintesis ucapan yang telah Anda jalankan di wilayah saat ini menggunakan `ListSpeechSynthesisTasks` operasi.

Misalnya, contoh berikut dapat digunakan untuk menjalankan `list-speech-synthesis-tasks` AWS CLI perintah:

```
aws polly list-speech-synthesis-tasks \  
--region us-east-2 \  
--endpoint-url "https:// polly.us-east-2.amazonaws.com/"
```

Kuota di Amazon Polly

Amazon Polly menerapkan kuota untuk lalu lintas pelanggan dengan menolak permintaan yang berlebihan. Kuota default untuk SynthesizeSpeech permintaan dengan suara standar adalah 80 transaksi per detik (tps), dalam satu wilayah, untuk satu Akun AWS. Jika batas tidak meningkat, dan jika Anda menghasilkan 100 SynthesizeSpeech permintaan per detik menggunakan suara standar, 80 permintaan per detik akan berhasil, dan 20 permintaan per detik akan dibatasi oleh Amazon Polly. Permintaan ini akan mengembalikan respons dengan status HTTP 400, dan header respons yang menunjukkan `ThrottlingException`. Amazon Polly juga membatasi lalu lintas ke semua operasi berdasarkan tingkat permintaan.

Contoh batas sintesis ucapan

- Sintesis 24 huruf pertama dari alfabet Inggris satu huruf pada satu waktu. Jika sintesis setiap huruf membutuhkan waktu kurang dari 50 milidetik, dengan batas operasi delapan tps, mensintesis 24 huruf akan memakan waktu setidaknya tiga detik. Selama waktu itu, Anda dapat mensintesis hingga delapan huruf per detik. Permintaan lebih lanjut akan dibatasi. Karena permintaan berlangsung dalam waktu singkat, mereka akan disintesis secara serial tanpa tumpang tindih.
- Sintesis 16 paragraf teks. Jika setiap paragraf disintesis dan diterima sepenuhnya di sisi klien dalam dua detik atau kurang, dengan batas operasi delapan permintaan bersamaan, dibutuhkan setidaknya empat detik untuk mensintesis semua 16 artikel. Di detik pertama, Anda dapat memulai hingga delapan permintaan. Selama permintaan bersamaan, setiap upaya untuk memulai sintesis baru akan dibatasi karena batas konkurensi. Anda dapat mensintesis delapan paragraf yang tersisa setelah dua detik pertama, setelah batch permintaan pertama selesai.

Ingatlah batasan berikut saat menggunakan Amazon Polly.

Topik

- [Wilayah yang didukung](#)
- [Kuota dan laju throttle](#)
- [Leksikon pengucapan](#)
- [SynthesizeSpeech Operasi API](#)
- [SpeechSynthesisTask Operasi API](#)
- [Bahasa Markup Sintesis Pidato \(SSML\)](#)

Wilayah yang didukung

Untuk daftar AWS Wilayah tempat Amazon Polly tersedia, lihat [Titik Akhir dan Kuota Amazon Polly](#) di Referensi Umum Amazon Web Services

- Untuk Wilayah yang mendukung suara generatif, lihat Suara [generatif](#).
- [Untuk Wilayah yang mendukung suara bentuk panjang](#), lihat [Long-form suara](#).
- Untuk Wilayah yang mendukung suara saraf, lihat [the section called “Kompatibilitas fitur dan wilayah”](#) TTS saraf.

Kuota dan laju throttle

Tabel berikut mendefinisikan laju throttle per operasi Amazon Polly. Anda dapat menggunakan AWS Management Console untuk meminta kenaikan kuota untuk kuota yang dapat disesuaikan bila diperlukan.

Operasi	Kuota
Leksikon	
DeleteLexicon	Setiap 2 transaksi per detik (tps) dari operasi ini digabungkan. Letupan maksimum yang diizinkan sebesar 4 tps.
PutLexicon	
GetLexicon	
ListLexicons	
Pidato	
DescribeVoices	80 tps dengan batas burst 100 tps
SynthesizeSpeech	Suara generatif: 8 tps Long-form suara: 8 tps dengan batas burst 10 tps Suara saraf: 8 tps dengan batas burst 10 tps Suara standar: 80 tps dengan batas burst 100 tps

Operasi	Kuota
<code>StartSpeechSynthesisTask</code>	Suara generatif: 1 tps Long-form suara: 1 tps Suara saraf: 10 tps Suara standar: 10 tps dengan batas burst 12 tps
<code>StartSpeechSynthesisStream</code>	Suara generatif: 8 tps
<code>GetSynthesizeSpeechTask</code> dan <code>ListSynthesizeSpeechTask</code>	Maksimum yang diizinkan 10 tps digabungkan

Permintaan bersamaan

Untuk suara generatif, Amazon Polly mendukung hingga 26 permintaan bersamaan. Untuk suara bentuk panjang, Amazon Polly mendukung hingga 26 permintaan bersamaan. Untuk suara saraf, Amazon Polly mendukung 8 tps dengan batas burst 10 tps, hingga 18 permintaan bersamaan. Amazon Polly juga mendukung batasan untuk permintaan bersamaan. Untuk suara standar, Amazon Polly mendukung 80 tps hingga 80 permintaan bersamaan.

Untuk `StartSpeechSynthesisStream`, Amazon Polly mendukung hingga 8 permintaan bersamaan.

Praktik terbaik untuk mengurangi pelambatan

- Coba lagi throttle dengan backoff dan jitter sehingga Anda dapat menyebarkan beban dalam waktu singkat, dan menangani puncak penggunaan yang tidak terduga tanpa mengorbankan ketersediaan. Katalog Kode Sampel AWS sudah dikonfigurasi untuk melakukan ini secara default dalam banyak bahasa pemrograman. Kunjungi [perilaku coba lagi fitur](#) untuk melihat detailnya.
- Gunakan [metrik Amazon Polly](#). Amazon Polly secara otomatis menerbitkan CloudWatch untuk menganalisis penggunaan Anda saat ini dan memperkirakan pertumbuhan penggunaan.

Note

Sebelum meminta kenaikan kuota (jika berlaku), hitung kebutuhan tps Anda mengikuti pedoman di halaman ini. Amazon Polly hanya mengamankan sumber daya komputasi yang diperlukan sesuai dengan permintaan pelanggan untuk menjaga biaya Anda tetap rendah.

Leksikon pengucapan

- Anda dapat menyimpan hingga 100 leksikon per akun.
- Nama leksikon dapat berupa string alfanumerik hingga 20 karakter.
- Setiap leksikon dapat berukuran hingga 40.000 karakter. (Perhatikan bahwa ukuran leksikon mempengaruhi latensi SynthesizeSpeech operasi.)
- Anda dapat menentukan hingga 100 karakter untuk masing-masing <phoneme> atau <alias> penggantian dalam leksikon.

Untuk informasi tentang penggunaan leksikon, lihat [Mengelola leksikon](#).

SynthesizeSpeech Operasi API

Saat memperkirakan penggunaan SynthesizeSpeech, perlu diingat bahwa audio yang diproduksi oleh Amazon Polly, terutama untuk aplikasi interaktif, biasanya membutuhkan waktu setidaknya beberapa detik untuk diputar. Ini mengurangi tingkat permintaan SynthesizeSpeech, bahkan untuk sejumlah besar konsumen bersamaan. Selain itu, Amazon Polly membatasi permintaan berdasarkan jumlah SynthesizeSpeech permintaan bersamaan yang disintesis. Tidak ada pengaturan terpisah untuk permintaan bersamaan. Batas permintaan bersamaan selalu memiliki nilai yang sama dengan jumlah tps yang diizinkan dan diskalakan dengannya.

Contoh aplikasi cerita pendek. Anda dapat menggunakan Amazon Polly untuk membangun aplikasi yang memainkan serangkaian cerita pendek. Dengan aplikasi semacam ini, cerita pertama akan mulai diputar, dan kemudian berikutnya, dan seterusnya, sampai pengguna keluar dari aplikasi. Setiap cerita akan memakan waktu sekitar 0,5 detik untuk disintesis dan 10 detik untuk diputar. Dalam skenario ini, Anda dapat mengharapkan satu panggilan SynthesizeSpeech untuk setiap 10 detik yang dihabiskan pelanggan menggunakan aplikasi. Ini akan diterjemahkan menjadi satu panggilan per detik untuk setiap 10 pelanggan yang secara bersamaan menggunakan aplikasi.

Jika Anda memiliki 1000 pelanggan secara bersamaan menggunakan aplikasi, Anda dapat mengharapkan tingkat panggilan rata-rata hanya 100 transaksi per detik. SynthesizeSpeech

Perhatikan batasan berikut yang terkait dengan penggunaan operasi SynthesizeSpeech API:

- Ukuran teks input dapat mencapai 3000 karakter yang ditagih (total 6000 karakter). Tag SSML tidak dihitung sebagai karakter yang ditagih.
- Anda dapat menentukan hingga lima leksikon untuk diterapkan ke teks input.
- Aliran audio output (sintesis) dibatasi hingga 10 menit. Setelah ini tercapai, pidato yang tersisa terputus.

Untuk informasi selengkapnya, lihat [SynthesizeSpeech](#).

Note

Beberapa batasan operasi SynthesizeSpeech API dapat dilewati menggunakan operasi StartSynthesizeSpeechTask API. Untuk informasi selengkapnya, lihat [File audio panjang](#).

SpeechSynthesisTask Operasi API

Perhatikan batasan berikut terkait penggunaan StartSpeechSynthesisTask, GetSpeechSynthesisTask, dan operasi ListSpeechSynthesisTasks API:

- Ukuran teks input dapat mencapai 100.000 karakter yang ditagih (total 200.000 karakter). Tag SSML tidak dihitung sebagai karakter yang ditagih.
- Anda dapat menentukan hingga lima leksikon untuk diterapkan ke teks input.

Bahasa Markup Sintesis Pidato (SSML)

Perhatikan batasan berikut yang terkait dengan penggunaan SSML:

- Tag <audio><lexicon>, <lookup>, dan <voice> tag tidak didukung.
- <break> elemen dapat menentukan durasi maksimum masing-masing 10 detik.

- `<prosody>` Tag tidak mendukung nilai untuk atribut `rate` yang lebih rendah dari -80%.

Lihat informasi yang lebih lengkap di [Menghasilkan pidato dari dokumen SSML](#).

Contoh kode dan aplikasi untuk Amazon Polly

Bagian ini menyediakan contoh kode dan contoh aplikasi yang dapat Anda gunakan untuk menjelajahi Amazon Polly.

Topik Kode Sampel berisi cuplikan kode yang diatur oleh bahasa pemrograman dan dipisahkan menjadi contoh untuk fungsionalitas Amazon Polly yang berbeda. Topik Contoh Aplikasi berisi aplikasi yang diatur oleh bahasa pemrograman yang dapat digunakan secara independen untuk menjelajahi Amazon Polly.

Sebelum Anda mulai menggunakan contoh-contoh ini, kami sarankan Anda terlebih dahulu membaca [Bagaimana Amazon Polly bekerja](#) dan mengikuti langkah-langkah yang dijelaskan di [Memulai dengan Amazon Polly](#).

Topik

- [Sampel Java](#)
- [Sampel Python](#)
- [Contoh Java](#)
- [Contoh Python \(HTML5 Klien dan Server Python\)](#)
- [Contoh iOS](#)
- [Contoh Android](#)

Sampel Java

Contoh kode berikut menunjukkan cara menggunakan aplikasi berbasis Java untuk menyelesaikan berbagai tugas dengan Amazon Polly. Sampel ini bukan contoh lengkap, tetapi dapat dimasukkan dalam aplikasi Java yang lebih besar yang menggunakan [AWS SDK untuk Java](#).

Cuplikan Kode

- [DeleteLexicon](#)
- [DescribeVoices](#)
- [GetLexicon](#)
- [ListLexicons](#)
- [PutLexicon](#)

- [StartSpeechSynthesisTask](#)
- [Tanda Pidato](#)
- [SynthesizeSpeech](#)

DeleteLexicon

Contoh kode Java berikut menunjukkan cara menggunakan aplikasi berbasis Java untuk menghapus leksikon tertentu yang disimpan di Wilayah. AWS Sebuah leksikon yang telah dihapus tidak tersedia untuk sintesis ucapan, juga tidak dapat diambil menggunakan salah satu atau. GetLexicon ListLexicon APIs

Untuk informasi selengkapnya tentang operasi ini, lihat referensi untuk [DeleteLexiconAPI](#).

SDK v2

```
/*
   Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
   SPDX-License-Identifier: Apache-2.0
 */

package com.example.polly;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.polly.PollyClient;
import software.amazon.awssdk.services.polly.model.DeleteLexiconRequest;
import software.amazon.awssdk.services.polly.model.DeleteLexiconResponse;
import software.amazon.awssdk.services.polly.model.PollyException ;

/**
 * Before running this Java V2 code example, set up your development environment,
 * including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class DeleteLexiconSample {

    public static void main(String args[]) {
```

```

        PollyClient polly = PollyClient.builder()
            .region(Region.US_WEST_2)
            .credentialsProvider(ProfileCredentialsProvider.create())
            .build();

        deleteLexicon(polly) ;
        polly.close();
    }

    private static String LEXICON_NAME = "SampleLexicon";
    public static void deleteLexicon(PollyClient client) {

        try {
            DeleteLexiconRequest deleteLexiconRequest = DeleteLexiconRequest.builder()
                .name(LEXICON_NAME).build();

            DeleteLexiconResponse deleteLexiconResult =
client.deleteLexicon(deleteLexiconRequest);

        } catch (PollyException e) {
            System.err.println("Exception caught: " + e);
            System.exit(1);
        }
    }
}

```

SDK v1

```

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

Contoh kode Java berikut menunjukkan cara menggunakan aplikasi berbasis Java untuk menghasilkan daftar suara yang tersedia untuk digunakan saat meminta sintesis ucapan. Anda dapat secara opsional menentukan kode bahasa untuk memfilter suara yang tersedia. Misalnya, jika Anda menentukan en-US, operasi mengembalikan daftar semua suara bahasa Inggris AS yang tersedia.

Untuk informasi selengkapnya tentang operasi ini, lihat referensi untuk [DescribeVoices](#) API.

```
package com.amazonaws.polly.samples;

import com.amazonaws.services.polly.AmazonPolly;
import com.amazonaws.services.polly.AmazonPollyClientBuilder;
import com.amazonaws.services.polly.model.DescribeVoicesRequest;
import com.amazonaws.services.polly.model.DescribeVoicesResult;

public class DescribeVoicesSample {
    AmazonPolly client = AmazonPollyClientBuilder.defaultClient();

    public void describeVoices() {
        DescribeVoicesRequest allVoicesRequest = new DescribeVoicesRequest();
        DescribeVoicesRequest enUsVoicesRequest = new
        DescribeVoicesRequest().withLanguageCode("en-US");

        try {
            String nextToken;
            do {
                DescribeVoicesResult allVoicesResult =
                client.describeVoices(allVoicesRequest);
                nextToken = allVoicesResult.getNextToken();
                allVoicesRequest.setNextToken(nextToken);

                System.out.println("All voices: " + allVoicesResult.getVoices());
            } while (nextToken != null);
        }
    }
}
```

```

        do {
            DescribeVoicesResult enUsVoicesResult =
client.describeVoices(enUsVoicesRequest);
            nextToken = enUsVoicesResult.getNextToken();
            enUsVoicesRequest.setNextToken(nextToken);

            System.out.println("en-US voices: " + enUsVoicesResult.getVoices());
        } while (nextToken != null);
    } catch (Exception e) {
        System.err.println("Exception caught: " + e);
    }
}
}

```

GetLexicon

Contoh kode Java berikut menunjukkan cara menggunakan aplikasi berbasis Java untuk menghasilkan konten leksikon pengucapan tertentu yang disimpan di Wilayah. AWS

Untuk informasi selengkapnya tentang operasi ini, lihat referensi untuk [GetLexiconAPI](#).

SDK v2

```

/*
    Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
    SPDX-License-Identifier: Apache-2.0
*/

package com.example.polly;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.polly.PollyClient;
import software.amazon.awssdk.services.polly.model.GetLexiconRequest;
import software.amazon.awssdk.services.polly.model.GetLexiconResponse;
import software.amazon.awssdk.services.polly.model.PollyException ;

/**
 * Before running this Java V2 code example, set up your development environment,
 * including your credentials.
 *
 * For more information, see the following documentation topic:

```

```

*
* https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
*/
public class GetLexiconSample {

    public static void main(String args[]) {

        PollyClient polly = PollyClient.builder()
            .region(Region.US_WEST_2)
            .credentialsProvider(ProfileCredentialsProvider.create())
            .build();

        getLexicon(polly) ;
        polly.close();
    }

    private static String LEXICON_NAME = "SampleLexicon";
    public static void getLexicon(PollyClient client) {

        try {
            GetLexiconRequest getLexiconRequest = GetLexiconRequest.builder()
                .name(LEXICON_NAME).build();

            GetLexiconResponse getLexiconResult = client.getLexicon(getLexiconRequest);
            System.out.println("The name of the Lexicon is " +
getLexiconResult.lexicon().name());

        } catch (PollyException e) {
            System.err.println("Exception caught: " + e);
            System.exit(1);
        }
    }
}

```

SDK v1

```

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

Contoh kode Java berikut menunjukkan cara menggunakan aplikasi berbasis Java untuk menghasilkan daftar leksikon pengucapan yang disimpan di Wilayah. AWS

Untuk informasi selengkapnya tentang operasi ini, lihat referensi untuk [ListLexiconsAPI](#).

```
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

Contoh kode Java berikut menunjukkan cara menggunakan aplikasi berbasis Java untuk menyimpan leksikon pengucapan di Wilayah. AWS

Untuk informasi selengkapnya tentang operasi ini, lihat referensi untuk [PutLexiconAPI](#).

SDK v2

```

/*
    Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
    SPDX-License-Identifier: Apache-2.0
*/

package com.example.polly;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.polly.PollyClient;

```

```

import software.amazon.awssdk.services.polly.model.PutLexiconRequest;
import software.amazon.awssdk.services.polly.model.PutLexiconResponse;
import software.amazon.awssdk.services.polly.model.PollyException ;

/**
 * Before running this Java V2 code example, set up your development environment,
 * including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class PutLexiconSample {

    public static void main(String args[]) {

        PollyClient polly = PollyClient.builder()
            .region(Region.US_WEST_2)
            .credentialsProvider(ProfileCredentialsProvider.create())
            .build();

        putLexicon(polly) ;
        polly.close();
    }

    private static 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 static String LEXICON_NAME = "SampleLexicon";
    public static void putLexicon(PollyClient client) {

        try {
            PutLexiconRequest putLexiconRequest = PutLexiconRequest.builder()
                .name(LEXICON_NAME).content(LEXICON_CONTENT).build();

            PutLexiconResponse putLexiconResult = client.putLexicon(putLexiconRequest);

        } catch (PollyException e) {

```

```

        System.err.println("Exception caught: " + e);
        System.exit(1);
    }
}
}

```

SDK v1

```

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

Contoh kode Java berikut menunjukkan cara menggunakan aplikasi berbasis Java untuk mensintesis pidato panjang (hingga 100.000 karakter yang ditagih) dan menyimpannya langsung di bucket Amazon S3.

Untuk informasi selengkapnya, lihat referensi untuk [StartSpeechSynthesisTask](#) API.

SDK v2

```
/*
    Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
    SPDX-License-Identifier: Apache-2.0
*/

package com.example.polly;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.polly.PollyClient;
import software.amazon.awssdk.services.polly.model.*;

import java.time.Duration;
import org.awaitility.Durations;
import org.awaitility.Awaitility;

/**
 * Before running this Java V2 code example, set up your development environment,
 * including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class StartSpeechSynthesisTaskSample {

    public static void main(String args[]) {

        PollyClient polly = PollyClient.builder()
            .region(Region.US_WEST_2)
            .credentialsProvider(ProfileCredentialsProvider.create())
            .build();
```

```

        startSpeechSynthesisTask(polly) ;
        polly.close();
    }

    private static final String PLAIN_TEXT = "This is a sample text to be
synthesized.";
    private static final String OUTPUT_FORMAT_MP3 = OutputFormat.MP3.toString();
    private static final String OUTPUT_BUCKET = "synth-books-buckets";
    private static final String SNS_TOPIC_ARN = "arn:aws:sns:eu-
west-2:123456789012:synthesize-finish-topic";
    private static final Duration SYNTHESIS_TASK_POLL_INTERVAL =
Durations.FIVE_SECONDS;
    private static final Duration SYNTHESIS_TASK_POLL_DELAY = Durations.TEN_SECONDS;
    private static final Duration SYNTHESIS_TASK_TIMEOUT = Durations.FIVE_MINUTES;
    public static void startSpeechSynthesisTask(PollyClient client) {

        try {
            StartSpeechSynthesisTaskRequest startSpeechSynthesisTaskRequest =
StartSpeechSynthesisTaskRequest.builder()

.outputFormat(OUTPUT_FORMAT_MP3).text(PLAIN_TEXT).textType(TextType.TEXT)

.voiceId(VoiceId.AMY).outputS3BucketName(OUTPUT_BUCKET).snsTopicArn(SNS_TOPIC_ARN)
                .engine("neural").build();

            StartSpeechSynthesisTaskResponse startSpeechSynthesisTaskResponse =
                client.startSpeechSynthesisTask(startSpeechSynthesisTaskRequest);
            String taskId = startSpeechSynthesisTaskResponse.synthesisTask().taskId();

            Awaitility.await().with()
                .pollInterval(SYNTHESIS_TASK_POLL_INTERVAL)
                .pollDelay(SYNTHESIS_TASK_POLL_DELAY)
                .atMost(SYNTHESIS_TASK_TIMEOUT)
                .until(
                    () -> getSynthesisTaskStatus(client,
taskId).equals(TaskStatus.COMPLETED.toString())
                );

        } catch (PollyException e) {
            System.err.println("Exception caught: " + e);
            System.exit(1);
        }
    }
}

```

```

    private static String getSynthesisTaskStatus(PollyClient client, String taskId) {
        GetSpeechSynthesisTaskRequest getSpeechSynthesisTaskRequest =
        GetSpeechSynthesisTaskRequest.builder()
            .taskId(taskId).build();
        GetSpeechSynthesisTaskResponse result =
        client.getSpeechSynthesisTask(getSpeechSynthesisTaskRequest);
        return result.synthesisTask().taskStatusAsString();
    }
}

```

SDK v1

```

package com.amazonaws.parrot.service.tests.speech.task;

import com.amazonaws.parrot.service.tests.AbstractParrotServiceTest;
import com.amazonaws.services.polly.AmazonPolly;
import com.amazonaws.services.polly.model.*;
import org.awaitility.Duration;

import java.util.concurrent.TimeUnit;

import static org.awaitility.Awaitility.await;

public class StartSpeechSynthesisTaskSample {

    private static final int SYNTHESIS_TASK_TIMEOUT_SECONDS = 300;
    private static final AmazonPolly AMAZON_POLLY_CLIENT =
    AmazonPollyClientBuilder.defaultClient();
    private static final String PLAIN_TEXT = "This is a sample text to be
    synthesized.";
    private static final String OUTPUT_FORMAT_MP3 = OutputFormat.Mp3.toString();
    private static final String OUTPUT_BUCKET = "synth-books-buckets";
    private static final String SNS_TOPIC_ARN = "arn:aws:sns:eu-
    west-2:123456789012:synthesize-finish-topic";
    private static final Duration SYNTHESIS_TASK_POLL_INTERVAL = Duration.FIVE_SECONDS;
    private static final Duration SYNTHESIS_TASK_POLL_DELAY = Duration.TEN_SECONDS;

    public static void main(String... args) {
        StartSpeechSynthesisTaskRequest request = new StartSpeechSynthesisTaskRequest()
            .withOutputFormat(OUTPUT_FORMAT_MP3)
            .withText(PLAIN_TEXT)
            .withTextType(TextType.Text)
    }
}

```

```

        .withVoiceId(VoiceId.Amy)
        .withOutputS3BucketName(OUTPUT_BUCKET)
        .withSnsTopicArn(SNS_TOPIC_ARN)
        .withEngine("neural");

    StartSpeechSynthesisTaskResult result =
    AMAZON_POLLY_CLIENT.startSpeechSynthesisTask(request);
    String taskId = result.getSynthesisTask().getTaskId();

    await().with()
        .pollInterval(SYNTHESIS_TASK_POLL_INTERVAL)
        .pollDelay(SYNTHESIS_TASK_POLL_DELAY)
        .atMost(SYNTHESIS_TASK_TIMEOUT_SECONDS, TimeUnit.SECONDS)
        .until(
            () ->
            getSynthesisTaskStatus(taskId).equals(TaskStatus.Completed.toString())
        );
    }

    private static SynthesisTask getSynthesisTask(String taskId) {
        GetSpeechSynthesisTaskRequest getSpeechSynthesisTaskRequest = new
        GetSpeechSynthesisTaskRequest()
            .withTaskId(taskId);
        GetSpeechSynthesisTaskResult result
        =AMAZON_POLLY_CLIENT.getSpeechSynthesisTask(getSpeechSynthesisTaskRequest);
        return result.getSynthesisTask();
    }

    private static String getSynthesisTaskStatus(String taskId) {
        GetSpeechSynthesisTaskRequest getSpeechSynthesisTaskRequest = new
        GetSpeechSynthesisTaskRequest()
            .withTaskId(taskId);
        GetSpeechSynthesisTaskResult result
        =AMAZON_POLLY_CLIENT.getSpeechSynthesisTask(getSpeechSynthesisTaskRequest);
        return result.getSynthesisTask().getTaskStatus();
    }
}

```

Tanda Pidato

Contoh kode berikut menunjukkan cara menggunakan aplikasi berbasis Java untuk mensintesis tanda ucapan untuk teks yang dimasukkan. Fungsionalitas ini menggunakan SynthesizeSpeech API.

Untuk informasi lebih lanjut tentang fungsi ini, lihat [Tanda bicara](#).

Untuk informasi selengkapnya tentang API, lihat referensi untuk [SynthesizeSpeech](#) API.

SDK v2

```
/*
    Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
    SPDX-License-Identifier: Apache-2.0
*/

package com.example.polly;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.core.ResponseInputStream;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.polly.PollyClient;
import software.amazon.awssdk.services.polly.model.*;

import java.io.File;
import java.io.FileOutputStream;
import java.io.InputStream;

/**
 * Before running this Java V2 code example, set up your development environment,
 * including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class SpeechMarksSample {

    public static void main(String args[]) {

        PollyClient polly = PollyClient.builder()
            .region(Region.US_WEST_2)
            .credentialsProvider(ProfileCredentialsProvider.create())
            .build();

        speechMarksSample(polly) ;
        polly.close();
    }
}
```

```

private static final String OUTPUT_FILE = "./speechMarks.json";
public static void speechMarksSample(PollyClient client) {

    try {
        SynthesizeSpeechRequest speechMarksSampleRequest =
SynthesizeSpeechRequest.builder()
            .outputFormat(OutputFormat.JSON)
            .speechMarkTypes(SpeechMarkType.VISEME, SpeechMarkType.WORD)
            .voiceId(VoiceId.JOANNA)
            .text("This is a sample text to be synthesized")
            .build();
        try (FileOutputStream outputStream = new FileOutputStream(new
File(OUTPUT_FILE))) {
            ResponseInputStream<SynthesizeSpeechResponse> synthesizeSpeechResponse
= client
                .synthesizeSpeech(speechMarksSampleRequest);
            byte[] buffer = new byte[2 * 1024];
            int readBytes;

            try (InputStream in = synthesizeSpeechResponse){
                while ((readBytes = in.read(buffer)) > 0) {
                    outputStream.write(buffer, 0, readBytes);
                }
            }
        } catch (Exception e) {
            System.err.println("Exception caught: " + e);
        }

        } catch (PollyException e) {
            System.err.println("Exception caught: " + e);
            System.exit(1);
        }
    }
}

```

SDK v1

```

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

Contoh kode Java berikut menunjukkan cara menggunakan aplikasi berbasis Java untuk mensintesis ucapan dengan teks yang lebih pendek untuk pemrosesan waktu nyaris nyata.

Untuk informasi selengkapnya, lihat referensi untuk [SynthesizeSpeech](#) API.

```
package com.amazonaws.polly.samples;

import com.amazonaws.services.polly.AmazonPolly;
import com.amazonaws.services.polly.AmazonPollyClientBuilder;
import com.amazonaws.services.polly.model.OutputFormat;
import com.amazonaws.services.polly.model.SynthesizeSpeechRequest;
import com.amazonaws.services.polly.model.SynthesizeSpeechResult;
import com.amazonaws.services.polly.model.VoiceId;

import java.io.File;
import java.io.FileOutputStream;
import java.io.InputStream;

public class SynthesizeSpeechSample {
    AmazonPolly client = AmazonPollyClientBuilder.defaultClient();

    public void synthesizeSpeech() {
        String outputFileName = "/tmp/speech.mp3";

        SynthesizeSpeechRequest synthesizeSpeechRequest = new SynthesizeSpeechRequest()
            .withOutputFormat(OutputFormat.Mp3)
            .withVoiceId(VoiceId.Joanna)
            .withText("This is a sample text to be synthesized.")
            .withEngine("neural");

        try (FileOutputStream outputStream = new FileOutputStream(new
File(outputFileName))) {
            SynthesizeSpeechResult synthesizeSpeechResult =
client.synthesizeSpeech(synthesizeSpeechRequest);
            byte[] buffer = new byte[2 * 1024];
            int readBytes;

            try (InputStream in = synthesizeSpeechResult.getAudioStream()){
                while ((readBytes = in.read(buffer)) > 0) {
                    outputStream.write(buffer, 0, readBytes);
                }
            }
        } catch (Exception e) {
            System.err.println("Exception caught: " + e);
        }
    }
}
```

```
}
```

Sampel Python

Contoh kode berikut menunjukkan cara menggunakan aplikasi berbasis Python (boto3) untuk menyelesaikan berbagai tugas dengan Amazon Polly. Sampel ini tidak dimaksudkan untuk menjadi contoh lengkap, tetapi dapat dimasukkan dalam aplikasi Python yang lebih besar yang menggunakan [AWS SDK for Python \(Boto\)](#).

Cuplikan Kode

- [DeleteLexicon](#)
- [GetLexicon](#)
- [ListLexicon](#)
- [PutLexicon](#)
- [StartSpeechSynthesisTask](#)
- [SynthesizeSpeech](#)

DeleteLexicon

Contoh kode Python berikut menggunakan AWS SDK for Python (Boto) untuk menghapus leksikon di wilayah yang ditentukan dalam konfigurasi lokal Anda. AWS Contoh hanya menghapus leksikon yang ditentukan. Ini meminta Anda untuk mengonfirmasi bahwa Anda ingin melanjutkan sebelum benar-benar menghapus leksikon.

Contoh kode berikut menggunakan kredensi default yang disimpan dalam file konfigurasi AWS SDK. Untuk informasi tentang membuat file konfigurasi, lihat [Menyiapkan AWS CLI](#).

Untuk informasi selengkapnya tentang operasi ini, lihat referensi untuk [DeleteLexicon](#) API.

```
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

Kode Python berikut menggunakan AWS SDK for Python (Boto) untuk mengambil semua leksikon yang disimpan di Wilayah. AWS Contoh menerima nama leksikon sebagai parameter baris perintah dan mengambil leksikon itu saja, mencetak jalur tmp di mana ia telah disimpan secara lokal.

Contoh kode berikut menggunakan kredensi default yang disimpan dalam file konfigurasi AWS SDK. Untuk informasi tentang membuat file konfigurasi, lihat [Menyiapkan AWS CLI](#).

Untuk informasi selengkapnya tentang operasi ini, lihat referensi untuk [GetLexicon API](#).

```

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

Contoh kode Python berikut menggunakan daftar leksikon di akun Anda di wilayah yang ditentukan dalam konfigurasi lokal Anda. AWS SDK for Python (Boto) AWS Untuk informasi tentang membuat file konfigurasi, lihat [Menyiapkan AWS CLI](#).

Untuk informasi selengkapnya tentang operasi ini, lihat referensi untuk [ListLexicons](#) API.

```
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

Contoh kode berikut menunjukkan cara menggunakan aplikasi berbasis Python (boto3) untuk menyimpan leksikon pengucapan di Wilayah. AWS

Untuk informasi selengkapnya tentang operasi ini, lihat referensi untuk [PutLexicon](#) API.

Perhatikan hal berikut:

- Anda perlu memperbarui kode dengan memberikan nama file leksikon lokal dan nama leksikon yang disimpan.
- Contoh mengasumsikan Anda memiliki file leksikon yang dibuat dalam subdirektori yang disebut `pls`. Anda perlu memperbarui jalur yang sesuai.

Contoh kode berikut menggunakan kredensi default yang disimpan dalam file konfigurasi AWS SDK. Untuk informasi tentang membuat file konfigurasi, lihat [Menyiapkan AWS CLI](#).

Untuk informasi selengkapnya tentang operasi ini, lihat referensi untuk [PutLexicon](#) API.

```
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

Contoh kode Python berikut menggunakan daftar leksikon di akun Anda di wilayah yang ditentukan dalam konfigurasi lokal Anda. AWS SDK for Python (Boto) AWS Untuk informasi tentang membuat file konfigurasi, lihat [Menyiapkan AWS CLI](#).

Untuk informasi selengkapnya, lihat referensi untuk [StartSpeechSynthesisTask](#) API.

```
import boto3
import time

polly_client = boto3.Session(
    aws_access_key_id='',
    aws_secret_access_key='',
    region_name='eu-west-2').client('polly')

response = polly_client.start_speech_synthesis_task(VoiceId='Joanna',
    OutputS3BucketName='synth-books-buckets',
    OutputS3KeyPrefix='key',
    OutputFormat='mp3',
    Text='This is a sample text to be synthesized.',
    Engine='neural')

taskId = response['SynthesisTask']['TaskId']

print( "Task id is {} ".format(taskId))

task_status = polly_client.get_speech_synthesis_task(TaskId = taskId)

print(task_status)
```

SynthesizeSpeech

Contoh kode Python berikut menggunakan ucapan AWS SDK for Python (Boto) sintesis dengan teks yang lebih pendek untuk pemrosesan mendekati waktu nyata. Untuk informasi lebih lanjut, lihat referensi untuk [SynthesizeSpeech](#) operasi.

Contoh ini menggunakan string pendek teks biasa. Anda dapat menggunakan teks SSML untuk kontrol lebih besar atas output. Lihat informasi yang lebih lengkap di [Menghasilkan pidato dari dokumen SSML](#).

```
import boto3

polly_client = boto3.Session(
    aws_access_key_id=,
    aws_secret_access_key=,
    region_name='us-west-2').client('polly')

response = polly_client.synthesize_speech(VoiceId='Joanna',
    OutputFormat='mp3',
    Text = 'This is a sample text to be synthesized.',
    Engine = 'neural')

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

Contoh Java

Contoh ini menunjukkan cara menggunakan Amazon Polly untuk melakukan streaming ucapan dari aplikasi berbasis Java. Contoh menggunakan [AWS SDK untuk Java](#) untuk membaca teks yang ditentukan menggunakan suara yang dipilih dari daftar.

Kode yang ditampilkan mencakup tugas-tugas utama, tetapi hanya melakukan pemeriksaan kesalahan minimal. Jika Amazon Polly mengalami kesalahan, aplikasi akan berakhir.

Untuk menjalankan aplikasi contoh ini, Anda memerlukan yang berikut:

- Kit Pengembangan Java 8 Java (JDK)
- [AWS SDK untuk Java](#)
- [Apache Maven](#)

Untuk menguji aplikasi

1. Pastikan bahwa variabel lingkungan JAVA_HOME diatur untuk JDK.

Misalnya, jika Anda menginstal JDK 1.8.0_121 pada Windows di C:\Program Files\Java\jdk1.8.0_121, Anda akan mengetik yang berikut ini di command prompt:

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

Jika Anda menginstal JDK 1.8.0_121 di Linux di /usr/lib/jvm/java8-openjdk-amd64, Anda akan mengetik yang berikut di command prompt:

```
export JAVA_HOME=/usr/lib/jvm/java8-openjdk-amd64
```

2. Atur variabel lingkungan Maven untuk menjalankan Maven dari baris perintah.

Misalnya, jika Anda menginstal Maven 3.3.9 pada Windows di C:\Program Files\apache-maven-3.3.9, Anda akan mengetik yang berikut ini:

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

Jika Anda menginstal Maven 3.3.9 di Linux di /home/ec2-user/opt/apache-maven-3.3.9, Anda akan mengetik yang berikut ini:

```
export M2_HOME=/home/ec2-user/opt/apache-maven-3.3.9
export M2=$M2_HOME/bin
export PATH=$M2:$PATH
```

3. Buat direktori baru yang disebut `polly-java-demo`.
4. Di `polly-java-demo` direktori, buat file baru bernama `pom.xml`, dan tempelkan kode berikut ke dalamnya:

```
<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. Buat direktori baru yang disebut polly di src/main/java/com/amazonaws/demos.
6. Di polly direktori, buat file sumber Java baru yang disebut PollyDemo.java, dan tempel dalam kode berikut:

```
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.sound.sampled.AudioFormat;
import javax.sound.sampled.AudioInputStream;
import javax.sound.sampled.AudioSystem;
import javax.sound.sampled.DataLine;
import javax.sound.sampled.LineUnavailableException;
import javax.sound.sampled.SourceDataLine;

import javax.swing.*;
import javax.swing.text.TextArea;

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class PollyDemo {

    private final AmazonPollyClient polly;
    private final Voice voice;
    private static final String SAMPLE = "Congratulations. You have successfully built this working demo of Amazon Polly in Java. Have fun building voice enabled apps with Amazon Polly (that's me!), and always look at the AWS website for tips and tricks on using Amazon Polly and other great services from AWS";

    public PollyDemo(Region region) {
        // create an Amazon Polly client in a specific region
        polly = new AmazonPollyClient(new DefaultAWSCredentialsProviderChain(),
            new ClientConfiguration());
        polly.setRegion(region);
        // Create describe voices request.
        DescribeVoicesRequest describeVoicesRequest = new DescribeVoicesRequest();

        // Synchronously ask Amazon Polly to describe available TTS voices.
        DescribeVoicesResult describeVoicesResult =
            polly.describeVoices(describeVoicesRequest);
        voice = describeVoicesResult.getVoices().get(0);
    }
}
```

```
public InputStream synthesize(String text, OutputFormat format) throws IOException
{
    SynthesizeSpeechRequest synthReq =
        new SynthesizeSpeechRequest().withText(text).withVoiceId(voice.getId())
            .withOutputFormat(format).withEngine("neural");
    SynthesizeSpeechResult synthRes = polly.synthesizeSpeech(synthReq);

    return synthRes.getAudioStream();
}

public static void main(String args[]) throws Exception {
    //create the test class
    PollyDemo helloWorld = new PollyDemo(Region.getRegion(Regions.US_EAST_1));
    //get the audio stream
    InputStream speechStream = helloWorld.synthesize(SAMPLE, OutputFormat.Mp3);

    //create an MP3 player
    AdvancedPlayer player = new AdvancedPlayer(speechStream,
        javazoom.jl.player.FactoryRegistry.systemRegistry().createAudioDevice());

    player.setPlaybackListener(new PlaybackListener() {
        @Override
        public void playbackStarted(PlaybackEvent evt) {
            System.out.println("Playback started");
            System.out.println(SAMPLE);
        }

        @Override
        public void playbackFinished(PlaybackEvent evt) {
            System.out.println("Playback finished");
        }
    });

    // play it!
    player.play();
}
```

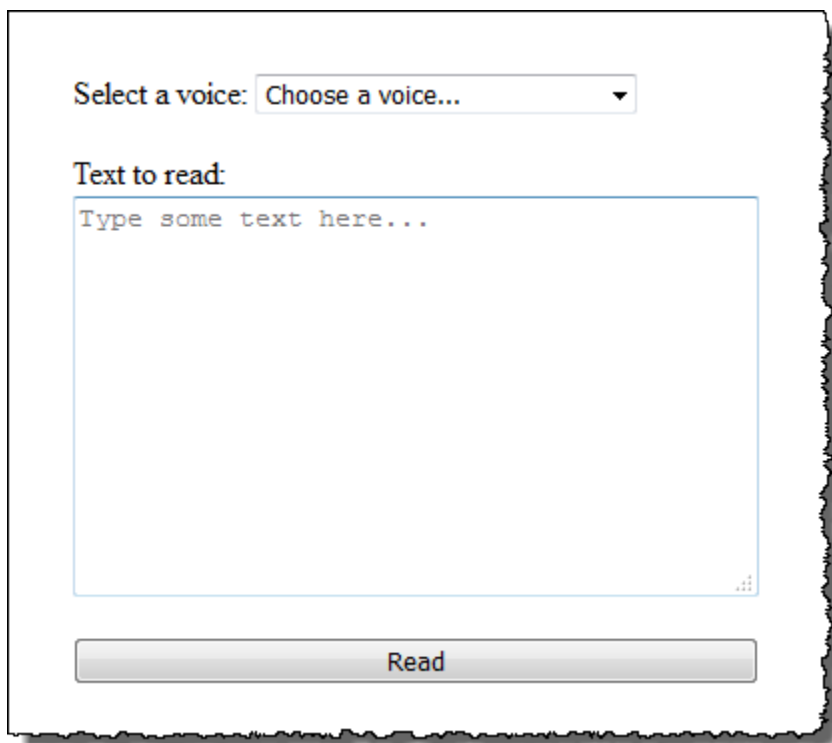
7. Kembali ke polly-java-demo direktori untuk membersihkan, mengkompilasi, dan menjalankan demo:

```
mvn clean compile exec:java
```

Contoh Python (HTML5 Klien dan Server Python)

Contoh aplikasi ini terdiri dari yang berikut:

- Server HTTP 1.1 yang menggunakan kode transfer chunked HTTP (lihat [Chunked Transfer Coding](#))
- Antarmuka HTML5 pengguna sederhana yang berinteraksi dengan server HTTP 1.1 (ditunjukkan di bawah):



The screenshot shows a web browser window with a simple interface. At the top, there is a label "Select a voice:" followed by a dropdown menu with the text "Choose a voice...". Below this, there is a label "Text to read:" followed by a large text input area. Inside the text input area, the placeholder text "Type some text here..." is visible. At the bottom of the interface, there is a button labeled "Read".

Tujuan dari contoh ini adalah untuk menunjukkan cara menggunakan Amazon Polly untuk mengalirkan ucapan dari aplikasi berbasis browser HTML5. Mengonsumsi aliran audio yang dihasilkan oleh Amazon Polly saat teks disintesis adalah pendekatan yang disarankan untuk kasus penggunaan di mana daya tanggap merupakan faktor penting (misalnya, sistem dialog, pembaca layar, dll.).

Untuk menjalankan aplikasi contoh ini, Anda memerlukan yang berikut:

- Browser web sesuai dengan standar HTML5 dan EcmaScript 5 (misalnya, Chrome 23.0 atau lebih tinggi, Firefox 21.0 atau lebih tinggi, Internet Explorer 9.0, atau lebih tinggi)
- Versi Python lebih besar dari 3.0

Untuk menguji aplikasi

1. Simpan kode server sebagai `server.py`. Untuk kodenya, lihat [Contoh Python: Kode Server Python \(server.py\)](#).
2. Simpan kode HTML5 klien sebagai `index.html`. Untuk kodenya, lihat [Contoh Python: Antarmuka HTML5 Pengguna \(index.html\)](#).
3. Jalankan perintah berikut dari jalur tempat Anda menyimpan `server.py` untuk memulai aplikasi (pada beberapa sistem yang mungkin perlu Anda gunakan `python3` alih-alih `python` saat menjalankan perintah).

```
$ python server.py
```

Setelah aplikasi dimulai, URL muncul di terminal.

4. Buka URL yang ditampilkan di terminal di browser web.

Anda dapat meneruskan alamat dan port untuk server aplikasi untuk digunakan sebagai parameter `server.py`. Untuk informasi lebih lanjut, jalankan `python server.py -h`.

5. Untuk mendengarkan pidato, pilih suara dari daftar, ketik beberapa teks, lalu pilih Baca. Pidato mulai diputar segera setelah Amazon Polly mentransfer potongan data audio pertama yang dapat digunakan.
6. Untuk menghentikan server Python ketika Anda selesai menguji aplikasi, tekan Ctrl+C di terminal tempat server berjalan.

Note

Server membuat klien Boto3 menggunakan file. AWS SDK for Python (Boto) Klien menggunakan kredensial yang disimpan dalam file AWS konfigurasi di komputer Anda untuk menandatangani dan mengautentikasi permintaan ke Amazon Polly. Untuk informasi selengkapnya tentang cara membuat file AWS konfigurasi dan menyimpan kredensialnya,

lihat [Mengonfigurasi AWS Command Line Interface dalam Panduan Pengguna](#).AWS
Command Line Interface

Contoh Python: Antarmuka HTML5 Pengguna (index.html)

Bagian ini menyediakan kode untuk HTML5 klien yang dijelaskan dalam [Contoh Python \(HTML5 Klien dan Server Python\)](#).

```
<html>

<head>
  <title>Text-to-Speech Example Application</title>
  <script>
    /*
     * This sample code requires a web browser with support for both the
     * HTML5 and ECMAScript 5 standards; the following is a non-comprehensive
     * list of compliant browsers and their minimum version:
     *
     * - Chrome 23.0+
     * - Firefox 21.0+
     * - Internet Explorer 9.0+
     * - Edge 12.0+
     * - Opera 15.0+
     * - Safari 6.1+
     * - Android (stock web browser) 4.4+
     * - Chrome for Android 51.0+
     * - Firefox for Android 48.0+
     * - Opera Mobile 37.0+
     * - iOS (Safari Mobile and Chrome) 3.2+
     * - Internet Explorer Mobile 10.0+
     * - Blackberry Browser 10.0+
     */

    // Mapping of the OutputFormat parameter of the SynthesizeSpeech API
    // and the audio format strings understood by the browser
    var AUDIO_FORMATS = {
      'ogg_vorbis': 'audio/ogg',
      'mp3': 'audio/mpeg',
      'pcm': 'audio/wave; codecs=1',
      'mulaw': 'audio/mulaw',
      'alaw': 'audio/alaw'
```

```
};

/**
 * 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>

```

Contoh Python: Kode Server Python (server.py)

Bagian ini menyediakan kode untuk server Python yang dijelaskan dalam [Contoh Python \(HTML5 Klien dan Server Python\)](#)

```

"""

```

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",
                  "mulaw": "audio/mulaw",
```

```

        "alaw": "audio/alaw"}
CHUNK_SIZE = 1024
HTTP_STATUS = {"OK": ResponseStatus(code=200, message="OK"),
               "BAD_REQUEST": ResponseStatus(code=400, message="Bad request"),
               "NOT_FOUND": ResponseStatus(code=404, message="Not found"),
               "INTERNAL_SERVER_ERROR": ResponseStatus(code=500, message="Internal
server error")}
PROTOCOL = "http"
ROUTE_INDEX = "/index.html"
ROUTE_VOICES = "/voices"
ROUTE_READ = "/read"

# Create a client using the credentials and region defined in the adminuser
# section of the AWS credentials and configuration files
session = Session(profile_name="adminuser")
polly = session.client("polly")

class HTTPStatusError(Exception):
    """Exception wrapping a value from http.server.HTTPStatus"""

    def __init__(self, status, description=None):
        """
        Constructs an error instance from a tuple of
        (code, message, description), see http.server.HTTPStatus
        """
        super(HTTPStatusError, self).__init__()
        self.code = status.code
        self.message = status.message
        self.explain = description

class ThreadedHTTPServer(ThreadingMixIn, HTTPServer):
    """An HTTP Server that handle each request in a new thread"""
    daemon_threads = True

class ChunkedHTTPRequestHandler(BaseHTTPRequestHandler):
    """HTTP 1.1 Chunked encoding request handler"""
    # Use HTTP 1.1 as 1.0 doesn't support chunked encoding
    protocol_version = "HTTP/1.1"

    def query_get(self, queryData, key, default=""):

```

```

        """Helper for getting values from a pre-parsed query string"""
        return queryData.get(key, [default])[0]

def do_GET(self):
    """Handles GET requests"""

    # Extract values from the query string
    path, _, query_string = self.path.partition('?')
    query = parse_qs(query_string)

    response = None

    print(u"[START]: Received GET for %s with query: %s" % (path, query))

    try:
        # Handle the possible request paths
        if path == ROUTE_INDEX:
            response = self.route_index(path, query)
        elif path == ROUTE_VOICES:
            response = self.route_voices(path, query)
        elif path == ROUTE_READ:
            response = self.route_read(path, query)
        else:
            response = self.route_not_found(path, query)

        self.send_headers(response.status, response.content_type)
        self.stream_data(response.data_stream)

    except HTTPStatusError as err:
        # Respond with an error and log debug
        # information
        if sys.version_info >= (3, 0):
            self.send_error(err.code, err.message, err.explain)
        else:
            self.send_error(err.code, err.message)

        self.log_error(u"%s %s %s - [%d] %s", self.client_address[0],
                      self.command, self.path, err.code, err.explain)

    print("[END]")

def route_not_found(self, path, query):
    """Handles routing for unexpected paths"""
    raise HTTPStatusError(HTTP_STATUS["NOT_FOUND"], "Page not found")

```

```

def route_index(self, path, query):
    """Handles routing for the application's entry point"""
    try:
        return ResponseData(status=HTTP_STATUS["OK"], content_type="text_html",
                             # Open a binary stream for reading the index
                             # HTML file
                             data_stream=open(os.path.join(sys.path[0],
                                                             path[1:]), "rb"))
    except IOError as err:
        # Couldn't open the stream
        raise HTTPStatusError(HTTP_STATUS["INTERNAL_SERVER_ERROR"],
                               str(err))

def route_voices(self, path, query):
    """Handles routing for listing available voices"""
    params = {}
    voices = []

    while True:
        try:
            # Request list of available voices, if a continuation token
            # was returned by the previous call then use it to continue
            # listing
            response = polly.describe_voices(**params)
        except (BotoCoreError, ClientError) as err:
            # The service returned an error
            raise HTTPStatusError(HTTP_STATUS["INTERNAL_SERVER_ERROR"],
                                   str(err))

        # Collect all the voices
        voices.extend(response.get("Voices", []))

        # If a continuation token was returned continue, stop iterating
        # otherwise
        if "NextToken" in response:
            params = {"NextToken": response["NextToken"]}
        else:
            break

    json_data = json_encode(voices)
    bytes_data = bytes(json_data, "utf-8") if sys.version_info >= (3, 0) \
        else bytes(json_data)

```

```

        return ResponseData(status=HTTP_STATUS["OK"],
                             content_type="application/json",
                             # Create a binary stream for the JSON data
                             data_stream=BytesIO(bytes_data))

def route_read(self, path, query):
    """Handles routing for reading text (speech synthesis)"""
    # Get the parameters from the query string
    text = self.query_get(query, "text")
    voiceId = self.query_get(query, "voiceId")
    outputFormat = self.query_get(query, "outputFormat")

    # Validate the parameters, set error flag in case of unexpected
    # values
    if len(text) == 0 or len(voiceId) == 0 or \
        outputFormat not in AUDIO_FORMATS:
        raise HTTPStatusError(HTTP_STATUS["BAD_REQUEST"],
                              "Wrong parameters")
    else:
        try:
            # Request speech synthesis
            response = polly.synthesize_speech(Text=text,
                                                VoiceId=voiceId,
                                                OutputFormat=outputFormat,
                                                Engine="neural")

        except (BotoCoreError, ClientError) as err:
            # The service returned an error
            raise HTTPStatusError(HTTP_STATUS["INTERNAL_SERVER_ERROR"],
                                  str(err))

        return ResponseData(status=HTTP_STATUS["OK"],
                             content_type=AUDIO_FORMATS[outputFormat],
                             # Access the audio stream in the response
                             data_stream=response.get("AudioStream"))

def send_headers(self, status, content_type):
    """Send out the group of headers for a successful request"""
    # Send HTTP headers
    self.send_response(status.code, status.message)
    self.send_header('Content-type', content_type)
    self.send_header('Transfer-Encoding', 'chunked')
    self.send_header('Connection', 'close')
    self.end_headers()

```

```
def stream_data(self, stream):
    """Consumes a stream in chunks to produce the response's output"""
    print("Streaming started...")

    if stream:
        # Note: Closing the stream is important as the service throttles on
        # the number of parallel connections. Here we are using
        # contextlib.closing to ensure the close method of the stream object
        # will be called automatically at the end of the with statement's
        # scope.
        with closing(stream) as managed_stream:
            # Push out the stream's content in chunks
            while True:
                data = managed_stream.read(CHUNK_SIZE)
                self.wfile.write(b"%X\r\n%s\r\n" % (len(data), data))

                # If there's no more data to read, stop streaming
                if not data:
                    break

            # Ensure any buffered output has been transmitted and close the
            # stream
            self.wfile.flush()

        print("Streaming completed.")
    else:
        # The stream passed in is empty
        self.wfile.write(b"0\r\n\r\n")
        print("Nothing to stream.")

# Define and parse the command line arguments
cli = ArgumentParser(description='Example Python Application')
cli.add_argument(
    "-p", "--port", type=int, metavar="PORT", dest="port", default=8000)
cli.add_argument(
    "--host", type=str, metavar="HOST", dest="host", default="localhost")
arguments = cli.parse_args()

# If the module is invoked directly, initialize the application
if __name__ == '__main__':
    # Create and configure the HTTP server instance
    server = ThreadedHTTPServer((arguments.host, arguments.port),
                               ChunkedHTTPRequestHandler)
    print("Starting server, use <Ctrl-C> to stop...")
```

```
print(u"Open {0}://{1}:{2}{3} in a web browser.".format(PROTOCOL,
                                                         arguments.host,
                                                         arguments.port,
                                                         ROUTE_INDEX))

try:
    # Listen for requests indefinitely
    server.serve_forever()
except KeyboardInterrupt:
    # A request to terminate has been received, stop the server
    print("\nShutting down...")
    server.socket.close()
```

Contoh iOS

Contoh berikut menggunakan SDK iOS untuk Amazon Polly untuk membaca teks yang ditentukan menggunakan suara yang dipilih dari daftar suara.

Kode yang ditampilkan di sini mencakup tugas-tugas utama tetapi tidak menangani kesalahan. Untuk kode lengkapnya, lihat demo [AWS Mobile SDK for iOS Amazon Polly](#).

Inisialisasi

```
// 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()
```

Dapatkan Daftar Suara yang Tersedia

```
// 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
})

```

Sintesis Pidato

```

// 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
})
```

Contoh Android

Contoh berikut menggunakan Android SDK for Amazon Polly untuk membaca teks yang ditentukan menggunakan suara yang dipilih dari daftar suara.

Kode yang ditampilkan di sini mencakup tugas-tugas utama tetapi tidak menangani kesalahan. Untuk kode lengkapnya, lihat demo [AWS Mobile SDK for Android Amazon Polly](#).

Inisialisasi

```
// 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);
```

Dapatkan Daftar Suara yang Tersedia

```
// 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();
```

Dapatkan URL untuk Audio Stream

```
// 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);
```

Mainkan Pidato yang Disintesis

```
// 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();
    }
});
```

Contoh kode untuk Amazon Polly menggunakan AWS SDK

Contoh kode berikut menunjukkan cara menggunakan Amazon Polly dengan AWS perangkat pengembangan perangkat lunak (SDK).

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Skenario adalah contoh kode yang menunjukkan kepada Anda bagaimana menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan atau dikombinasikan dengan yang lain Layanan AWS.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan Amazon Polly dengan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Contoh kode

- [Contoh dasar untuk Amazon Polly menggunakan AWS SDK](#)
 - [Tindakan untuk Amazon Polly menggunakan AWS SDK](#)
 - [Gunakan DeleteLexicon dengan AWS SDK atau CLI](#)
 - [Gunakan DescribeVoices dengan AWS SDK](#)
 - [Gunakan GetLexicon dengan AWS SDK atau CLI](#)
 - [Gunakan GetSpeechSynthesisTask dengan AWS SDK atau CLI](#)
 - [Gunakan ListLexicons dengan AWS SDK atau CLI](#)
 - [Gunakan ListSpeechSynthesisTasks dengan AWS SDK atau CLI](#)
 - [Gunakan PutLexicon dengan AWS SDK atau CLI](#)
 - [Gunakan StartSpeechSynthesisTask dengan AWS SDK atau CLI](#)
 - [Gunakan SynthesizeSpeech dengan AWS SDK](#)
 - [Skenario untuk Amazon Polly menggunakan AWS SDK](#)
 - [Mengkonversi teks menjadi ucapan dan kembali ke teks menggunakan AWS SDK](#)
 - [Buat aplikasi lip-sync dengan Amazon Polly menggunakan AWS SDK](#)
 - [Buat aplikasi yang menganalisis umpan balik pelanggan dan mensintesis audio](#)
 - [Memulai dengan sintesis text-to-speech](#)

Contoh dasar untuk Amazon Polly menggunakan AWS SDK

Contoh kode berikut menunjukkan cara menggunakan dasar-dasar Amazon Polly dengan AWS SDK.

Contoh

- [Tindakan untuk Amazon Polly menggunakan AWS SDK](#)
 - [Gunakan DeleteLexicondengan AWS SDK atau CLI](#)
 - [Gunakan DescribeVoicesdengan AWS SDK](#)
 - [Gunakan GetLexicondengan AWS SDK atau CLI](#)
 - [Gunakan GetSpeechSynthesisTaskdengan AWS SDK atau CLI](#)
 - [Gunakan ListLexiconsdengan AWS SDK atau CLI](#)
 - [Gunakan ListSpeechSynthesisTasksdengan AWS SDK atau CLI](#)
 - [Gunakan PutLexicondengan AWS SDK atau CLI](#)
 - [Gunakan StartSpeechSynthesisTaskdengan AWS SDK atau CLI](#)
 - [Gunakan SynthesizeSpeechdengan AWS SDK](#)

Tindakan untuk Amazon Polly menggunakan AWS SDK

Contoh kode berikut menunjukkan cara melakukan tindakan Amazon Polly individual dengan AWS SDK. Setiap contoh menyertakan tautan ke GitHub, di mana Anda dapat menemukan instruksi untuk mengatur dan menjalankan kode.

Kutipan ini memanggil Amazon Polly API dan merupakan kutipan kode dari program yang lebih besar yang harus dijalankan dalam konteks. Anda dapat melihat tindakan dalam konteks di [Skenario untuk Amazon Polly menggunakan AWS SDK](#).

Contoh berikut hanya mencakup tindakan yang paling umum digunakan. Untuk daftar lengkapnya, lihat Referensi [API Amazon Polly](#).

Contoh

- [Gunakan DeleteLexicondengan AWS SDK atau CLI](#)
- [Gunakan DescribeVoicesdengan AWS SDK](#)
- [Gunakan GetLexicondengan AWS SDK atau CLI](#)
- [Gunakan GetSpeechSynthesisTaskdengan AWS SDK atau CLI](#)
- [Gunakan ListLexiconsdengan AWS SDK atau CLI](#)

- [Gunakan ListSpeechSynthesisTasksdengan AWS SDK atau CLI](#)
- [Gunakan PutLexicondengan AWS SDK atau CLI](#)
- [Gunakan StartSpeechSynthesisTaskdengan AWS SDK atau CLI](#)
- [Gunakan SynthesizeSpeechdengan AWS SDK](#)

Gunakan **DeleteLexicon** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DeleteLexicon`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Memulai dengan sintesis text-to-speech](#)

.NET

SDK untuk .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
using System;
using System.Threading.Tasks;
using Amazon.Polly;
using Amazon.Polly.Model;

/// <summary>
/// Deletes an existing Amazon Polly lexicon using the AWS SDK for .NET.
/// </summary>
public class DeleteLexicon
{
    public static async Task Main()
    {
        string lexiconName = "SampleLexicon";

        var client = new AmazonPollyClient();
```

```

        var success = await DeletePollyLexiconAsync(client, lexiconName);

        if (success)
        {
            Console.WriteLine($"Successfully deleted {lexiconName}.");
        }
        else
        {
            Console.WriteLine($"Could not delete {lexiconName}.");
        }
    }

    /// <summary>
    /// Deletes the named Amazon Polly lexicon.
    /// </summary>
    /// <param name="client">The initialized Amazon Polly client object.</
param>
    /// <param name="lexiconName">The name of the Amazon Polly lexicon to
    /// delete.</param>
    /// <returns>A Boolean value indicating the success of the operation.</
returns>
    public static async Task<bool> DeletePollyLexiconAsync(
        AmazonPollyClient client,
        string lexiconName)
    {
        var deleteLexiconRequest = new DeleteLexiconRequest()
        {
            Name = lexiconName,
        };

        var response = await client.DeleteLexiconAsync(deleteLexiconRequest);

        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
}

```

- Untuk detail API, lihat [DeleteLexicon](#) di Referensi AWS SDK untuk .NET API.

CLI

AWS CLI

Untuk menghapus leksikon

`delete-lexicon` Contoh berikut menghapus leksikon yang ditentukan.

```
aws polly delete-lexicon \  
  --name w3c
```

Perintah ini tidak menghasilkan output.

Untuk informasi selengkapnya, lihat [Menggunakan DeleteLexicon operasi](#) di Panduan Pengembang Amazon Polly.

- Untuk detail API, lihat [DeleteLexicon](#) di Referensi AWS CLI Perintah.

SAP ABAP

SDK for SAP ABAP

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
TRY.  
  lo_ply->deletelexicon( iv_name ).  
  MESSAGE 'Lexicon deleted successfully.' TYPE 'I'.  
CATCH /aws1/cx_plylexiconnotfoundex.  
  MESSAGE 'Lexicon not found.' TYPE 'E'.  
CATCH /aws1/cx_plyservicefailureex.  
  MESSAGE 'Service failure occurred.' TYPE 'E'.  
ENDTRY.
```

- Untuk detail API, lihat [DeleteLexicon](#) di AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan Amazon Polly dengan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DescribeVoices** dengan AWS SDK

Contoh kode berikut menunjukkan cara menggunakan `DescribeVoices`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Memulai dengan sintesis text-to-speech](#)

.NET

SDK untuk .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
using System;
using System.Threading.Tasks;
using Amazon.Polly;
using Amazon.Polly.Model;

public class DescribeVoices
{
    public static async Task Main()
    {
        var client = new AmazonPollyClient();

        var allVoicesRequest = new DescribeVoicesRequest();
        var enUSVoicesRequest = new DescribeVoicesRequest()
        {
            LanguageCode = "en-US",
        };

        try
```

```

        {
            string nextToken;
            do
            {
                var allVoicesResponse = await
client.DescribeVoicesAsync(allVoicesRequest);
                nextToken = allVoicesResponse.NextToken;
                allVoicesRequest.NextToken = nextToken;

                Console.WriteLine("\nAll voices: ");
                allVoicesResponse.Voices.ForEach(voice =>
                {
                    DisplayVoiceInfo(voice);
                });
            }
            while (nextToken is not null);

            do
            {
                var enUsVoicesResponse = await
client.DescribeVoicesAsync(enUsVoicesRequest);
                nextToken = enUsVoicesResponse.NextToken;
                enUsVoicesRequest.NextToken = nextToken;

                Console.WriteLine("\nen-US voices: ");
                enUsVoicesResponse.Voices.ForEach(voice =>
                {
                    DisplayVoiceInfo(voice);
                });
            }
            while (nextToken is not null);
        }
        catch (Exception ex)
        {
            Console.WriteLine("Exception caught: " + ex.Message);
        }
    }

    public static void DisplayVoiceInfo(Voice voice)
    {
        Console.WriteLine($" Name: {voice.Name}\tGender:
{voice.Gender}\tLanguageName: {voice.LanguageName}");
    }
}

```

- Untuk detail API, lihat [DescribeVoices](#) di Referensi AWS SDK untuk .NET API.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.polly.PollyClient;
import software.amazon.awssdk.services.polly.model.DescribeVoicesRequest;
import software.amazon.awssdk.services.polly.model.DescribeVoicesResponse;
import software.amazon.awssdk.services.polly.model.PollyException;
import software.amazon.awssdk.services.polly.model.Voice;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
public class DescribeVoicesSample {
    public static void main(String args[]) {
        PollyClient polly = PollyClient.builder()
            .region(Region.US_WEST_2)
            .build();

        describeVoice(polly);
        polly.close();
    }
}
```

```

public static void describeVoice(PollyClient polly) {
    try {
        DescribeVoicesRequest voicesRequest = DescribeVoicesRequest.builder()
            .languageCode("en-US")
            .build();

        DescribeVoicesResponse enUsVoicesResult =
polly.describeVoices(vocesRequest);
        List<Voice> voices = enUsVoicesResult.voices();
        for (Voice myVoice : voices) {
            System.out.println("The ID of the voice is " + myVoice.id());
            System.out.println("The gender of the voice is " +
myVoice.gender());
        }

        } catch (PollyException e) {
            System.err.println("Exception caught: " + e);
            System.exit(1);
        }
    }
}

```

- Untuk detail API, lihat [DescribeVoices](#) di Referensi AWS SDK for Java 2.x API.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

class PollyWrapper:
    """Encapsulates Amazon Polly functions."""

    def __init__(self, polly_client, s3_resource):
        """
        :param polly_client: A Boto3 Amazon Polly client.

```

```

        :param s3_resource: A Boto3 Amazon Simple Storage Service (Amazon S3)
resource.
        """
        self.polly_client = polly_client
        self.s3_resource = s3_resource
        self.voice_metadata = None

    def describe_voices(self):
        """
        Gets metadata about available voices.

        :return: The list of voice metadata.
        """
        try:
            response = self.polly_client.describe_voices()
            self.voice_metadata = response["Voices"]
            logger.info("Got metadata about %s voices.",
len(self.voice_metadata))
        except ClientError:
            logger.exception("Couldn't get voice metadata.")
            raise
        else:
            return self.voice_metadata

```

- Untuk detail API, lihat [DescribeVoices](#) di AWS SDK for Python (Boto3) Referensi API.

Ruby

SDK untuk Ruby

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
require 'aws-sdk-polly' # In v2: require 'aws-sdk'
```

```
begin
  # Create an Amazon Polly client using
  # credentials from the shared credentials file ~/.aws/credentials
  # and the configuration (region) from the shared configuration file ~/.aws/
  config
  polly = Aws::Polly::Client.new

  # Get US English voices
  resp = polly.describe_voices(language_code: 'en-US')

  resp.voices.each do |v|
    puts v.name
    puts "  #{v.gender}"
    puts
  end
rescue StandardError => e
  puts 'Could not get voices'
  puts 'Error message:'
  puts e.message
end
```

- Untuk detail API, lihat [DescribeVoices](#) di Referensi AWS SDK for Ruby API.

Rust

SDK for Rust

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
async fn list_voices(client: &Client) -> Result<(), Error> {
  let resp = client.describe_voices().send().await?;

  println!("Voices:");

  let voices = resp.voices();
  for voice in voices {
    println!("  Name:      {}", voice.name().unwrap_or("No name!"));
  }
}
```

```
println!(
    "  Language: {}",
    voice.language_name().unwrap_or("No language!")
);

println!();
}

println!("Found {} voices", voices.len());

Ok(())
}
```

- Untuk detail API, lihat [DescribeVoices](#) referensi AWS SDK for Rust API.

SAP ABAP

SDK for SAP ABAP

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
TRY.
    " Only pass optional parameters if they have values
    IF iv_engine IS NOT INITIAL AND iv_language IS NOT INITIAL.
        oo_result = lo_ply->describevoices(
            iv_engine = iv_engine
            iv_languagecode = iv_language ).
    ELSEIF iv_engine IS NOT INITIAL.
        oo_result = lo_ply->describevoices(
            iv_engine = iv_engine ).
    ELSEIF iv_language IS NOT INITIAL.
        oo_result = lo_ply->describevoices(
            iv_languagecode = iv_language ).
    ELSE.
        oo_result = lo_ply->describevoices( ).
    ENDIF.
    MESSAGE 'Retrieved voice metadata.' TYPE 'I'.
```

```
CATCH /aws1/cx_plyinvalidnexttokenex.
    MESSAGE 'The NextToken is invalid.' TYPE 'E'.
CATCH /aws1/cx_plyservicefailureex.
    MESSAGE 'Service failure occurred.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat [DescribeVoices](#) di AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan Amazon Polly dengan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **GetLexicon** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `GetLexicon`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Memulai dengan sintesis text-to-speech](#)

.NET

SDK untuk .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
using System;
using System.Threading.Tasks;
using Amazon.Polly;
using Amazon.Polly.Model;

/// <summary>
/// Retrieves information about a specific Amazon Polly lexicon.
/// </summary>
```

```
public class GetLexicon
{
    public static async Task Main(string[] args)
    {
        string lexiconName = "SampleLexicon";

        var client = new AmazonPollyClient();

        await GetPollyLexiconAsync(client, lexiconName);
    }

    public static async Task GetPollyLexiconAsync(AmazonPollyClient client,
string lexiconName)
    {
        var getLexiconRequest = new GetLexiconRequest()
        {
            Name = lexiconName,
        };

        try
        {
            var response = await client.GetLexiconAsync(getLexiconRequest);
            Console.WriteLine($"Lexicon:\n Name: {response.Lexicon.Name}");
            Console.WriteLine($"Content: {response.Lexicon.Content}");
        }
        catch (Exception ex)
        {
            Console.WriteLine("Error: " + ex.Message);
        }
    }
}
```

- Untuk detail API, lihat [GetLexicon](#) di Referensi AWS SDK untuk .NET API.

CLI

AWS CLI

Untuk mengambil konten leksikon

`get-lexicon` Contoh berikut mengambil isi dari leksikon pengucapan yang ditentukan.

```
aws polly get-lexicon \
  --name w3c
```

Output:

```
{
  "Lexicon": {
    "Content": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<lexicon version=
\"1.0\" \n      xmlns=      \"http://www.w3.org/2005/01/pronunciation-lexicon
\" \n      xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\" \n
xsi:schemaLocation=\"http://www.w3.org/2005/01/pronunciation-lexicon \n
http://www.w3.org/TR/2007/CR-pronunciation-lexicon-20071212/pls.xsd\" \n
  alphabet=\"ipa\" \n      xml:lang=\"en-US\">\n  <lexeme>\n    <grapheme>W3C</
grapheme>\n      <alias>World Wide Web Consortium</alias>\n  </lexeme>\n</
lexicon>\n",
    "Name": "w3c"
  },
  "LexiconAttributes": {
    "Alphabet": "ipa",
    "LanguageCode": "en-US",
    "LastModified": 1603908910.99,
    "LexiconArn": "arn:aws:polly:us-west-2:880185128111:lexicon/w3c",
    "LexemesCount": 1,
    "Size": 492
  }
}
```

Untuk informasi selengkapnya, lihat [Menggunakan GetLexicon operasi](#) di Panduan Pengembang Amazon Polly.

- Untuk detail API, lihat [GetLexicon](#) di Referensi AWS CLI Perintah.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
class PollyWrapper:
    """Encapsulates Amazon Polly functions."""

    def __init__(self, polly_client, s3_resource):
        """
        :param polly_client: A Boto3 Amazon Polly client.
        :param s3_resource: A Boto3 Amazon Simple Storage Service (Amazon S3)
        resource.
        """
        self.polly_client = polly_client
        self.s3_resource = s3_resource
        self.voice_metadata = None

    def get_lexicon(self, name):
        """
        Gets metadata and contents of an existing lexicon.

        :param name: The name of the lexicon to retrieve.
        :return: The retrieved lexicon.
        """
        try:
            response = self.polly_client.get_lexicon(Name=name)
            logger.info("Got lexicon %s.", name)
        except ClientError:
            logger.exception("Couldn't get lexicon %s.", name)
            raise
        else:
            return response
```

- Untuk detail API, lihat [GetLexicon](#) di AWS SDK for Python (Boto3) Referensi API.

SAP ABAP

SDK for SAP ABAP

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
TRY.
    oo_result = lo_ply->getlexicon( iv_name ).
    DATA(lo_lexicon) = oo_result->get_lexicon( ).
    IF lo_lexicon IS BOUND.
        DATA(lv_lex_name) = lo_lexicon->get_name( ).
        MESSAGE |Retrieved lexicon: { lv_lex_name }| TYPE 'I'.
    ENDIF.
    CATCH /aws1/cx_plylexiconnotfoundex.
        MESSAGE 'Lexicon not found.' TYPE 'E'.
    CATCH /aws1/cx_plyservicefailureex.
        MESSAGE 'Service failure occurred.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat [GetLexicon](#) di AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan Amazon Polly dengan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **GetSpeechSynthesisTask** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `GetSpeechSynthesisTask`.

CLI

AWS CLI

Untuk mendapatkan informasi tentang tugas sintesis ucapan

get-speech-synthesis-task Contoh berikut mengambil informasi tentang tugas sintesis ucapan yang ditentukan.

```
aws polly get-speech-synthesis-task \
  --task-id 70b61c0f-57ce-4715-a247-cae8729dcce9
```

Output:

```
{
  "SynthesisTask": {
    "TaskId": "70b61c0f-57ce-4715-a247-cae8729dcce9",
    "TaskStatus": "completed",
    "OutputUri": "https://s3.us-west-2.amazonaws.com/amzn-s3-demo-
bucket/70b61c0f-57ce-4715-a247-cae8729dcce9.mp3",
    "CreationTime": 1603911042.689,
    "RequestCharacters": 1311,
    "OutputFormat": "mp3",
    "TextType": "text",
    "VoiceId": "Joanna"
  }
}
```

Untuk informasi selengkapnya, lihat [Membuat file audio panjang](#) di Panduan Pengembang Amazon Polly.

- Untuk detail API, lihat [GetSpeechSynthesisTask](#) di Referensi AWS CLI Perintah.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
class PollyWrapper:
    """Encapsulates Amazon Polly functions."""
```

```

def __init__(self, polly_client, s3_resource):
    """
    :param polly_client: A Boto3 Amazon Polly client.
    :param s3_resource: A Boto3 Amazon Simple Storage Service (Amazon S3)
resource.
    """
    self.polly_client = polly_client
    self.s3_resource = s3_resource
    self.voice_metadata = None

def get_speech_synthesis_task(self, task_id):
    """
    Gets metadata about an asynchronous speech synthesis task, such as its
status.

    :param task_id: The ID of the task to retrieve.
    :return: Metadata about the task.
    """
    try:
        response =
self.polly_client.get_speech_synthesis_task(TaskId=task_id)
        task = response["SynthesisTask"]
        logger.info("Got synthesis task. Status is %s.", task["TaskStatus"])
    except ClientError:
        logger.exception("Couldn't get synthesis task %s.", task_id)
        raise
    else:
        return task

```

- Untuk detail API, lihat [GetSpeechSynthesisTask](#) di AWS SDK for Python (Boto3) Referensi API.

SAP ABAP

SDK for SAP ABAP

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
TRY.  
    oo_result = lo_ply->getspeechsynthesistask( iv_task_id ).  
    DATA(lo_task) = oo_result->get_synthesistask( ).  
    IF lo_task IS BOUND.  
        DATA(lv_status) = lo_task->get_taskstatus( ).  
        MESSAGE |Task status: { lv_status }| TYPE 'I'.  
    ENDIF.  
CATCH /aws1/cx_plyinvalidtaskidex.  
    MESSAGE 'Invalid task ID.' TYPE 'E'.  
CATCH /aws1/cx_plyservicefailureex.  
    MESSAGE 'Service failure occurred.' TYPE 'E'.  
CATCH /aws1/cx_plysynthesistsknotf00.  
    MESSAGE 'Synthesis task not found.' TYPE 'E'.  
ENDTRY.
```

- Untuk detail API, lihat [GetSpeechSynthesisTask](#) di AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan Amazon Polly dengan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **ListLexicons** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `ListLexicons`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Memulai dengan sintesis text-to-speech](#)

.NET

SDK untuk .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
using System;
using System.Threading.Tasks;
using Amazon.Polly;
using Amazon.Polly.Model;

/// <summary>
/// Lists the Amazon Polly lexicons that have been defined. By default,
/// lists the lexicons that are defined in the same AWS Region as the default
/// user. To view Amazon Polly lexicons that are defined in a different AWS
/// Region, supply it as a parameter to the Amazon Polly constructor.
/// </summary>
public class ListLexicons
{
    public static async Task Main()
    {
        var client = new AmazonPollyClient();
        var request = new ListLexiconsRequest();

        try
        {
            Console.WriteLine("All voices: ");

            do
            {
                var response = await client.ListLexiconsAsync(request);
                request.NextToken = response.NextToken;

                response.Lexicons.ForEach(lexicon =>
                {
                    var attributes = lexicon.Attributes;
                    Console.WriteLine($"Name: {lexicon.Name}");
                    Console.WriteLine($"\\tAlphabet: {attributes.Alphabet}");
                });
            } while (request.NextToken != null);
        }
        catch { }
    }
}
```

```

        Console.WriteLine($"\\tLanguageCode:
{attributes.LanguageCode}");
        Console.WriteLine($"\\tLastModified:
{attributes.LastModified}");
        Console.WriteLine($"\\tLexemesCount:
{attributes.LexemesCount}");
        Console.WriteLine($"\\tLexiconArn:
{attributes.LexiconArn}");
        Console.WriteLine($"\\tSize: {attributes.Size}");
    });
}
while (request.NextToken is not null);
}
catch (Exception ex)
{
    Console.WriteLine($"Error: {ex.Message}");
}
}
}

```

- Untuk detail API, lihat [ListLexicons](#) di Referensi AWS SDK untuk .NET API.

CLI

AWS CLI

Untuk membuat daftar leksikon Anda

`list-lexicons` Contoh berikut mencantumkan leksikon pengucapan Anda.

```
aws polly list-lexicons
```

Output:

```

{
  "Lexicons": [
    {
      "Name": "w3c",
      "Attributes": {
        "Alphabet": "ipa",
        "LanguageCode": "en-US",

```

```

        "LastModified": 1603908910.99,
        "LexiconArn": "arn:aws:polly:us-east-2:123456789012:lexicon/w3c",
        "LexemesCount": 1,
        "Size": 492
    }
}
]
}

```

Untuk informasi selengkapnya, lihat [Menggunakan ListLexicons operasi](#) di Panduan Pengembang Amazon Polly.

- Untuk detail API, lihat [ListLexicons](#) di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.polly.PollyClient;
import software.amazon.awssdk.services.polly.model.ListLexiconsResponse;
import software.amazon.awssdk.services.polly.model.ListLexiconsRequest;
import software.amazon.awssdk.services.polly.model.LexiconDescription;
import software.amazon.awssdk.services.polly.model.PollyException;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
public class ListLexicons {

```

```

public static void main(String args[]) {
    PollyClient polly = PollyClient.builder()
        .region(Region.US_WEST_2)
        .build();

    listLexicons(polly);
    polly.close();
}

public static void listLexicons(PollyClient client) {
    try {
        ListLexiconsRequest listLexiconsRequest =
ListLexiconsRequest.builder()
            .build();

        ListLexiconsResponse listLexiconsResult =
client.listLexicons(listLexiconsRequest);
        List<LexiconDescription> lexiconDescription =
listLexiconsResult.lexicons();
        for (LexiconDescription lexDescription : lexiconDescription) {
            System.out.println("The name of the Lexicon is " +
lexDescription.name());
        }

    } catch (PollyException e) {
        System.err.println("Exception caught: " + e);
        System.exit(1);
    }
}
}

```

- Untuk detail API, lihat [ListLexicons](#) di Referensi AWS SDK for Java 2.x API.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
class PollyWrapper:
    """Encapsulates Amazon Polly functions."""

    def __init__(self, polly_client, s3_resource):
        """
        :param polly_client: A Boto3 Amazon Polly client.
        :param s3_resource: A Boto3 Amazon Simple Storage Service (Amazon S3)
        resource.
        """
        self.polly_client = polly_client
        self.s3_resource = s3_resource
        self.voice_metadata = None

    def list_lexicons(self):
        """
        Lists lexicons in the current account.

        :return: The list of lexicons.
        """
        try:
            response = self.polly_client.list_lexicons()
            lexicons = response["Lexicons"]
            logger.info("Got %s lexicons.", len(lexicons))
        except ClientError:
            logger.exception(
                "Couldn't get %s.",
            )
            raise
        else:
            return lexicons
```

- Untuk detail API, lihat [ListLexicons](#) di AWS SDK for Python (Boto3) Referensi API.

Ruby

SDK untuk Ruby

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
require 'aws-sdk-polly' # In v2: require 'aws-sdk'

begin
  # Create an Amazon Polly client using
  # credentials from the shared credentials file ~/.aws/credentials
  # and the configuration (region) from the shared configuration file ~/.aws/
  config
  polly = Aws::Polly::Client.new

  resp = polly.list_lexicons

  resp.lexicons.each do |l|
    puts l.name
    puts "  Alphabet:#{l.attributes.alphabet}"
    puts "  Language:#{l.attributes.language}"
    puts
  end
rescue StandardError => e
  puts 'Could not get lexicons'
  puts 'Error message:'
  puts e.message
end
```

- Untuk detail API, lihat [ListLexicons](#) di Referensi AWS SDK for Ruby API.

Rust

SDK for Rust

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
async fn show_lexicons(client: &Client) -> Result<(), Error> {
    let resp = client.list_lexicons().send().await?;

    println!("Lexicons:");

    let lexicons = resp.lexicons();

    for lexicon in lexicons {
        println!("  Name:      {}", lexicon.name().unwrap_or_default());
        println!(
            "    Language: {:?}\n",
            lexicon
                .attributes()
                .as_ref()
                .map(|attrib| attrib
                    .language_code
                    .as_ref()
                    .expect("languages must have language codes"))
                .expect("languages must have attributes")
        );
    }

    println();
    println!("Found {} lexicons.", lexicons.len());
    println();

    Ok(())
}
```

- Untuk detail API, lihat [ListLexicons](#) referensi AWS SDK for Rust API.

SAP ABAP

SDK for SAP ABAP

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
TRY.
    oo_result = lo_ply->listlexicons( ).
    DATA(lt_lexicons) = oo_result->get_lexicons( ).
    DATA(lv_count) = lines( lt_lexicons ).
    MESSAGE |Found { lv_count } lexicons| TYPE 'I'.
CATCH /aws1/cx_plyinvalidnexttokenex.
    MESSAGE 'Invalid NextToken.' TYPE 'E'.
CATCH /aws1/cx_plyservicefailureex.
    MESSAGE 'Service failure occurred.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat [ListLexicons](#) di AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan Amazon Polly dengan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **ListSpeechSynthesisTasks** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `ListSpeechSynthesisTasks`.

CLI

AWS CLI

Untuk membuat daftar tugas sintesis ucapan Anda

`list-speech-synthesis-tasks` Contoh berikut mencantumkan tugas sintesis ucapan Anda.

aws polly list-speech-synthesis-tasks

Output:

```
{
  "SynthesisTasks": [
    {
      "TaskId": "70b61c0f-57ce-4715-a247-cae8729dcce9",
      "TaskStatus": "completed",
      "OutputUri": "https://s3.us-west-2.amazonaws.com/amzn-s3-demo-
bucket/70b61c0f-57ce-4715-a247-cae8729dcce9.mp3",
      "CreationTime": 1603911042.689,
      "RequestCharacters": 1311,
      "OutputFormat": "mp3",
      "TextType": "text",
      "VoiceId": "Joanna"
    }
  ]
}
```

Untuk informasi selengkapnya, lihat [Membuat file audio panjang](#) di Panduan Pengembang Amazon Polly.

- Untuk detail API, lihat [ListSpeechSynthesisTasks](#) di Referensi AWS CLI Perintah.

SAP ABAP

SDK for SAP ABAP

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

TRY.

```
" Only pass optional parameters if they have values
IF iv_max_results IS NOT INITIAL AND iv_status IS NOT INITIAL.
  oo_result = lo_ply->listspeechsynthesistasks(
    iv_maxresults = iv_max_results
```

```

        iv_status = iv_status ).
    ELSEIF iv_max_results IS NOT INITIAL.
        oo_result = lo_ply->listspeechsynthesistasks(
            iv_maxresults = iv_max_results ).
    ELSEIF iv_status IS NOT INITIAL.
        oo_result = lo_ply->listspeechsynthesistasks(
            iv_status = iv_status ).
    ELSE.
        oo_result = lo_ply->listspeechsynthesistasks( ).
    ENDIF.
    DATA(lt_tasks) = oo_result->get_synthesistasks( ).
    DATA(lv_count) = lines( lt_tasks ).
    MESSAGE |Found { lv_count } synthesis tasks| TYPE 'I'.
CATCH /aws1/cx_plyinvalidnexttokenex.
    MESSAGE 'Invalid NextToken.' TYPE 'E'.
CATCH /aws1/cx_plyservicefailureex.
    MESSAGE 'Service failure occurred.' TYPE 'E'.
ENDTRY.

```

- Untuk detail API, lihat [ListSpeechSynthesisTasks](#) di AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan Amazon Polly dengan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **PutLexicon** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan **PutLexicon**.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Memulai dengan sintesis text-to-speech](#)

.NET

SDK untuk .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
using System;
using System.Threading.Tasks;
using Amazon.Polly;
using Amazon.Polly.Model;

/// <summary>
/// Creates a new Amazon Polly lexicon using the AWS SDK for .NET.
/// </summary>
public class PutLexicon
{
    public static async Task Main()
    {
        string lexiconContent = "<?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>";
        string lexiconName = "SampleLexicon";

        var client = new AmazonPollyClient();
        var putLexiconRequest = new PutLexiconRequest()
        {
            Name = lexiconName,
            Content = lexiconContent,
        };
    }
}
```

```

        try
        {
            var response = await client.PutLexiconAsync(putLexiconRequest);
            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"Successfully created Lexicon:
{lexiconName}.");
            }
            else
            {
                Console.WriteLine($"Could not create Lexicon:
{lexiconName}.");
            }
        }
        catch (Exception ex)
        {
            Console.WriteLine("Exception caught: " + ex.Message);
        }
    }
}

```

- Untuk detail API, lihat [PutLexicon](#) di Referensi AWS SDK untuk .NET API.

CLI

AWS CLI

Untuk menyimpan leksikon

put-lexicon Contoh berikut menyimpan leksikon pengucapan yang ditentukan.
example.plsFile menentukan leksikon W3C PLS-compliant .

```

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

```

Isi dari example.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>
}

```

Perintah ini tidak menghasilkan output.

Untuk informasi selengkapnya, lihat [Menggunakan PutLexicon operasi](#) di Panduan Pengembang Amazon Polly.

- Untuk detail API, lihat [PutLexicon](#) di Referensi AWS CLI Perintah.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

class PollyWrapper:
    """Encapsulates Amazon Polly functions."""

    def __init__(self, polly_client, s3_resource):
        """
        :param polly_client: A Boto3 Amazon Polly client.
        :param s3_resource: A Boto3 Amazon Simple Storage Service (Amazon S3)
        resource.
        """
        self.polly_client = polly_client
        self.s3_resource = s3_resource

```

```

self.voice_metadata = None

def create_lexicon(self, name, content):
    """
    Creates a lexicon with the specified content. A lexicon contains custom
    pronunciations.

    :param name: The name of the lexicon.
    :param content: The content of the lexicon.
    """
    try:
        self.polly_client.put_lexicon(Name=name, Content=content)
        logger.info("Created lexicon %s.", name)
    except ClientError:
        logger.exception("Couldn't create lexicon %s.")
        raise

```

- Untuk detail API, lihat [PutLexicon](#) di AWS SDK for Python (Boto3) Referensi API.

Rust

SDK for Rust

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

async fn make_lexicon(client: &Client, name: &str, from: &str, to: &str) ->
    Result<(), Error> {
    let content = format!("<?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>{}</grapheme><alias>{}</alias></lexeme>
    </lexicon>", from, to);

```

```

client
    .put_lexicon()
    .name(name)
    .content(content)
    .send()
    .await?;

println!("Added lexicon");

Ok(())
}

```

- Untuk detail API, lihat [PutLexicon](#) referensi AWS SDK for Rust API.

SAP ABAP

SDK for SAP ABAP

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

TRY.
    lo_ply->putlexicon(
        iv_name = iv_name
        iv_content = iv_content ).
    MESSAGE 'Lexicon created successfully.' TYPE 'I'.
CATCH /aws1/cx_plyinvalidlexiconex.
    MESSAGE 'Invalid lexicon.' TYPE 'E'.
CATCH /aws1/cx_plylexiconsizeexcdex.
    MESSAGE 'Lexicon size exceeded.' TYPE 'E'.
CATCH /aws1/cx_plymaxlexemelengthe00.
    MESSAGE 'Maximum lexeme length exceeded.' TYPE 'E'.
CATCH /aws1/cx_plymaxlexiconsnoexc00.
    MESSAGE 'Maximum number of lexicons exceeded.' TYPE 'E'.
CATCH /aws1/cx_plyservicefailureex.
    MESSAGE 'Service failure occurred.' TYPE 'E'.
CATCH /aws1/cx_plyunsuppedplsalpha00.

```

```
MESSAGE 'Unsupported PLS alphabet.' TYPE 'E'.
CATCH /aws1/cx_plyunsuppdp1slangu00.
MESSAGE 'Unsupported PLS language.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat [PutLexicon](#) di AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan Amazon Polly dengan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **StartSpeechSynthesisTask** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan **StartSpeechSynthesisTask**.

CLI

AWS CLI

Untuk mensintesis teks

start-speech-synthesis-task Contoh berikut mensintesis teks `text_file.txt` dan menyimpan file MP3 yang dihasilkan dalam bucket yang ditentukan.

```
aws polly start-speech-synthesis-task \
  --output-format mp3 \
  --output-s3-bucket-name amzn-s3-demo-bucket \
  --text file://text_file.txt \
  --voice-id Joanna
```

Output:

```
{
  "SynthesisTask": {
    "TaskId": "70b61c0f-57ce-4715-a247-cae8729dcce9",
    "TaskStatus": "scheduled",
    "OutputUri": "https://s3.us-east-2.amazonaws.com/amzn-s3-demo-
bucket/70b61c0f-57ce-4715-a247-cae8729dcce9.mp3",
    "CreationTime": 1603911042.689,
    "RequestCharacters": 1311,
    "OutputFormat": "mp3",
```

```

        "TextType": "text",
        "VoiceId": "Joanna"
    }
}

```

Untuk informasi selengkapnya, lihat [Membuat file audio panjang](#) di Panduan Pengembang Amazon Polly.

- Untuk detail API, lihat [StartSpeechSynthesisTask](#) di Referensi AWS CLI Perintah.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

class PollyWrapper:
    """Encapsulates Amazon Polly functions."""

    def __init__(self, polly_client, s3_resource):
        """
        :param polly_client: A Boto3 Amazon Polly client.
        :param s3_resource: A Boto3 Amazon Simple Storage Service (Amazon S3)
        resource.
        """
        self.polly_client = polly_client
        self.s3_resource = s3_resource
        self.voice_metadata = None

    def do_synthesis_task(
        self,
        text,
        engine,
        voice,
        audio_format,
        s3_bucket,
        lang_code=None,

```

```

        include_visemes=False,
        wait_callback=None,
    ):
        """
        Start an asynchronous task to synthesize speech or speech marks, wait for
        the task to complete, retrieve the output from Amazon S3, and return the
        data.

        An asynchronous task is required when the text is too long for near-real
        time
        synthesis.

        :param text: The text to synthesize.
        :param engine: The kind of engine used. Can be standard or neural.
        :param voice: The ID of the voice to use.
        :param audio_format: The audio format to return for synthesized speech.
    When
        speech marks are synthesized, the output format is
    JSON.
        :param s3_bucket: The name of an existing Amazon S3 bucket that you have
        write access to. Synthesis output is written to this
    bucket.
        :param lang_code: The language code of the voice to use. This has an
    effect
        only when a bilingual voice is selected.
        :param include_visemes: When True, a second request is made to Amazon
    Polly
        to synthesize a list of visemes, using the
    specified
        text and voice. A viseme represents the visual
    position
        of the face and mouth when saying part of a word.
        :param wait_callback: A callback function that is called periodically
    during
        task processing, to give the caller an opportunity
    to
        take action, such as to display status.
        :return: The audio stream that contains the synthesized speech and a list
        of visemes that are associated with the speech audio.
        """
    try:
        kwargs = {
            "Engine": engine,
            "OutputFormat": audio_format,

```

```

        "OutputS3BucketName": s3_bucket,
        "Text": text,
        "VoiceId": voice,
    }
    if lang_code is not None:
        kwargs["LanguageCode"] = lang_code
    response = self.polly_client.start_speech_synthesis_task(**kwargs)
    speech_task = response["SynthesisTask"]
    logger.info("Started speech synthesis task %s.",
speech_task["TaskId"])

    viseme_task = None
    if include_visemes:
        kwargs["OutputFormat"] = "json"
        kwargs["SpeechMarkTypes"] = ["viseme"]
        response =
self.polly_client.start_speech_synthesis_task(**kwargs)
        viseme_task = response["SynthesisTask"]
        logger.info("Started viseme synthesis task %s.",
viseme_task["TaskId"])
    except ClientError:
        logger.exception("Couldn't start synthesis task.")
        raise
    else:
        bucket = self.s3_resource.Bucket(s3_bucket)
        audio_stream = self._wait_for_task(
            10, speech_task["TaskId"], "speech", wait_callback, bucket
        )

        visemes = None
        if include_visemes:
            viseme_data = self._wait_for_task(
                10, viseme_task["TaskId"], "viseme", wait_callback, bucket
            )
            visemes = [
                json.loads(v) for v in viseme_data.read().decode().split() if
v
            ]

        return audio_stream, visemes

```

- Untuk detail API, lihat [StartSpeechSynthesisTask](#) di AWS SDK for Python (Boto3) Referensi API.

SAP ABAP

SDK for SAP ABAP

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
TRY.
    " Only pass optional parameters if they have values
    IF iv_lang_code IS NOT INITIAL AND iv_s3_key_prefix IS NOT INITIAL.
        oo_result = lo_ply->startspeechsynthesistask(
            iv_engine = iv_engine
            iv_outputformat = iv_audio_format
            iv_outputs3bucketname = iv_s3_bucket
            iv_outputs3keyprefix = iv_s3_key_prefix
            iv_text = iv_text
            iv_voiceid = iv_voice_id
            iv_languagecode = iv_lang_code ).
    ELSEIF iv_lang_code IS NOT INITIAL.
        oo_result = lo_ply->startspeechsynthesistask(
            iv_engine = iv_engine
            iv_outputformat = iv_audio_format
            iv_outputs3bucketname = iv_s3_bucket
            iv_text = iv_text
            iv_voiceid = iv_voice_id
            iv_languagecode = iv_lang_code ).
    ELSEIF iv_s3_key_prefix IS NOT INITIAL.
        oo_result = lo_ply->startspeechsynthesistask(
            iv_engine = iv_engine
            iv_outputformat = iv_audio_format
            iv_outputs3bucketname = iv_s3_bucket
            iv_outputs3keyprefix = iv_s3_key_prefix
            iv_text = iv_text
            iv_voiceid = iv_voice_id ).
    ELSE.
```

```

        oo_result = lo_ply->startspeechsynthesistask(
            iv_engine = iv_engine
            iv_outputformat = iv_audio_format
            iv_outputs3bucketname = iv_s3_bucket
            iv_text = iv_text
            iv_voiceid = iv_voice_id ).
    ENDIF.
    MESSAGE 'Speech synthesis task started.' TYPE 'I'.
CATCH /aws1/cx_plyinvalids3bucketex.
    MESSAGE 'Invalid S3 bucket.' TYPE 'E'.
CATCH /aws1/cx_plyinvalidssmlex.
    MESSAGE 'Invalid SSML.' TYPE 'E'.
CATCH /aws1/cx_plylexiconnotfoundex.
    MESSAGE 'Lexicon not found.' TYPE 'E'.
CATCH /aws1/cx_plyservicefailureex.
    MESSAGE 'Service failure occurred.' TYPE 'E'.
CATCH /aws1/cx_plytextlengthexcdex.
    MESSAGE 'Text length exceeded maximum.' TYPE 'E'.
ENDTRY.

```

- Untuk detail API, lihat [StartSpeechSynthesisTask](#) di AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan Amazon Polly dengan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **SynthesizeSpeech** dengan AWS SDK

Contoh kode berikut menunjukkan cara menggunakan `SynthesizeSpeech`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Memulai dengan sintesis text-to-speech](#)

.NET

SDK untuk .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
using System;
using System.IO;
using System.Threading.Tasks;
using Amazon.Polly;
using Amazon.Polly.Model;

public class SynthesizeSpeech
{
    public static async Task Main()
    {
        string outputFileName = "speech.mp3";
        string text = "Twas brillig, and the slithy toves did gyre and gimbol
in the wabe";

        var client = new AmazonPollyClient();
        var response = await PollySynthesizeSpeech(client, text);

        WriteSpeechToStream(response.AudioStream, outputFileName);
    }

    /// <summary>
    /// Calls the Amazon Polly SynthesizeSpeechAsync method to convert text
    /// to speech.
    /// </summary>
    /// <param name="client">The Amazon Polly client object used to connect
    /// to the Amazon Polly service.</param>
    /// <param name="text">The text to convert to speech.</param>
    /// <returns>A SynthesizeSpeechResponse object that includes an
    AudioStream
    /// object with the converted text.</returns>
    private static async Task<SynthesizeSpeechResponse>
PollySynthesizeSpeech(IAmazonPolly client, string text)
```

```
{
    var synthesizeSpeechRequest = new SynthesizeSpeechRequest()
    {
        OutputFormat = OutputFormat.Mp3,
        VoiceId = VoiceId.Joanna,
        Text = text,
    };

    var synthesizeSpeechResponse =
        await client.SynthesizeSpeechAsync(synthesizeSpeechRequest);

    return synthesizeSpeechResponse;
}

/// <summary>
/// Writes the AudioStream returned from the call to
/// SynthesizeSpeechAsync to a file in MP3 format.
/// </summary>
/// <param name="audioStream">The AudioStream returned from the
/// call to the SynthesizeSpeechAsync method.</param>
/// <param name="outputFileName">The full path to the file in which to
/// save the audio stream.</param>
private static void WriteSpeechToStream(Stream audioStream, string
outputFileName)
{
    var outputStream = new FileStream(
        outputFileName,
        FileMode.Create,
        FileAccess.Write);
    byte[] buffer = new byte[2 * 1024];
    int readBytes;

    while ((readBytes = audioStream.Read(buffer, 0, 2 * 1024)) > 0)
    {
        outputStream.Write(buffer, 0, readBytes);
    }

    // Flushes the buffer to avoid losing the last second or so of
    // the synthesized text.
    outputStream.Flush();
    Console.WriteLine($"Saved {outputFileName} to disk.");
}
}
```

Sintesis ucapan dari teks menggunakan tanda ucapan dengan Amazon Polly menggunakan AWS SDK.

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Threading.Tasks;
using Amazon.Polly;
using Amazon.Polly.Model;

public class SynthesizeSpeechMarks
{
    public static async Task Main()
    {
        var client = new AmazonPollyClient();
        string outputFileName = "speechMarks.json";

        var synthesizeSpeechRequest = new SynthesizeSpeechRequest()
        {
            OutputFormat = OutputFormat.Json,
            SpeechMarkTypes = new List<string>
            {
                SpeechMarkType.Viseme,
                SpeechMarkType.Word,
            },
            VoiceId = VoiceId.Joanna,
            Text = "This is a sample text to be synthesized.",
        };

        try
        {
            using (var outputStream = new FileStream(outputFileName,
                FileMode.Create, FileAccess.Write))
            {
                var synthesizeSpeechResponse = await
                    client.SynthesizeSpeechAsync(synthesizeSpeechRequest);
                var buffer = new byte[2 * 1024];
                int readBytes;

                var inputStream = synthesizeSpeechResponse.AudioStream;
```

```

        while ((readBytes = inputStream.Read(buffer, 0, 2 * 1024)) >
0)
        {
            outputStream.Write(buffer, 0, readBytes);
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error: {ex.Message}");
    }
}

```

- Untuk detail API, lihat [SynthesizeSpeech](#) di Referensi AWS SDK untuk .NET API.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

import javazoom.jl.decoder.JavaLayerException;
import software.amazon.awssdk.core.ResponseInputStream;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.polly.PollyClient;
import software.amazon.awssdk.services.polly.model.DescribeVoicesRequest;
import software.amazon.awssdk.services.polly.model.Voice;
import software.amazon.awssdk.services.polly.model.DescribeVoicesResponse;
import software.amazon.awssdk.services.polly.model.OutputFormat;
import software.amazon.awssdk.services.polly.model.PollyException;
import software.amazon.awssdk.services.polly.model.SynthesizeSpeechRequest;
import software.amazon.awssdk.services.polly.model.SynthesizeSpeechResponse;
import java.io.IOException;
import java.io.InputStream;
import javazoom.jl.player.advanced.AdvancedPlayer;

```

```
import javazoom.jl.player.advanced.PlaybackEvent;
import javazoom.jl.player.advanced.PlaybackListener;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class PollyDemo {
    private static final String SAMPLE = "Congratulations. You have successfully
        built this working demo " +
        " of Amazon Polly in Java Version 2. 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 static void main(String args[]) {
        PollyClient polly = PollyClient.builder()
            .region(Region.US_WEST_2)
            .build();

        talkPolly(polly);
        polly.close();
    }

    public static void talkPolly(PollyClient polly) {
        try {
            DescribeVoicesRequest describeVoiceRequest =
                DescribeVoicesRequest.builder()
                    .engine("standard")
                    .build();

            DescribeVoicesResponse describeVoicesResult =
                polly.describeVoices(describeVoiceRequest);
            Voice voice = describeVoicesResult.voices().stream()
                .filter(v -> v.name().equals("Joanna"))
                .findFirst()
                .orElseThrow(() -> new RuntimeException("Voice not found"));
        }
    }
}
```

```

        InputStream stream = synthesize(polly, SAMPLE, voice,
        OutputFormat.MP3);
        AdvancedPlayer player = new AdvancedPlayer(stream,

javazoom.jl.player.FactoryRegistry.systemRegistry().createAudioDevice());
        player.setPlaybackListener(new PlaybackListener() {
            public void playbackStarted(PlaybackEvent evt) {
                System.out.println("Playback started");
                System.out.println(SAMPLE);
            }

            public void playbackFinished(PlaybackEvent evt) {
                System.out.println("Playback finished");
            }
        });

        // play it!
        player.play();

    } catch (PollyException | JavaLayerException | IOException e) {
        System.err.println(e.getMessage());
        System.exit(1);
    }
}

public static InputStream synthesize(PollyClient polly, String text, Voice
voice, OutputFormat format)
    throws IOException {
    SynthesizeSpeechRequest synthReq = SynthesizeSpeechRequest.builder()
        .text(text)
        .voiceId(voice.id())
        .outputFormat(format)
        .build();

    ResponseInputStream<SynthesizeSpeechResponse> synthRes =
polly.synthesizeSpeech(synthReq);
    return synthRes;
}
}

```

- Untuk detail API, lihat [SynthesizeSpeech](#) di Referensi AWS SDK for Java 2.x API.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
class PollyWrapper:
    """Encapsulates Amazon Polly functions."""

    def __init__(self, polly_client, s3_resource):
        """
        :param polly_client: A Boto3 Amazon Polly client.
        :param s3_resource: A Boto3 Amazon Simple Storage Service (Amazon S3)
        resource.
        """
        self.polly_client = polly_client
        self.s3_resource = s3_resource
        self.voice_metadata = None

    def synthesize(
        self, text, engine, voice, audio_format, lang_code=None,
        include_visemes=False
    ):
        """
        Synthesizes speech or speech marks from text, using the specified voice.

        :param text: The text to synthesize.
        :param engine: The kind of engine used. Can be standard or neural.
        :param voice: The ID of the voice to use.
        :param audio_format: The audio format to return for synthesized speech.
        When
                               speech marks are synthesized, the output format is
        JSON.
        :param lang_code: The language code of the voice to use. This has an
        effect
                               only when a bilingual voice is selected.
```

```

        :param include_visemes: When True, a second request is made to Amazon
Polly
                                to synthesize a list of visemes, using the
specified
                                text and voice. A viseme represents the visual
position
                                of the face and mouth when saying part of a word.
:return: The audio stream that contains the synthesized speech and a list
        of visemes that are associated with the speech audio.
"""
try:
    kwargs = {
        "Engine": engine,
        "OutputFormat": audio_format,
        "Text": text,
        "VoiceId": voice,
    }
    if lang_code is not None:
        kwargs["LanguageCode"] = lang_code
    response = self.polly_client.synthesize_speech(**kwargs)
    audio_stream = response["AudioStream"]
    logger.info("Got audio stream spoken by %s.", voice)
    visemes = None
    if include_visemes:
        kwargs["OutputFormat"] = "json"
        kwargs["SpeechMarkTypes"] = ["viseme"]
        response = self.polly_client.synthesize_speech(**kwargs)
        visemes = [
            json.loads(v)
            for v in response["AudioStream"].read().decode().split()
            if v
        ]
        logger.info("Got %s visemes.", len(visemes))
except ClientError:
    logger.exception("Couldn't get audio stream.")
    raise
else:
    return audio_stream, visemes

```

- Untuk detail API, lihat [SynthesizeSpeech](#) di AWS SDK for Python (Boto3) Referensi API.

Ruby

SDK untuk Ruby

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
require 'aws-sdk-polly' # In v2: require 'aws-sdk'

begin
  # Get the filename from the command line
  if ARGV.empty?
    puts 'You must supply a filename'
    exit 1
  end

  filename = ARGV[0]

  # Open file and get the contents as a string
  if File.exist?(filename)
    contents = IO.read(filename)
  else
    puts "No such file: #{filename}"
    exit 1
  end

  # Create an Amazon Polly client using
  # credentials from the shared credentials file ~/.aws/credentials
  # and the configuration (region) from the shared configuration file ~/.aws/
  config
  polly = Aws::Polly::Client.new

  resp = polly.synthesize_speech({
    output_format: 'mp3',
    text: contents,
    voice_id: 'Joanna'
  })
```

```

# Save output
# Get just the file name
# abc/xyz.txt -> xyx.txt
name = File.basename(filename)

# Split up name so we get just the xyz part
parts = name.split('.')
first_part = parts[0]
mp3_file = "#{first_part}.mp3"

IO.copy_stream(resp.audio_stream, mp3_file)

puts "Wrote MP3 content to: #{mp3_file}"
rescue StandardError => e
  puts 'Got error:'
  puts 'Error message:'
  puts e.message
end

```

- Untuk detail API, lihat [SynthesizeSpeech](#) di Referensi AWS SDK for Ruby API.

Rust

SDK for Rust

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

async fn synthesize(client: &Client, filename: &str) -> Result<(), Error> {
  let content = fs::read_to_string(filename);

  let resp = client
    .synthesize_speech()
    .output_format(OutputFormat::Mp3)
    .text(content.unwrap())
    .voice_id(VoiceId::Joanna)
    .send()
    .await?;
}

```

```

// Get MP3 data from response and save it
let mut blob = resp
    .audio_stream
    .collect()
    .await
    .expect("failed to read data");

let parts: Vec<&str> = filename.split('.').collect();
let out_file = format!("{}", String::from(parts[0]), ".mp3");

let mut file = tokio::fs::File::create(out_file)
    .await
    .expect("failed to create file");

file.write_all_buf(&mut blob)
    .await
    .expect("failed to write to file");

Ok(())
}

```

- Untuk detail API, lihat [SynthesizeSpeech](#) referensi AWS SDK for Rust API.

SAP ABAP

SDK for SAP ABAP

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

TRY.

```

" Only pass optional language code if it has a value
IF iv_lang_code IS NOT INITIAL.
    oo_result = lo_ply->synthesizespeech(
        iv_engine = iv_engine
        iv_outputformat = iv_output_fmt
        iv_text = iv_text

```

```

        iv_voiceid = iv_voice_id
        iv_languagecode = iv_lang_code ).
ELSE.
    oo_result = lo_ply->synthesizespeech(
        iv_engine = iv_engine
        iv_outputformat = iv_output_fmt
        iv_text = iv_text
        iv_voiceid = iv_voice_id ).
ENDIF.
MESSAGE 'Speech synthesized successfully.' TYPE 'I'.
CATCH /aws1/cx_plyinvalidssmlex.
    MESSAGE 'Invalid SSML.' TYPE 'E'.
CATCH /aws1/cx_plylexiconnotfoundex.
    MESSAGE 'Lexicon not found.' TYPE 'E'.
CATCH /aws1/cx_plyservicefailureex.
    MESSAGE 'Service failure occurred.' TYPE 'E'.
CATCH /aws1/cx_plytextlengthexcdex.
    MESSAGE 'Text length exceeded maximum.' TYPE 'E'.
ENDTRY.

```

- Untuk detail API, lihat [SynthesizeSpeech](#) di AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan Amazon Polly dengan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Skenario untuk Amazon Polly menggunakan AWS SDK

Contoh kode berikut menunjukkan cara menerapkan skenario umum di Amazon Polly dengan AWS SDK. Skenario ini menunjukkan kepada Anda cara menyelesaikan tugas tertentu dengan memanggil beberapa fungsi di Amazon Polly atau digabungkan dengan yang lain. Layanan AWS Setiap skenario menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode.

Skenario menargetkan tingkat pengalaman menengah untuk membantu Anda memahami tindakan layanan dalam konteks.

Contoh

- [Mengkonversi teks menjadi ucapan dan kembali ke teks menggunakan AWS SDK](#)

- [Buat aplikasi lip-sync dengan Amazon Polly menggunakan AWS SDK](#)
- [Buat aplikasi yang menganalisis umpan balik pelanggan dan mensintesis audio](#)
- [Memulai dengan sintesis text-to-speech](#)

Mengkonversi teks menjadi ucapan dan kembali ke teks menggunakan AWS SDK

Contoh kode berikut ini menunjukkan cara untuk melakukan:

- Gunakan Amazon Polly untuk mensintesis file input teks biasa (UTF-8) ke file audio.
- Mengunggah file audio ke bucket Amazon S3.
- Menggunakan Amazon Transcribe untuk mengonversi file audio menjadi teks.
- Tampilkan teks.

Rust

SDK for Rust

Gunakan Amazon Polly untuk mensintesis file input teks biasa (UTF-8) ke file audio, mengunggah file audio ke bucket Amazon S3, menggunakan Amazon Transcribe untuk mengonversi file audio tersebut menjadi teks, dan menampilkan teks.

Untuk kode sumber lengkap dan instruksi tentang cara mengatur dan menjalankan, lihat contoh lengkapnya di [GitHub](#).

Layanan yang digunakan dalam contoh ini

- Amazon Polly
- Amazon S3
- Amazon Transcribe

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan Amazon Polly dengan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Buat aplikasi lip-sync dengan Amazon Polly menggunakan AWS SDK

Contoh kode berikut menunjukkan cara membuat aplikasi lip-sync dengan Amazon Polly.

Python

SDK untuk Python (Boto3)

Menunjukkan cara menggunakan Amazon Polly dan Tkinter untuk membuat aplikasi sinkronisasi bibir yang menampilkan wajah animasi berbicara bersama dengan pidato yang disintesis oleh Amazon Polly. Lip-sync dicapai dengan meminta daftar viseme dari Amazon Polly yang cocok dengan pidato yang disintesis.

- Dapatkan metadata suara dari Amazon Polly dan tampilkan di aplikasi Tkinter.
- Dapatkan audio ucapan yang disintesis dan tanda ucapan viseme yang cocok dari Amazon Polly.
- Putar audio dengan gerakan mulut yang disinkronkan dalam wajah animasi.
- Kirim tugas sintesis asinkron untuk teks panjang dan ambil output dari bucket Amazon Simple Storage Service (Amazon S3).

Untuk kode sumber lengkap dan instruksi tentang cara mengatur dan menjalankan, lihat contoh lengkapnya di [GitHub](#).

Layanan yang digunakan dalam contoh ini

- Amazon Polly

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan Amazon Polly dengan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Buat aplikasi yang menganalisis umpan balik pelanggan dan mensintesis audio

Contoh kode berikut menunjukkan cara membuat aplikasi yang menganalisis kartu komentar pelanggan, menerjemahkannya dari bahasa aslinya, menentukan sentimen mereka, dan menghasilkan file audio dari teks yang diterjemahkan.

.NET

SDK untuk .NET

Aplikasi contoh ini menganalisis dan menyimpan kartu umpan balik pelanggan. Secara khusus, ini memenuhi kebutuhan hotel fiktif di New York City. Hotel menerima umpan balik dari para tamu dalam berbagai bahasa dalam bentuk kartu komentar fisik. Umpan balik itu diunggah ke aplikasi melalui klien web. Setelah gambar kartu komentar diunggah, langkah-langkah berikut terjadi:

- Teks diekstraksi dari gambar menggunakan Amazon Textract.
- Amazon Comprehend menentukan sentimen teks yang diekstraksi dan bahasanya.
- Teks yang diekstraksi diterjemahkan ke bahasa Inggris menggunakan Amazon Translate.
- Amazon Polly mensintesis file audio dari teks yang diekstraksi.

Aplikasi lengkap dapat digunakan dengan AWS CDK Untuk kode sumber dan petunjuk penerapan, lihat proyek di [GitHub](#).

Layanan yang digunakan dalam contoh ini

- Amazon Comprehend
- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

Java

SDK untuk Java 2.x

Aplikasi contoh ini menganalisis dan menyimpan kartu umpan balik pelanggan. Secara khusus, ini memenuhi kebutuhan hotel fiktif di New York City. Hotel menerima umpan balik dari para tamu dalam berbagai bahasa dalam bentuk kartu komentar fisik. Umpan balik itu diunggah ke aplikasi melalui klien web. Setelah gambar kartu komentar diunggah, langkah-langkah berikut terjadi:

- Teks diekstraksi dari gambar menggunakan Amazon Textract.
- Amazon Comprehend menentukan sentimen teks yang diekstraksi dan bahasanya.
- Teks yang diekstraksi diterjemahkan ke bahasa Inggris menggunakan Amazon Translate.

- Amazon Polly mensintesis file audio dari teks yang diekstraksi.

Aplikasi lengkap dapat digunakan dengan. AWS CDK Untuk kode sumber dan petunjuk penerapan, lihat proyek di [GitHub](#).

Layanan yang digunakan dalam contoh ini

- Amazon Comprehend
- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

JavaScript

SDK untuk JavaScript (v3)

Aplikasi contoh ini menganalisis dan menyimpan kartu umpan balik pelanggan. Secara khusus, ini memenuhi kebutuhan hotel fiktif di New York City. Hotel menerima umpan balik dari para tamu dalam berbagai bahasa dalam bentuk kartu komentar fisik. Umpan balik itu diunggah ke aplikasi melalui klien web. Setelah gambar kartu komentar diunggah, langkah-langkah berikut terjadi:

- Teks diekstraksi dari gambar menggunakan Amazon Textract.
- Amazon Comprehend menentukan sentimen teks yang diekstraksi dan bahasanya.
- Teks yang diekstraksi diterjemahkan ke bahasa Inggris menggunakan Amazon Translate.
- Amazon Polly mensintesis file audio dari teks yang diekstraksi.

Aplikasi lengkap dapat digunakan dengan. AWS CDK Untuk kode sumber dan petunjuk penerapan, lihat proyek di [GitHub](#). Kutipan berikut menunjukkan bagaimana yang AWS SDK untuk JavaScript digunakan di dalam fungsi Lambda.

```
import {
    ComprehendClient,
    DetectDominantLanguageCommand,
    DetectSentimentCommand,
} from "@aws-sdk/client-comprehend";

/**
```

```

* Determine the language and sentiment of the extracted text.
*
* @param {{ source_text: string }} extractTextOutput
*/
export const handler = async (extractTextOutput) => {
  const comprehendClient = new ComprehendClient({});

  const detectDominantLanguageCommand = new DetectDominantLanguageCommand({
    Text: extractTextOutput.source_text,
  });

  // The source language is required for sentiment analysis and
  // translation in the next step.
  const { Languages } = await comprehendClient.send(
    detectDominantLanguageCommand,
  );

  const languageCode = Languages[0].LanguageCode;

  const detectSentimentCommand = new DetectSentimentCommand({
    Text: extractTextOutput.source_text,
    LanguageCode: languageCode,
  });

  const { Sentiment } = await comprehendClient.send(detectSentimentCommand);

  return {
    sentiment: Sentiment,
    language_code: languageCode,
  };
};

```

```

import {
  DetectDocumentTextCommand,
  TextractClient,
} from "@aws-sdk/client-textract";

/**
* Fetch the S3 object from the event and analyze it using Amazon Textract.
*
* @param {import("@types/aws-lambda").EventBridgeEvent<"Object Created">}
eventBridgeS3Event
*/

```

```

export const handler = async (eventBridgeS3Event) => {
  const textractClient = new TextractClient();

  const detectDocumentTextCommand = new DetectDocumentTextCommand({
    Document: {
      S3Object: {
        Bucket: eventBridgeS3Event.bucket,
        Name: eventBridgeS3Event.object,
      },
    },
  });

  // Textract returns a list of blocks. A block can be a line, a page, word, etc.
  // Each block also contains geometry of the detected text.
  // For more information on the Block type, see https://docs.aws.amazon.com/textract/latest/dg/API\_Block.html.
  const { Blocks } = await textractClient.send(detectDocumentTextCommand);

  // For the purpose of this example, we are only interested in words.
  const extractedWords = Blocks.filter((b) => b.BlockType === "WORD").map(
    (b) => b.Text,
  );

  return extractedWords.join(" ");
};

```

```

import { PollyClient, SynthesizeSpeechCommand } from "@aws-sdk/client-polly";
import { S3Client } from "@aws-sdk/client-s3";
import { Upload } from "@aws-sdk/lib-storage";

/**
 * Synthesize an audio file from text.
 *
 * @param {{ bucket: string, translated_text: string, object: string }}
 * sourceDestinationConfig
 */
export const handler = async (sourceDestinationConfig) => {
  const pollyClient = new PollyClient({});

  const synthesizeSpeechCommand = new SynthesizeSpeechCommand({
    Engine: "neural",
    Text: sourceDestinationConfig.translated_text,
    VoiceId: "Ruth",
  });

```

```

    OutputFormat: "mp3",
  });

  const { AudioStream } = await pollyClient.send(synthesizeSpeechCommand);

  const audioKey = `${sourceDestinationConfig.object}.mp3`;

  // Store the audio file in S3.
  const s3Client = new S3Client();
  const upload = new Upload({
    client: s3Client,
    params: {
      Bucket: sourceDestinationConfig.bucket,
      Key: audioKey,
      Body: AudioStream,
      ContentType: "audio/mp3",
    },
  });

  await upload.done();
  return audioKey;
};

```

```

import {
  TranslateClient,
  TranslateTextCommand,
} from "@aws-sdk/client-translate";

/**
 * Translate the extracted text to English.
 *
 * @param {{ extracted_text: string, source_language_code: string }}
 * textAndSourceLanguage
 */
export const handler = async (textAndSourceLanguage) => {
  const translateClient = new TranslateClient({});

  const translateCommand = new TranslateTextCommand({
    SourceLanguageCode: textAndSourceLanguage.source_language_code,
    TargetLanguageCode: "en",
    Text: textAndSourceLanguage.extracted_text,
  });
};

```

```
const { TranslatedText } = await translateClient.send(translateCommand);

return { translated_text: TranslatedText };
};
```

Layanan yang digunakan dalam contoh ini

- Amazon Comprehend
- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

Ruby

SDK untuk Ruby

Aplikasi contoh ini menganalisis dan menyimpan kartu umpan balik pelanggan. Secara khusus, ini memenuhi kebutuhan hotel fiktif di New York City. Hotel menerima umpan balik dari para tamu dalam berbagai bahasa dalam bentuk kartu komentar fisik. Umpan balik itu diunggah ke aplikasi melalui klien web. Setelah gambar kartu komentar diunggah, langkah-langkah berikut terjadi:

- Teks diekstraksi dari gambar menggunakan Amazon Textract.
- Amazon Comprehend menentukan sentimen teks yang diekstraksi dan bahasanya.
- Teks yang diekstraksi diterjemahkan ke bahasa Inggris menggunakan Amazon Translate.
- Amazon Polly mensintesis file audio dari teks yang diekstraksi.

Aplikasi lengkap dapat digunakan dengan AWS CDK Untuk kode sumber dan petunjuk penerapan, lihat proyek di [GitHub](#).

Layanan yang digunakan dalam contoh ini

- Amazon Comprehend
- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan Amazon Polly dengan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Memulai dengan sintesis text-to-speech

Contoh kode berikut ini menunjukkan cara untuk melakukan:

- Pembersihan sumber daya

Bash

AWS CLI dengan skrip Bash

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankan di repositori [tutorial pengembang Sample](#).

```
#!/bin/bash

# Amazon Polly Getting Started Script
# This script demonstrates how to use Amazon Polly with the AWS CLI

set -euo pipefail

# Set up logging
LOG_FILE="polly-tutorial.log"
SCRIPT_DIR="$(cd "$(dirname "${BASH_SOURCE[0]}")" && pwd)"
WORK_DIR=$(mktemp -d)
trap 'cleanup_temp' EXIT

cleanup_temp() {
    rm -rf "$WORK_DIR"
}

echo "Starting Amazon Polly tutorial at $(date)" > "$LOG_FILE"

# Function to log commands and their output
log_cmd() {
```

```

    echo "Running: $1" | tee -a "$LOG_FILE"
    # Use bash array to safely handle arguments
    bash -c "$1" 2>&1 | tee -a "$LOG_FILE" || return $?
}

# Function to check for errors
check_error() {
    if echo "$1" | grep -iq "error"; then
        echo "ERROR detected in output. Exiting script." | tee -a "$LOG_FILE"
        echo "$1" | tee -a "$LOG_FILE"
        return 1
    fi
    return 0
}

# Function to handle errors and cleanup
handle_error() {
    local line_number=$1
    echo "Error occurred at line $line_number. Attempting cleanup..." | tee -a
"$LOG_FILE"
    cleanup
    exit 1
}

# Function to clean up resources
cleanup() {
    echo "" | tee -a "$LOG_FILE"
    echo "===== " | tee -a
"$LOG_FILE"
    echo "CLEANUP PROCESS" | tee -a "$LOG_FILE"
    echo "===== " | tee -a
"$LOG_FILE"

    # Delete lexicon if it exists
    if [[ -n "${LEXICON_NAME:-}" ]]; then
        echo "Deleting lexicon: $LEXICON_NAME" | tee -a "$LOG_FILE"
        if aws polly delete-lexicon --name "$LEXICON_NAME" 2>&1 | tee -a
"$LOG_FILE"; then
            echo "Lexicon deleted successfully." | tee -a "$LOG_FILE"
        else
            echo "Warning: Failed to delete lexicon." | tee -a "$LOG_FILE"
        fi
    fi
}

```

```
# Remove audio files
for file in output.mp3 ssm1-output.mp3 lexicon-output.mp3 example.pls; do
    if [[ -f "$file" ]]; then
        rm -f "$file"
        echo "Removed $file" | tee -a "$LOG_FILE"
    fi
done

echo "Cleanup complete." | tee -a "$LOG_FILE"
}

# Trap errors with line number
trap 'handle_error ${LINENO}' ERR

# Verify AWS CLI is available
if ! command -v aws &> /dev/null; then
    echo "AWS CLI is not installed. Please install it first." | tee -a
"$LOG_FILE"
    exit 1
fi

# Verify AWS credentials are configured
if ! aws sts get-caller-identity &> /dev/null; then
    echo "AWS credentials are not configured. Please configure them first." | tee
-a "$LOG_FILE"
    exit 1
fi

# Step 1: Verify Amazon Polly is available
echo "Step 1: Verifying Amazon Polly availability" | tee -a "$LOG_FILE"
if aws polly describe-voices --query 'Voices[0].Name' --output text &> /dev/null;
then
    echo "Amazon Polly is available. Proceeding with tutorial." | tee -a
"$LOG_FILE"
else
    echo "Amazon Polly is not available in your AWS CLI installation or region."
| tee -a "$LOG_FILE"
    echo "Please update your AWS CLI to the latest version or check your region."
| tee -a "$LOG_FILE"
    exit 1
fi

# Step 2: List available voices
echo "" | tee -a "$LOG_FILE"
```

```
echo "Step 2: Listing available voices" | tee -a "$LOG_FILE"
log_cmd "aws polly describe-voices --language-code en-US --output text --query
'Voices[0:3].[Id, LanguageCode, Gender]'" || true

# Step 3: Basic text-to-speech conversion
echo "" | tee -a "$LOG_FILE"
echo "Step 3: Converting text to speech" | tee -a "$LOG_FILE"
OUTPUT_FILE="${WORK_DIR}/output.mp3"
POLLY_TEXT="Hello, welcome to Amazon Polly. This is a sample text to speech
conversion."
log_cmd "aws polly synthesize-speech --output-format mp3 --voice-id Joanna --text
'$POLLY_TEXT' '$OUTPUT_FILE'" || true

if [[ -f "$OUTPUT_FILE" ]]; then
    echo "Successfully created output.mp3 file." | tee -a "$LOG_FILE"
    echo "You can play this file with your preferred audio player." | tee -a
"$LOG_FILE"
    cp "$OUTPUT_FILE" output.mp3
else
    echo "Failed to create output.mp3 file." | tee -a "$LOG_FILE"
    exit 1
fi

# Step 4: Using SSML for enhanced speech
echo "" | tee -a "$LOG_FILE"
echo "Step 4: Using SSML for enhanced speech" | tee -a "$LOG_FILE"
SSML_OUTPUT="${WORK_DIR}/ssml-output.mp3"
SSML_TEXT='<speack>Hello! <break time="1s"/> This is a sample of <emphasis>SSML
enhanced speech</emphasis>.</speack>'
log_cmd "aws polly synthesize-speech --output-format mp3 --voice-id Matthew --
text-type ssml --text '$SSML_TEXT' '$SSML_OUTPUT'" || true

if [[ -f "$SSML_OUTPUT" ]]; then
    echo "Successfully created ssml-output.mp3 file." | tee -a "$LOG_FILE"
    echo "You can play this file with your preferred audio player." | tee -a
"$LOG_FILE"
    cp "$SSML_OUTPUT" ssml-output.mp3
else
    echo "Failed to create ssml-output.mp3 file." | tee -a "$LOG_FILE"
    exit 1
fi

# Step 5: Working with lexicons
echo "" | tee -a "$LOG_FILE"
```

```

echo "Step 5: Working with lexicons" | tee -a "$LOG_FILE"

# Generate a random identifier for the lexicon (max 20 chars, alphanumeric only)
LEXICON_NAME="example$(openssl rand -hex 6 | cut -c 1-10)"
echo "Using lexicon name: $LEXICON_NAME" | tee -a "$LOG_FILE"

# Create a lexicon file
echo "Creating lexicon file..." | tee -a "$LOG_FILE"
LEXICON_FILE="${WORK_DIR}/example.pls"
cat > "$LEXICON_FILE" << 'EOF'
<?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>AWS</grapheme>
    <alias>Amazon Web Services</alias>
  </lexeme>
</lexicon>
EOF

# Upload the lexicon
echo "Uploading lexicon..." | tee -a "$LOG_FILE"
LEXICON_ARN=$(aws polly put-lexicon --name "$LEXICON_NAME" --content
  file://" "$LEXICON_FILE" --query 'LexiconArn' --output text 2>&1) || true
if [[ -n "$LEXICON_ARN" && "$LEXICON_ARN" != "" && ! "$LEXICON_ARN" =~ error ]];
then
  echo "Lexicon uploaded with ARN: $LEXICON_ARN" | tee -a "$LOG_FILE"
  aws polly tag-resource --resource-arn "$LEXICON_ARN" --tags
    Key=project,Value=doc-smith Key=tutorial,Value=amazon-polly-gs 2>&1 | tee -a
    "$LOG_FILE" || true
else
  echo "Lexicon uploaded." | tee -a "$LOG_FILE"
fi

# List available lexicons
echo "Listing available lexicons..." | tee -a "$LOG_FILE"
log_cmd "aws polly list-lexicons --output text --query 'Lexicons[*].[Name]'" ||
  true

```

```

# Get details about the lexicon
echo "Getting details about the lexicon..." | tee -a "$LOG_FILE"
log_cmd "aws polly get-lexicon --name '$LEXICON_NAME' --output text --query
'Lexicon.Name'" || true

# Use the lexicon when synthesizing speech
echo "Using the lexicon for speech synthesis..." | tee -a "$LOG_FILE"
LEXICON_OUTPUT="${WORK_DIR}/lexicon-output.mp3"
LEXICON_TEXT="I work with AWS every day."
log_cmd "aws polly synthesize-speech --output-format mp3 --voice-id Joanna --
lexicon-names '$LEXICON_NAME' --text '$LEXICON_TEXT' '$LEXICON_OUTPUT'" || true

if [[ -f "$LEXICON_OUTPUT" ]]; then
    echo "Successfully created lexicon-output.mp3 file." | tee -a "$LOG_FILE"
    echo "You can play this file with your preferred audio player." | tee -a
"$LOG_FILE"
    cp "$LEXICON_OUTPUT" lexicon-output.mp3
else
    echo "Failed to create lexicon-output.mp3 file." | tee -a "$LOG_FILE"
    exit 1
fi

# Summary of created resources
echo "" | tee -a "$LOG_FILE"
echo "===== " | tee -a
"$LOG_FILE"
echo "TUTORIAL SUMMARY" | tee -a "$LOG_FILE"
echo "===== " | tee -a
"$LOG_FILE"
echo "Created resources:" | tee -a "$LOG_FILE"
echo "1. Lexicon: $LEXICON_NAME" | tee -a "$LOG_FILE"
echo "2. Audio files:" | tee -a "$LOG_FILE"
echo "    - output.mp3" | tee -a "$LOG_FILE"
echo "    - ssm1-output.mp3" | tee -a "$LOG_FILE"
echo "    - lexicon-output.mp3" | tee -a "$LOG_FILE"
echo "" | tee -a "$LOG_FILE"

# Cleanup with auto-confirmation
echo "" | tee -a "$LOG_FILE"
echo "===== " | tee -a
"$LOG_FILE"
echo "CLEANUP CONFIRMATION" | tee -a "$LOG_FILE"
echo "===== " | tee -a
"$LOG_FILE"

```

```
echo "Cleaning up all created resources..." | tee -a "$LOG_FILE"
cleanup

echo "" | tee -a "$LOG_FILE"
echo "Tutorial completed successfully!" | tee -a "$LOG_FILE"
echo "Log file: $LOG_FILE" | tee -a "$LOG_FILE"
```

- Untuk detail API, lihat topik berikut di Referensi Perintah AWS CLI .
 - [DeleteLexicon](#)
 - [DescribeVoices](#)
 - [GetLexicon](#)
 - [Bantuan](#)
 - [ListLexicons](#)
 - [PutLexicon](#)
 - [SynthesizeSpeech](#)

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan Amazon Polly dengan SDK AWS](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Keamanan di Amazon Polly

Keamanan cloud di AWS adalah prioritas tertinggi. Sebagai AWS pelanggan, Anda mendapat manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan.

Keamanan adalah tanggung jawab bersama antara Anda AWS dan Anda. [Model tanggung jawab bersama](#) menjelaskan hal ini sebagai keamanan dari cloud dan keamanan dalam cloud:

- Keamanan cloud — AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan AWS layanan di AWS Cloud. AWS juga memberi Anda layanan yang dapat Anda gunakan dengan aman. Third-party auditor secara teratur menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari Program Kepatuhan Program [AWS Kepatuhan Program AWS](#) . Untuk mempelajari tentang program kepatuhan yang berlaku untuk Amazon Polly, lihat [AWS Layanan dalam Lingkup berdasarkan AWS Layanan Program Kepatuhan dalam Lingkup oleh Program](#) .
- Keamanan di cloud — Tanggung jawab Anda ditentukan oleh AWS layanan yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain termasuk sensitivitas data Anda, persyaratan perusahaan Anda, serta hukum dan peraturan yang berlaku.

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan Amazon Polly. Topik berikut menunjukkan cara mengonfigurasi Amazon Polly untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga mempelajari cara menggunakan AWS layanan lain yang membantu Anda memantau dan mengamankan sumber daya Amazon Polly Anda.

Topik

- [Perlindungan Data di Amazon Polly](#)
- [Identity and Access Management di Amazon Polly](#)
- [Pencatatan dan Pemantauan di Amazon Polly](#)
- [Validasi Kepatuhan untuk Amazon Polly](#)
- [Ketahanan di Amazon Polly](#)
- [Keamanan Infrastruktur di Amazon Polly](#)
- [Praktik Terbaik Keamanan untuk Amazon Polly](#)
- [Menggunakan Amazon Polly dengan titik akhir VPC antarmuka](#)

Perlindungan Data di Amazon Polly

Amazon Polly sesuai dengan [model tanggung jawab AWS bersama](#), yang mencakup peraturan dan pedoman untuk perlindungan data. AWS bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS layanan. AWS Mempertahankan kontrol atas data yang dihosting di infrastruktur ini, termasuk kontrol konfigurasi keamanan untuk menangani konten pelanggan dan data pribadi. AWS pelanggan dan mitra APN, yang bertindak sebagai pengontrol data atau pengolah data, bertanggung jawab atas data pribadi apa pun yang mereka masukkan ke Cloud. AWS

Untuk tujuan perlindungan data, kami menyarankan Anda melindungi kredensi AWS akun dan mengatur pengguna individu dengan AWS Identity and Access Management (IAM), sehingga setiap pengguna hanya diberikan izin yang diperlukan untuk memenuhi tugas pekerjaan mereka. Kami juga menyarankan supaya Anda mengamankan data dengan cara-cara berikut:

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk berkomunikasi dengan AWS sumber daya.
- Siapkan API dan pencatatan aktivitas pengguna dengan AWS CloudTrail.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default dalam AWS layanan.

Sebaiknya jangan pernah memasukkan informasi identitas yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam bidang isian bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan Amazon Polly atau AWS layanan lain menggunakan konsol, API AWS CLI, atau AWS SDK. Data apa pun yang Anda masukkan ke Amazon Polly atau layanan lain mungkin diambil untuk dimasukkan dalam log diagnostik. Saat Anda memberikan URL ke server eksternal, jangan menyertakan informasi kredensial di URL untuk memvalidasi permintaan Anda ke server tersebut.

Untuk informasi selengkapnya tentang perlindungan data, lihat postingan blog [Model Tanggung Jawab Bersama AWS dan GDPR](#) di Blog Keamanan AWS .

Enkripsi saat Data Tidak Berpindah

Output dari sintesis suara Amazon Polly Anda dapat disimpan di sistem Anda sendiri. Anda juga dapat menelepon Amazon Polly, lalu mengenkripsi file dengan kunci enkripsi pilihan Anda dan menyimpannya di Amazon Simple Storage Service (Amazon S3) atau penyimpanan aman lainnya. [the section called “SynthesizeSpeech”](#) Operasi Amazon Polly tidak memiliki kewarganegaraan dan tidak terkait dengan identitas pelanggan. Anda tidak dapat mengambilnya dari Amazon Polly nanti.

Enkripsi Saat Data Berpindah

Semua pengiriman teks dilindungi oleh TLS saat transit. Amazon Polly tidak menyimpan konten kiriman teks.

Privasi Lalu Lintas Kerja Internet

Akses ke Amazon Polly adalah melalui AWS konsol, CLI, atau SDK. Komunikasi menggunakan enkripsi sesi Transport Layer Security (TLS) untuk kerahasiaan dan [tanda tangan digital](#) untuk otentikasi dan integritas.

Identity and Access Management di Amazon Polly

AWS Identity and Access Management (IAM) adalah Layanan AWS yang membantu administrator mengontrol akses ke AWS sumber daya dengan aman. Administrator IAM mengontrol siapa yang dapat diautentikasi (masuk) dan diberi wewenang (memiliki izin) untuk menggunakan sumber daya Amazon Polly. IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Topik

- [Audiens](#)
- [Mengautentikasi dengan identitas](#)
- [Mengelola akses menggunakan kebijakan](#)
- [Bagaimana Amazon Polly bekerja dengan IAM](#)
- [Identity-based contoh kebijakan untuk Amazon Polly](#)
- [Izin API Amazon Polly: Tindakan, Izin, dan Referensi Sumber Daya](#)
- [Memecahkan masalah identitas dan akses Amazon Polly](#)

Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda berdasarkan peran Anda:

- Pengguna layanan - minta izin dari administrator Anda jika Anda tidak dapat mengakses fitur (lihat [Memecahkan masalah identitas dan akses Amazon Polly](#))
- Administrator layanan - tentukan akses pengguna dan mengirimkan permintaan izin (lihat [Bagaimana Amazon Polly bekerja dengan IAM](#))

- Administrator IAM - tulis kebijakan untuk mengelola akses (lihat [Identity-based contoh kebijakan untuk Amazon Polly](#))

Mengautentikasi dengan identitas

Otentikasi adalah cara Anda masuk AWS menggunakan kredensi identitas Anda. Anda harus diautentikasi sebagai Pengguna root akun AWS, pengguna IAM, atau dengan mengasumsikan peran IAM.

Anda dapat masuk sebagai identitas federasi menggunakan kredensial dari sumber identitas seperti AWS IAM Identity Center (Pusat Identitas IAM), autentikasi masuk tunggal, atau kredensial. Google/Facebook Untuk informasi selengkapnya tentang cara masuk, lihat [Cara masuk ke Akun AWS Anda](#) dalam Panduan Pengguna AWS Sign-In .

Untuk akses terprogram, AWS sediakan SDK dan CLI untuk menandatangani permintaan secara kriptografis. Untuk informasi selengkapnya, lihat [AWS Signature Version 4 untuk permintaan API](#) dalam Panduan Pengguna IAM.

Akun AWS pengguna root

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang disebut pengguna Akun AWS root yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari. Untuk tugas yang memerlukan kredensial pengguna root, lihat [Tugas yang memerlukan kredensial pengguna root](#) dalam Panduan Pengguna IAM.

Identitas terfederasi

Sebagai praktik terbaik, mewajibkan pengguna manusia untuk menggunakan federasi dengan penyedia identitas untuk mengakses Layanan AWS menggunakan kredensial sementara.

Identitas federasi adalah pengguna dari direktori perusahaan Anda, penyedia identitas web, atau Directory Service yang mengakses Layanan AWS menggunakan kredensial dari sumber identitas. Identitas terfederasi mengambil peran yang memberikan kredensial sementara.

Untuk manajemen akses terpusat, kami menyarankan AWS IAM Identity Center. Untuk informasi selengkapnya, lihat [Apa itu Pusat Identitas IAM?](#) dalam Panduan Pengguna AWS IAM Identity Center .

Pengguna dan grup IAM

[Pengguna IAM](#) adalah identitas dengan izin khusus untuk satu orang atau aplikasi. Sebaiknya gunakan kredensial sementara alih-alih pengguna IAM dengan kredensial jangka panjang. Untuk informasi selengkapnya, lihat [Mewajibkan pengguna manusia untuk menggunakan federasi dengan penyedia identitas untuk mengakses AWS menggunakan kredensi sementara](#) di Panduan Pengguna IAM.

[Grup IAM](#) menentukan kumpulan pengguna IAM dan mempermudah pengelolaan izin untuk pengguna dalam jumlah besar. Untuk mempelajari selengkapnya, lihat [Kasus penggunaan untuk pengguna IAM](#) dalam Panduan Pengguna IAM.

Peran IAM

[Peran IAM](#) adalah identitas dengan izin khusus yang menyediakan kredensial sementara. Anda dapat mengambil peran dengan [beralih dari pengguna ke peran IAM \(konsol\)](#) atau dengan memanggil operasi AWS CLI atau AWS API. Untuk informasi selengkapnya, lihat [Metode untuk mengambil peran](#) dalam Panduan Pengguna IAM.

Peran IAM berguna untuk akses pengguna terfederasi, izin pengguna IAM sementara, akses lintas akun, akses lintas layanan, dan aplikasi yang berjalan di Amazon EC2. Untuk informasi selengkapnya, lihat [Akses sumber daya lintas akun di IAM](#) dalam Panduan Pengguna IAM.

Mengelola akses menggunakan kebijakan

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke AWS identitas atau sumber daya. Kebijakan menentukan izin saat dikaitkan dengan identitas atau sumber daya. AWS mengevaluasi kebijakan ini ketika kepala sekolah membuat permintaan. Sebagian besar kebijakan disimpan AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang dokumen kebijakan JSON, lihat [Gambaran umum kebijakan JSON](#) dalam Panduan Pengguna IAM.

Menggunakan kebijakan, administrator menentukan siapa yang memiliki akses ke apa dengan mendefinisikan principal mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Secara default, pengguna dan peran tidak memiliki izin. Administrator IAM membuat kebijakan IAM dan menambahkannya ke peran, yang kemudian dapat diambil oleh pengguna. Kebijakan IAM mendefinisikan izin terlepas dari metode yang Anda gunakan untuk melakukan operasinya.

Identity-based kebijakan

Identity-based kebijakan adalah dokumen kebijakan izin JSON yang Anda lampirkan ke identitas (pengguna, grup, atau peran). Kebijakan ini mengontrol tindakan apa yang bisa dilakukan oleh identitas tersebut, terhadap sumber daya yang mana, dan dalam kondisi apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat [Tentukan izin IAM kustom dengan kebijakan yang dikelola pelanggan](#) dalam Panduan Pengguna IAM.

Identity-based kebijakan dapat berupa kebijakan inline (disematkan langsung ke dalam satu identitas) atau kebijakan terkelola (kebijakan mandiri yang dilampirkan pada beberapa identitas). Untuk mempelajari cara memilih antara kebijakan terkelola dan kebijakan inline, lihat [Pilih antara kebijakan terkelola dan kebijakan inline](#) dalam Panduan Pengguna IAM.

Resource-based kebijakan

Resource-based kebijakan adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contohnya termasuk kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Anda harus [menentukan principal](#) dalam kebijakan berbasis sumber daya.

Resource-based kebijakan adalah kebijakan inline yang terletak di layanan tersebut. Anda tidak dapat menggunakan kebijakan AWS terkelola dari IAM dalam kebijakan berbasis sumber daya.

Jenis-jenis kebijakan lain

AWS mendukung jenis kebijakan tambahan yang dapat menetapkan izin maksimum yang diberikan oleh jenis kebijakan yang lebih umum:

- Batasan izin – Menetapkan izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas kepada entitas IAM. Untuk informasi selengkapnya, lihat [Batasan izin untuk entitas IAM](#) dalam Panduan Pengguna IAM.
- Kebijakan kontrol layanan (SCP) – Menentukan izin maksimum untuk organisasi atau unit organisasi di AWS Organizations. Untuk informasi selengkapnya, lihat [Kebijakan kontrol layanan](#) dalam Panduan Pengguna AWS Organizations .
- Kebijakan kontrol sumber daya (RCP) – Menetapkan izin maksimum yang tersedia untuk sumber daya di akun Anda. Untuk informasi selengkapnya, lihat [Kebijakan kontrol sumber daya \(RCP\)](#) dalam Panduan Pengguna AWS Organizations .

- Kebijakan sesi – Kebijakan lanjutan yang diteruskan sebagai parameter saat membuat sesi sementara untuk peran atau pengguna terfederasi. Untuk informasi selengkapnya, lihat [Kebijakan sesi](#) dalam Panduan Pengguna IAM.

Berbagai jenis kebijakan

Ketika beberapa jenis kebijakan berlaku pada suatu permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan apakah akan mengizinkan permintaan saat beberapa jenis kebijakan terlibat, lihat [Logika evaluasi kebijakan](#) di Panduan Pengguna IAM.

Bagaimana Amazon Polly bekerja dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses ke Amazon Polly, pelajari fitur IAM apa yang tersedia untuk digunakan dengan Amazon Polly.

Fitur IAM yang dapat Anda gunakan dengan Amazon Polly

Fitur IAM	Dukungan Amazon Polly
Identity-based kebijakan	Ya
Resource-based kebijakan	Tidak
Tindakan kebijakan	Ya
Sumber daya kebijakan	Ya
kunci-kunci persyaratan kebijakan (spesifik layanan)	Tidak
ACL	Tidak
ABAC (tanda dalam kebijakan)	Tidak
Kredensial sementara	Ya
Teruskan sesi akses (FAS) untuk Amazon Polly	Ya
Peran layanan	Tidak
Service-linked peran	Tidak

Untuk mendapatkan tampilan tingkat tinggi tentang cara kerja Amazon Polly dan layanan AWS lainnya dengan sebagian besar fitur IAM, [AWS lihat layanan yang bekerja dengan IAM di Panduan Pengguna IAM](#).

Identity-based kebijakan untuk Amazon Polly

Mendukung kebijakan berbasis identitas: Ya

Identity-based kebijakan adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke identitas, seperti pengguna IAM, grup pengguna, atau peran. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat [Tentukan izin IAM kustom dengan kebijakan terkelola pelanggan](#) dalam Panduan Pengguna IAM.

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan secara spesifik apakah tindakan dan sumber daya diizinkan atau ditolak, serta kondisi yang menjadi dasar dikabulkan atau ditolaknya tindakan tersebut. Untuk mempelajari semua elemen yang dapat Anda gunakan dalam kebijakan JSON, lihat [Referensi elemen kebijakan JSON IAM](#) dalam Panduan Pengguna IAM.

Identity-based contoh kebijakan untuk Amazon Polly

Untuk melihat contoh kebijakan berbasis identitas Amazon Polly, lihat. [Identity-based contoh kebijakan untuk Amazon Polly](#)

Resource-based kebijakan dalam Amazon Polly

Mendukung kebijakan berbasis sumber daya: Tidak

Resource-based kebijakan adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya tempat kebijakan dilampirkan, kebijakan menentukan tindakan apa yang dapat dilakukan oleh principal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus [menentukan principal](#) dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. Layanan AWS

Untuk mengaktifkan akses lintas akun, Anda dapat menentukan secara spesifik seluruh akun atau entitas IAM di akun lain sebagai principal dalam kebijakan berbasis sumber daya. Untuk informasi selengkapnya, lihat [Akses sumber daya lintas akun di IAM](#) dalam Panduan Pengguna IAM.

Tindakan kebijakan untuk Amazon Polly

Mendukung tindakan kebijakan: Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, di mana utama dapat melakukan tindakan pada sumber daya, dan dalam kondisi apa.

Elemen `Action` dari kebijakan JSON menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Sertakan tindakan dalam kebijakan untuk memberikan izin untuk melakukan operasi terkait.

Untuk melihat daftar tindakan Amazon Polly, lihat [Tindakan yang ditentukan oleh Amazon Polly](#) di Referensi Otorisasi Layanan.

Tindakan kebijakan di Amazon Polly menggunakan awalan berikut sebelum tindakan:

```
polly
```

Untuk menetapkan secara spesifik beberapa tindakan dalam satu pernyataan, pisahkan tindakan tersebut dengan koma.

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

Untuk melihat contoh kebijakan berbasis identitas Amazon Polly, lihat. [Identity-based contoh kebijakan untuk Amazon Polly](#)

Sumber daya kebijakan untuk Amazon Polly

Mendukung sumber daya kebijakan: Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, di mana utama dapat melakukan tindakan pada sumber daya, dan dalam kondisi apa.

Elemen kebijakan JSON `Resource` menentukan objek yang menjadi target penerapan tindakan. Praktik terbaiknya, tentukan sumber daya menggunakan [Amazon Resource Name \(ARN\)](#). Untuk tindakan yang tidak mendukung izin di tingkat sumber daya, gunakan wildcard (*) untuk menunjukkan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

```
"Resource": "*"

```

Untuk melihat daftar jenis sumber daya Amazon Polly dan ARNnya, lihat Sumber daya yang [ditentukan oleh Amazon Polly](#) di Referensi Otorisasi Layanan. Untuk mempelajari tindakan mana yang dapat Anda tentukan ARN dari setiap sumber daya, lihat [Tindakan yang ditentukan oleh Amazon Polly](#).

Untuk melihat contoh kebijakan berbasis identitas Amazon Polly, lihat. [Identity-based contoh kebijakan untuk Amazon Polly](#)

Kunci kondisi kebijakan untuk Amazon Polly

Mendukung kunci kondisi kebijakan khusus layanan: Tidak

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, principal dapat melakukan tindakan pada suatu sumber daya, dan dalam suatu syarat.

Elemen `Condition` menentukan ketika pernyataan dieksekusi berdasarkan kriteria yang ditetapkan. Anda dapat membuat ekspresi bersyarat yang menggunakan [operator kondisi](#), misalnya sama dengan atau kurang dari, untuk mencocokkan kondisi dalam kebijakan dengan nilai-nilai yang diminta. Untuk melihat semua kunci kondisi AWS global, lihat [kunci konteks kondisi AWS global](#) di Panduan Pengguna IAM.

Untuk melihat daftar kunci kondisi Amazon Polly, lihat Kunci kondisi [untuk Amazon Polly](#) di Referensi Otorisasi Layanan. Untuk mempelajari tindakan dan sumber daya yang dapat Anda gunakan kunci kondisi, lihat [Tindakan yang ditentukan oleh Amazon Polly](#).

Untuk melihat contoh kebijakan berbasis identitas Amazon Polly, lihat. [Identity-based contoh kebijakan untuk Amazon Polly](#)

ACL di Amazon Polly

Mendukung ACL: Tidak

Daftar kontrol akses (ACL) mengendalikan principal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACL serupa dengan kebijakan berbasis sumber daya, meskipun kebijakan tersebut tidak menggunakan format dokumen kebijakan JSON.

ABAC dengan Amazon Polly

Mendukung ABAC (tag dalam kebijakan): Tidak

Attribute-based Access Control (ABAC) adalah strategi otorisasi yang mendefinisikan izin berdasarkan atribut yang disebut tag. Anda dapat melampirkan tag ke entitas dan AWS sumber daya IAM, lalu merancang kebijakan ABAC untuk mengizinkan operasi saat tag prinsipal cocok dengan tag pada sumber daya.

Untuk mengendalikan akses berdasarkan tanda, berikan informasi tentang tanda di [elemen kondisi](#) dari kebijakan menggunakan kunci kondisi `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, atau `aws:TagKeys`.

Jika sebuah layanan mendukung ketiga kunci kondisi untuk setiap jenis sumber daya, nilainya adalah Ya untuk layanan tersebut. Jika suatu layanan mendukung ketiga kunci kondisi untuk hanya beberapa jenis sumber daya, nilainya adalah Parsial.

Untuk informasi selengkapnya tentang ABAC, lihat [Tentukan izin dengan otorisasi ABAC](#) dalam Panduan Pengguna IAM. Untuk melihat tutorial yang menguraikan langkah-langkah pengaturan ABAC, lihat [Menggunakan kontrol akses berbasis atribut \(ABAC\)](#) dalam Panduan Pengguna IAM.

Menggunakan kredensial sementara dengan Amazon Polly

Mendukung kredensial sementara: Ya

Kredensial sementara menyediakan akses jangka pendek ke AWS sumber daya dan secara otomatis dibuat saat Anda menggunakan federasi atau beralih peran. AWS merekomendasikan agar Anda menghasilkan kredensial sementara secara dinamis alih-alih menggunakan kunci akses jangka panjang. Untuk informasi selengkapnya, lihat [Kredensial keamanan sementara di IAM](#) dan [Layanan AWS yang berfungsi dengan IAM](#) dalam Panduan Pengguna IAM.

Cross-service sesi akses teruskan (FAS) untuk Amazon Polly

Mendukung sesi akses terusan (FAS): Ya

Sesi akses terusan (FAS) menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Untuk detail kebijakan ketika mengajukan permintaan FAS, lihat [Sesi akses terusan](#).

Peran layanan untuk Amazon Polly

Mendukung peran layanan: Tidak

Peran layanan adalah [peran IAM](#) yang diambil oleh sebuah layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat [Buat sebuah peran untuk mendelegasikan izin ke Layanan AWS](#) dalam Panduan pengguna IAM.

Warning

Mengubah izin untuk peran layanan dapat merusak fungsionalitas Amazon Polly. Edit peran layanan hanya jika Amazon Polly memberikan panduan untuk melakukannya.

Service-linked peran untuk Amazon Polly

Mendukung peran terkait layanan: Tidak

Peran terkait layanan adalah jenis peran layanan yang ditautkan ke. Layanan AWS Layanan dapat mengambil peran untuk melakukan tindakan atas nama Anda. Service-linked peran muncul di Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.

Untuk detail tentang pembuatan atau manajemen peran terkait layanan, lihat [Layanan AWS yang berfungsi dengan IAM](#). Temukan layanan dalam tabel yang Yes menyertakan kolom Service-linked peran. Pilih tautan Ya untuk melihat dokumentasi peran terkait layanan untuk layanan tersebut.

Peran IAM Amazon Polly

Anda dapat melampirkan kebijakan izin berbasis identitas ke IAM role untuk memberikan izin lintas akun. Misalnya, administrator di akun A dapat membuat peran untuk memberikan izin lintas akun ke AWS akun lain (misalnya, akun B) atau AWS layanan sebagai berikut:

1. Administrator akun A membuat IAM role dan melampirkan kebijakan izin untuk peran yang memberikan izin pada sumber daya di akun A.
2. Administrator akun A melampirkan kebijakan kepercayaan pada akun identifikasi peran B sebagai penanggung jawab yang dapat menjalankan peran tersebut.
3. Administrator Akun B kemudian dapat mendelegasikan izin untuk mengambil peran kepada setiap pengguna di akun B. Melakukan hal ini memungkinkan pengguna di akun B untuk membuat atau

mengakses sumber daya di akun A. Prinsip dalam kebijakan kepercayaan juga dapat menjadi prinsipal AWS layanan jika Anda ingin memberikan izin AWS layanan untuk mengambil peran.

Untuk informasi selengkapnya tentang penggunaan IAM untuk mendelegasikan izin, lihat [Manajemen Akses](#) dalam Panduan Pengguna IAM.

Berikut ini adalah contoh kebijakan yang memberikan izin untuk menempatkan dan mendapatkan leksikon serta mencantumkan leksikon yang saat ini tersedia.

Amazon Polly mendukung Identity-based kebijakan tindakan di tingkat sumber daya. Dalam beberapa kasus, sumber daya dapat dibatasi oleh ARN. Ini berlaku untuk `SynthesizeSpeech`, `StartSpeechSynthesisTask`, `PutLexiconGetLexicon`, dan `DeleteLexicon` operasi. Dalam kasus ini, `Resource` nilainya ditunjukkan oleh ARN. Misalnya: `arn:aws:polly:us-east-2:account-id:lexicon/*` karena `Resource` nilai menentukan izin pada semua leksikon yang dimiliki dalam Wilayah. `us-east-2`

Namun, tidak semua operasi menggunakan ARN. Ini adalah kasus dengan `DescribeVoices`, `ListLexiconsGetSpeechSynthesisTasks`, dan `ListSpeechSynthesisTasks` operasi.

Untuk informasi lebih lanjut tentang pengguna, grup, peran, dan izin, lihat [Identitas \(Pengguna, Grup, dan Peran\)](#) dalam Panduan Pengguna IAM.

Identity-based contoh kebijakan untuk Amazon Polly

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau memodifikasi sumber daya Amazon Polly. Untuk memberikan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM dengan menggunakan contoh dokumen kebijakan JSON ini, lihat [Membuat kebijakan IAM \(konsol\) di Panduan Pengguna IAM](#).

Untuk detail tentang tindakan dan jenis sumber daya yang ditentukan oleh Amazon Polly, termasuk format ARN untuk setiap jenis sumber daya, lihat [Tindakan, sumber daya, dan kunci kondisi untuk Amazon Polly](#) di Referensi Otorisasi Layanan.

Topik

- [Praktik terbaik kebijakan](#)
- [Menggunakan konsol Amazon Polly](#)
- [Mengizinkan pengguna melihat izin mereka sendiri](#)

- [AWS kebijakan terkelola \(standar\) untuk Amazon Polly](#)
- [Amazon Polly memperbarui ke AWS kebijakan terkelola](#)
- [Customer-managed contoh kebijakan](#)

Praktik terbaik kebijakan

Identity-based kebijakan menentukan apakah seseorang dapat membuat, mengakses, atau menghapus sumber daya Amazon Polly di akun Anda. Tindakan ini membuat Akun AWS Anda dikenai biaya. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan AWS terkelola dan beralih ke izin hak istimewa paling sedikit — Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan AWS terkelola yang memberikan izin untuk banyak kasus penggunaan umum. Mereka tersedia di Anda Akun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola AWS pelanggan yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat [Kebijakan yang dikelola AWS](#) atau [Kebijakan yang dikelola AWS untuk fungsi tugas](#) dalam Panduan Pengguna IAM.
- Menerapkan izin dengan hak akses paling rendah – Ketika Anda menetapkan izin dengan kebijakan IAM, hanya berikan izin yang diperlukan untuk melakukan tugas. Anda melakukannya dengan mendefinisikan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, yang juga dikenal sebagai izin dengan hak akses paling rendah. Untuk informasi selengkapnya tentang cara menggunakan IAM untuk mengajukan izin, lihat [Kebijakan dan izin dalam IAM](#) dalam Panduan Pengguna IAM.
- Gunakan kondisi dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan suatu kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Sebagai contoh, Anda dapat menulis kondisi kebijakan untuk menentukan bahwa semua permintaan harus dikirim menggunakan SSL. Anda juga dapat menggunakan ketentuan untuk memberikan akses ke tindakan layanan jika digunakan melalui yang spesifik Layanan AWS, seperti CloudFormation. Untuk informasi selengkapnya, lihat [Elemen kebijakan JSON IAM: Kondisi](#) dalam Panduan Pengguna IAM.
- Gunakan IAM Access Analyzer untuk memvalidasi kebijakan IAM Anda untuk memastikan izin yang aman dan fungsional – IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan tersebut mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM. IAM Access Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk

informasi selengkapnya, lihat [Validasi kebijakan dengan IAM Access Analyzer](#) dalam Panduan Pengguna IAM.

- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Anda, Akun AWS aktifkan MFA untuk keamanan tambahan. Untuk meminta MFA ketika operasi API dipanggil, tambahkan kondisi MFA pada kebijakan Anda. Untuk informasi selengkapnya, lihat [Amankan akses API dengan MFA](#) dalam Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, lihat [Praktik terbaik keamanan di IAM](#) dalam Panduan Pengguna IAM.

Menggunakan konsol Amazon Polly

Untuk mengakses konsol Amazon Polly, Anda harus memiliki set izin minimum. Izin ini harus memungkinkan Anda untuk membuat daftar dan melihat detail tentang sumber daya Amazon Polly di Anda. Akun AWS Jika Anda membuat kebijakan berbasis identitas yang lebih ketat daripada izin minimum yang diperlukan, konsol tidak akan berfungsi sebagaimana mestinya untuk entitas (pengguna atau peran) dengan kebijakan tersebut.

Anda tidak perlu mengizinkan izin konsol minimum untuk pengguna yang melakukan panggilan hanya ke AWS CLI atau AWS API. Sebagai gantinya, izinkan akses hanya ke tindakan yang sesuai dengan operasi API yang coba mereka lakukan.

Untuk memastikan bahwa pengguna dan peran masih dapat menggunakan konsol Amazon Polly, lampirkan juga Amazon *ConsoleAccess* Polly *ReadOnly* AWS atau kebijakan terkelola ke entitas. Untuk informasi selengkapnya, lihat [Menambah izin untuk pengguna](#) dalam Panduan Pengguna IAM.

Untuk menggunakan konsol Amazon Polly, berikan izin ke semua API Amazon Polly. Tidak ada izin tambahan yang diperlukan. Untuk mendapatkan fungsionalitas konsol lengkap, Anda dapat menggunakan kebijakan berikut:.

Mengizinkan pengguna melihat izin mereka sendiri

Contoh ini menunjukkan cara membuat kebijakan yang mengizinkan pengguna IAM melihat kebijakan inline dan terkelola yang dilampirkan ke identitas pengguna mereka. Kebijakan ini mencakup izin untuk menyelesaikan tindakan ini di konsol atau menggunakan API atau secara terprogram. AWS CLI AWS

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "ViewOwnUserInfo",
    "Effect": "Allow",
    "Action": [
      "iam:GetUserPolicy",
      "iam:ListGroupForUser",
      "iam:ListAttachedUserPolicies",
      "iam:ListUserPolicies",
      "iam:GetUser"
    ],
    "Resource": ["arn:aws:iam::*:user/${aws:username}"]
  },
  {
    "Sid": "NavigateInConsole",
    "Effect": "Allow",
    "Action": [
      "iam:GetGroupPolicy",
      "iam:GetPolicyVersion",
      "iam:GetPolicy",
      "iam:ListAttachedGroupPolicies",
      "iam:ListGroupPolicies",
      "iam:ListPolicyVersions",
      "iam:ListPolicies",
      "iam:ListUsers"
    ],
    "Resource": "*"
  }
]
```

AWS kebijakan terkelola (standar) untuk Amazon Polly

AWS mengatasi banyak kasus penggunaan umum dengan menyediakan kebijakan IAM mandiri yang dibuat dan dikelola oleh AWS. Kebijakan AWS terkelola ini memberikan izin yang diperlukan untuk kasus penggunaan umum sehingga Anda dapat menghindari keharusan menyelidiki izin apa yang diperlukan. Untuk informasi selengkapnya, lihat [Kebijakan Terkelola AWS](#) dalam Panduan Pengguna IAM.

Kebijakan AWS terkelola berikut, yang dapat Anda lampirkan ke pengguna di akun Anda, khusus untuk Amazon Polly:

- **AmazonPollyReadOnlyAccess**— Memberikan akses hanya-baca ke sumber daya, memungkinkan daftar leksikon, mengambil leksikon, mencantumkan suara yang tersedia, dan mensintesis ucapan (termasuk, menerapkan leksikon ke pidato yang disintesis).

Untuk melihat detail selengkapnya tentang kebijakan, termasuk versi terbaru dari dokumen kebijakan JSON, lihat [AmazonPollyReadOnlyAccess](#) di Panduan Referensi Kebijakan AWS Terkelola.

- **AmazonPollyFullAccess**— Memberikan akses penuh ke sumber daya dan semua operasi yang didukung.

Untuk melihat detail selengkapnya tentang kebijakan, termasuk versi terbaru dari dokumen kebijakan JSON, lihat [AmazonPollyFullAccess](#) di Panduan Referensi Kebijakan AWS Terkelola.

Note

Anda dapat meninjau kebijakan-kebijakan izin ini dengan masuk ke konsol IAM dan mencari kebijakan-kebijakan tertentu di sana.

Anda juga dapat membuat kebijakan IAM kustom Anda sendiri untuk mengizinkan izin untuk tindakan dan sumber daya Amazon Polly. Anda dapat melampirkan kebijakan-kebijakan kustom ini ke pengguna IAM atau grup yang memerlukan izin-izin tersebut.

Amazon Polly memperbarui ke AWS kebijakan terkelola

Lihat detail tentang pembaruan kebijakan AWS terkelola untuk Amazon Polly sejak layanan ini mulai melacak perubahan ini. Untuk peringatan otomatis tentang perubahan pada halaman ini, berlangganan umpan RSS di halaman Riwayat Dokumen Amazon Polly.

Amazon Polly memperbarui ke AWS kebijakan terkelola

Kebijakan	Ubah	Date
AmazonPollyReadOnlyAccess	Kebijakan terkelola yang diperbarui - Menambahkan izin <code>polly:StartSpeechSynthesisStream</code> untuk mendukung kemampuan sintesis ucapan streaming dua arah.	Maret 19, 2026

Customer-managed contoh kebijakan

Di bagian ini, Anda dapat menemukan contoh kebijakan pengguna yang memberikan izin untuk berbagai tindakan Amazon Polly. Kebijakan ini berfungsi saat Anda menggunakan AWS SDK atau AWS CLI Saat Anda menggunakan konsol, berikan izin ke semua API Amazon Polly.

Note

Semua contoh menggunakan Wilayah us-east-2 dan berisi ID akun fiktif.

Contoh

- [Contoh 1: Izinkan Semua Tindakan Polly Amazon](#)
- [Contoh 2: Izinkan semua tindakan Amazon Polly kecuali DeleteLexicon](#)
- [Contoh 3: Izinkan DeleteLexicon](#)
- [Contoh 4: Izinkan Hapus Leksikon di Wilayah tertentu](#)
- [Contoh 5: Izinkan DeleteLexicon Lexicon tertentu](#)

Contoh 1: Izinkan Semua Tindakan Polly Amazon

Setelah Anda mendaftar (lihat [Memulai dengan Amazon Polly](#)) buat pengguna administrator untuk mengelola akun Anda, termasuk membuat pengguna dan mengelola izin mereka.

Anda dapat membuat pengguna yang memiliki izin untuk semua tindakan Amazon Polly. Pikirkan pengguna ini sebagai administrator khusus layanan untuk bekerja dengan Amazon Polly. Anda dapat melampirkan kebijakan izin berikut ke pengguna ini.

JSON

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

```
]
}
```

Contoh 2: Izinkan semua tindakan Amazon Polly kecuali DeleteLexicon

Kebijakan izin berikut memberikan izin pengguna untuk melakukan semua tindakan kecuali `DeleteLexicon`, dengan izin untuk menghapus secara eksplisit ditolak di semua Wilayah.

Contoh 3: Izinkan DeleteLexicon

Kebijakan izin berikut memberikan izin pengguna untuk menghapus leksikon apa pun yang Anda miliki terlepas dari proyek atau Wilayah di mana ia berada.

JSON

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

Contoh 4: Izinkan Hapus Leksikon di Wilayah tertentu

Kebijakan izin berikut memberikan izin pengguna untuk menghapus leksikon apa pun dalam proyek apa pun yang Anda miliki yang terletak di satu Wilayah (dalam hal ini, us-east-2).

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "AllowDeleteSpecifiedRegion",
    "Effect": "Allow",
```

```
"Action": [
    "polly:DeleteLexicon"],
"Resource": "arn:aws:polly:us-east-2:111122223333:lexicon/*"
}
]
```

Contoh 5: Izinkan DeleteLexicon Lexicon tertentu

Kebijakan izin berikut memberikan izin pengguna untuk menghapus leksikon tertentu yang Anda miliki (dalam hal ini, MyLexicon) di Wilayah tertentu (dalam hal ini, us-east-2).

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "AllowDeleteForSpecifiedLexicon",
    "Effect": "Allow",
    "Action": [
      "polly:DeleteLexicon"],
    "Resource": "arn:aws:polly:us-east-2:111122223333:lexicon/myLexicon"
  ]
}
```

Izin API Amazon Polly: Tindakan, Izin, dan Referensi Sumber Daya

Saat menyiapkan kebijakan izin yang dapat dilampirkan ke identitas IAM (kebijakan berbasis identitas), Anda dapat menggunakan berikut sebagai referensi. Daftar mencakup setiap operasi Amazon Polly API, tindakan terkait yang dapat Anda berikan izin untuk melakukan tindakan, dan AWS sumber daya yang dapat Anda berikan izin. Anda menentukan tindakan dalam bidang `Action` kebijakan, dan Anda menentukan nilai sumber daya pada bidang `Resource` kebijakan.

Anda dapat menggunakan kunci kondisi AWS-wide dalam kebijakan Amazon Polly Anda untuk menyatakan kondisi. Untuk daftar lengkap tombol AWS-wide, lihat kunci yang [tersedia](#) di Panduan Pengguna IAM.

 Note

Untuk menentukan tindakan, gunakan awalan `polly` diikuti dengan nama operasi API (misalnya, `polly:GetLexicon`).

Amazon Polly mendukung Identity-based kebijakan tindakan di tingkat sumber daya. Oleh karena itu, Resource nilainya ditunjukkan oleh ARN. Misalnya: `arn:aws:polly:us-east-2:account-id:lexicon/*` karena Resource nilai menentukan izin pada semua leksikon yang dimiliki dalam Wilayah. `us-east-2`

Karena Amazon Polly tidak mendukung izin untuk tindakan di tingkat sumber daya, sebagian besar kebijakan menetapkan karakter wildcard (*) sebagai nilainya. Resource Namun, jika perlu membatasi izin ke Wilayah tertentu karakter wildcard ini diganti dengan ARN yang sesuai: `arn:aws:polly:region:account-id:lexicon/*`.

Amazon Polly API dan Izin yang Diperlukan untuk Tindakan

Operasi API: [DeleteLexicon](#)

Izin yang Diperlukan (Tindakan API): `polly:DeleteLexicon`

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

Operasi API: [DescribeVoices](#)

Izin yang Diperlukan (Tindakan API): `polly:DescribeVoices`

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

Operasi API: [GetLexicon](#)

Izin yang Diperlukan (Tindakan API): `polly:GetLexicon`

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

Operasi API: [ListLexicons](#)

Izin yang Diperlukan (Tindakan API): `polly:ListLexicons`

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

Operasi API: [PutLexicon](#)

Izin yang Diperlukan (Tindakan API): `polly:ListLexicons`

Sumber Daya: *

Operasi API: [SynthesizeSpeech](#)

Izin yang Diperlukan (Tindakan API): `polly:SynthesizeSpeech`

Sumber Daya: *

Memecahkan masalah identitas dan akses Amazon Polly

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui saat bekerja dengan Amazon Polly dan IAM.

Topik

- [Saya tidak berwenang untuk melakukan tindakan di Amazon Polly](#)
- [Saya tidak berwenang untuk melakukan iam: PassRole](#)
- [Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses sumber daya Amazon Polly saya](#)

Saya tidak berwenang untuk melakukan tindakan di Amazon Polly

Jika Anda menerima pesan kesalahan bahwa Anda tidak memiliki otorisasi untuk melakukan tindakan, kebijakan Anda harus diperbarui agar Anda dapat melakukan tindakan tersebut.

Contoh kesalahan berikut terjadi ketika pengguna IAM `mateojackson` mencoba menggunakan konsol untuk melihat detail tentang suatu sumber daya `my-example-widget` rekaan, tetapi tidak memiliki izin `polly:GetWidget` rekaan.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
polly:GetWidget on resource: my-example-widget
```

Dalam hal ini, kebijakan untuk pengguna `mateojackson` harus diperbarui untuk mengizinkan akses ke sumber daya `my-example-widget` dengan menggunakan tindakan `polly:GetWidget`.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya tidak berwenang untuk melakukan iam: PassRole

Jika Anda menerima kesalahan bahwa Anda tidak diizinkan untuk melakukan `iam:PassRole` tindakan, kebijakan Anda harus diperbarui agar Anda dapat meneruskan peran ke Amazon Polly.

Beberapa Layanan AWS memungkinkan Anda untuk meneruskan peran yang ada ke layanan tersebut alih-alih membuat peran layanan baru atau peran terkait layanan. Untuk melakukannya, Anda harus memiliki izin untuk meneruskan peran ke layanan.

Contoh kesalahan berikut terjadi ketika pengguna IAM bernama `marymajor` mencoba menggunakan konsol untuk melakukan tindakan di Amazon Polly. Namun, tindakan tersebut memerlukan layanan untuk mendapatkan izin yang diberikan oleh peran layanan. Mary tidak memiliki izin untuk meneruskan peran tersebut pada layanan.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

Dalam kasus ini, kebijakan Mary harus diperbarui agar dia mendapatkan izin untuk melakukan tindakan `iam:PassRole` tersebut.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses sumber daya Amazon Polly saya

Anda dapat membuat peran yang dapat digunakan pengguna di akun lain atau orang-orang di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa saja yang dipercaya untuk mengambil peran tersebut. Untuk layanan yang mendukung kebijakan berbasis sumber daya atau daftar kontrol akses (ACL), Anda dapat menggunakan kebijakan tersebut untuk memberi orang akses ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa referensi berikut:

- Untuk mengetahui apakah Amazon Polly mendukung fitur-fitur ini, lihat. [Bagaimana Amazon Polly bekerja dengan IAM](#)
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda di seluruh sumber daya Akun AWS yang Anda miliki, lihat [Menyediakan akses ke pengguna IAM di pengguna lain Akun AWS yang Anda miliki](#) di Panduan Pengguna IAM.

- Untuk mempelajari cara menyediakan akses ke sumber daya Anda kepada pihak ketiga Akun AWS, lihat [Menyediakan akses yang Akun AWS dimiliki oleh pihak ketiga](#) dalam Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses melalui federasi identitas, lihat [Menyediakan akses ke pengguna terautentikasi eksternal \(federasi identitas\)](#) dalam Panduan Pengguna IAM.
- Untuk mempelajari perbedaan antara menggunakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat [Akses sumber daya lintas akun di IAM di Panduan Pengguna IAM](#).

Pencatatan dan Pemantauan di Amazon Polly

Pemantauan adalah bagian penting dalam menjaga keandalan, ketersediaan, dan kinerja aplikasi Amazon Polly Anda. Untuk memantau panggilan Amazon Polly API, Anda dapat menggunakan AWS CloudTrail Untuk memantau status pekerjaan Anda, gunakan Amazon CloudWatch Logs.

- CloudWatch Alarm Amazon — Menggunakan CloudWatch alarm, Anda menonton satu metrik selama periode waktu yang Anda tentukan. Jika metrik melebihi ambang batas tertentu, pemberitahuan akan dikirim ke topik atau AWS Auto Scaling kebijakan Amazon Simple Notification Service. CloudWatch alarm tidak memanggil tindakan ketika metrik berada dalam keadaan tertentu. Sebaliknya, kondisi tersebut harus diubah dan dipertahankan selama periode tertentu. Untuk informasi selengkapnya, lihat [Integrasi CloudWatch dengan Amazon Polly](#).
- CloudTrail log - CloudTrail menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di Amazon Polly. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat ke Amazon Polly. Anda juga dapat menentukan alamat IP untuk membuat permintaan, siapa yang membuat permintaan, kapan permintaan dibuat, dan detail tambahan. Untuk informasi selengkapnya, lihat [Mencatat panggilan API Amazon Polly dengan AWS CloudTrail](#).

Validasi Kepatuhan untuk Amazon Polly

Third-party auditor menilai keamanan dan kepatuhan Amazon Polly sebagai bagian dari AWS beberapa program kepatuhan. Program ini mencakup SOC, PCI, FedRAMP, HIPAA, dan lainnya.

Untuk daftar AWS layanan dalam lingkup program kepatuhan tertentu, lihat [AWS Layanan dalam Lingkup oleh AWS Layanan Program Kepatuhan](#) . Untuk informasi umum, lihat [Program AWS Kepatuhan Program AWS](#) .

Anda dapat mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi selengkapnya, silakan lihat [Mengunduh Laporan di AWS Artifact](#).

Tanggung jawab kepatuhan Anda saat menggunakan Amazon Polly ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, serta undang-undang dan peraturan yang berlaku. AWS menyediakan sumber daya berikut untuk membantu kepatuhan:

- [Panduan Quick Start Keamanan dan Kepatuhan](#) – Panduan deployment ini membahas pertimbangan arsitektur dan menyediakan langkah-langkah untuk melakukan deployment terhadap lingkungan dasar di AWS yang menjadi fokus keamanan dan kepatuhan.
- [Arsitektur untuk HIPAA Security and Compliance Whitepaper](#) — Whitepaper ini menjelaskan bagaimana perusahaan dapat menggunakan untuk membuat aplikasi. AWS HIPAA-compliant
- [AWS Sumber Daya AWS](#) — Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- [Mengevaluasi Sumber Daya dengan Aturan](#) dalam Panduan AWS Config Pengembang — AWS Config Layanan menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- [AWS Security Hub CSPM](#)— AWS Layanan ini memberikan pandangan komprehensif tentang keadaan keamanan Anda di dalamnya AWS yang membantu Anda memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik.

Ketahanan di Amazon Polly

Infrastruktur AWS global dibangun di sekitar AWS Wilayah dan Zona Ketersediaan. AWS Wilayah menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi, yang terhubung dengan latensi rendah, throughput tinggi, dan jaringan yang sangat redundan. Dengan Zona Ketersediaan, Anda dapat merancang serta mengoperasikan aplikasi dan basis data yang secara otomatis melakukan fail over di antara zona tanpa gangguan. Zona Ketersediaan memiliki ketersediaan dan toleransi kesalahan yang lebih baik, dan dapat diskalakan dibandingkan infrastruktur pusat data tunggal atau multi tradisional.

Untuk informasi selengkapnya tentang AWS Wilayah dan Availability Zone, lihat [Infrastruktur AWS Global](#).

Keamanan Infrastruktur di Amazon Polly

Sebagai layanan terkelola, Amazon Polly dilindungi oleh prosedur keamanan jaringan AWS global yang dijelaskan dalam whitepaper [Amazon Web Services: Tinjauan Proses Keamanan](#).

Anda menggunakan panggilan API yang AWS dipublikasikan untuk mengakses Amazon Polly melalui jaringan. Klien harus mendukung Keamanan Lapisan Pengangkutan (TLS) 1.0 atau versi yang lebih baru. Kami merekomendasikan TLS 1.2 atau versi yang lebih baru. Klien juga harus mendukung cipher suite dengan perfect forward secrecy (PFS) seperti Ephemeral (DHE) atau Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Diffie-Hellman Sebagian besar sistem modern seperti Java 7 dan versi lebih baru mendukung mode-mode ini.

Selain itu, permintaan harus ditandatangani menggunakan ID kunci akses dan kunci akses rahasia yang terkait dengan principal IAM. Atau Anda dapat menggunakan [AWS Security Token Service](#) (AWS STS) untuk menghasilkan kredensial keamanan sementara untuk menandatangani permintaan.

Praktik Terbaik Keamanan untuk Amazon Polly

Kepercayaan, privasi, dan keamanan konten Anda adalah prioritas tertinggi kami. Kami menerapkan kontrol teknis dan fisik yang bertanggung jawab dan canggih yang dirancang untuk mencegah akses tidak sah ke, atau pengungkapan, konten Anda dan memastikan bahwa penggunaan kami sesuai dengan komitmen kami kepada Anda. Untuk informasi selengkapnya, lihat [FAQ Privasi AWS Data](#).

Amazon Polly tidak mempertahankan konten kiriman teks.

Untuk pandangan AWS keamanan yang luas, termasuk kepatuhan, pengujian penetrasi, buletin, dan sumber daya, kunjungi situs web [AWS Cloud Security](#).

Menggunakan Amazon Polly dengan titik akhir VPC antarmuka

Jika Anda menggunakan Amazon Virtual Private Cloud (Amazon VPC) untuk meng-host AWS sumber daya Anda, Anda dapat membuat koneksi pribadi antara VPC dan Amazon Polly. Anda dapat menggunakan koneksi ini untuk mensintesis ucapan dengan Amazon Polly tanpa melintasi internet publik.

Amazon VPC adalah AWS layanan yang dapat Anda gunakan untuk meluncurkan AWS sumber daya di jaringan virtual yang Anda tentukan. Dengan VPC, Anda memiliki kendali terhadap

pengaturan jaringan, seperti rentang alamat IP, subnet, tabel rute, dan pintu masuk jaringan. Untuk menghubungkan VPC Anda ke Amazon Polly, Anda menentukan titik akhir VPC antarmuka untuk Amazon Polly. Jenis titik akhir ini memungkinkan Anda untuk menghubungkan Layanan AWS VPC Anda ke. Titik akhir menyediakan konektivitas yang andal dan dapat diskalakan ke Amazon Polly tanpa memerlukan gateway internet, instance terjemahan alamat jaringan (NAT), atau koneksi VPN. Untuk informasi selengkapnya, lihat [Apa itu VPC Amazon](#) di Panduan Pengguna Amazon VPC.

Endpoint VPC antarmuka didukung oleh AWS PrivateLink, sebuah AWS teknologi yang memungkinkan komunikasi pribadi antara Layanan AWS menggunakan elastic network interface dengan alamat IP pribadi. Untuk informasi selengkapnya, lihat [Baru - AWS PrivateLink untuk Layanan AWS](#).

Langkah-langkah berikut ditujukan untuk para pengguna Amazon VPC. Untuk informasi selengkapnya, lihat [Memulai](#) di Panduan Pengguna Amazon VPC.

Ketersediaan

Titik akhir VPC didukung di semua [Wilayah tempat Amazon Polly](#) didukung. Untuk informasi selengkapnya tentang AWS Wilayah dan Availability Zone, lihat [Infrastruktur AWS Global](#).

Titik akhir FIPS tersedia di wilayah berikut:

- AS Timur (Virginia Utara) (us-east-1)
- AS Timur (Ohio) (us-east-2)
- AS Barat (California Utara) (us-west-1)
- AS Barat (Oregon) (us-west-2)
- AWS GovCloud (US-West) (kami-gov-barat-1)
- Kanada (Pusat) (ca-central-1)

Titik akhir FIPS adalah dari bentuk. com.amazonaws.**REGION**.polly-fips

Membuat titik akhir VPC untuk Amazon Polly

Untuk mulai menggunakan Amazon Polly dengan VPC Anda, buat titik akhir VPC antarmuka untuk Amazon Polly. Layanan yang harus dipilih adalah com.amazonaws.**Region**.polly. Anda tidak perlu mengubah pengaturan apa pun untuk Amazon Polly. Untuk informasi selengkapnya, lihat [Membuat Titik Akhir Antarmuka](#) di Panduan Pengguna Amazon VPC.

Menguji koneksi antara VPC Anda dan Amazon Polly

Setelah Anda membuat titik akhir, Anda dapat menguji koneksi.

Untuk menguji koneksi antara VPC Anda dan titik akhir Amazon Polly Anda

1. Connect ke instans Amazon EC2 yang berada di VPC Anda. Untuk informasi tentang menghubungkan, lihat [Connect ke instans Linux Anda](#) atau [Menyambungkan ke instans Windows Anda](#) dalam dokumentasi Amazon EC2.
2. Dari contoh, gunakan `aws polly describe-voices` dari daftar AWS CLI ke suara Amazon Polly yang tersedia.

Jika respons terhadap perintah menyertakan daftar suara Amazon Polly yang tersedia, perintah telah berhasil, dan titik akhir VPC Anda berfungsi.

Mengontrol akses ke titik akhir Amazon Polly Anda

Kebijakan titik akhir VPC adalah kebijakan sumber daya IAM yang Anda lampirkan ke titik akhir ketika membuat atau mengubah titik akhir. Jika Anda tidak melampirkan kebijakan ketika membuat titik akhir, kami melampirkan kebijakan default untuk Anda sehingga memungkinkan akses penuh ke layanan. Kebijakan titik akhir tidak membatalkan atau mengganti kebijakan pengguna IAM atau kebijakan khusus layanan. Ini adalah kebijakan terpisah untuk mengendalikan akses dari titik akhir ke layanan tertentu.

Kebijakan titik akhir harus ditulis dalam format JSON.

Untuk informasi selengkapnya, lihat [Pengontrolan Akses ke Layanan dengan Titik akhir VPC](#) dalam Panduan Pengguna Amazon VPC.

Berikut ini adalah contoh kebijakan endpoint untuk Amazon Polly. Kebijakan ini memungkinkan pengguna yang terhubung ke Amazon Polly melalui VPC untuk mendeskripsikan suara dan mensintesis ucapan dengan Amazon Polly, serta mencegah mereka melakukan tindakan Amazon Polly lainnya.

```
{
  "Statement": [
    {
      "Sid": "SynthesisAndDescribeVoicesOnly",
      "Principal": "*",
```

```
"Action": [  
    "polly:DescribeVoices",  
    "polly:SynthesizeSpeech"  
],  
"Effect": "Allow",  
"Resource": "*" ]  
}
```

Untuk mengubah kebijakan titik akhir VPC untuk Amazon Polly

1. Buka konsol VPC Amazon di <https://console.aws.amazon.com/vpc>
2. Di panel navigasi, pilih Titik akhir.
3. Jika Anda belum membuat titik akhir untuk Amazon Polly, pilih Buat titik akhir. Kemudian pilih com.amazonaws. **Region**.polly dan pilih Create endpoint.
4. Pilih com.amazonaws. **Region**.polly endpoint, dan pilih tab Policy di bagian bawah layar.
5. Pilih Edit Kebijakan dan buat perubahan pada kebijakan.

Support untuk kunci konteks VPC

Amazon Polly mendukung `aws:SourceVpc` dan kunci `aws:SourceVpce` konteks yang dapat membatasi akses ke VPC tertentu atau titik akhir VPC tertentu. Kunci ini bekerja hanya ketika pengguna menggunakan VPC endpoint. Untuk informasi selengkapnya, lihat [Kunci yang Tersedia untuk Beberapa Layanan](#) di Panduan pengguna IAM.

Mencatat panggilan API Amazon Polly dengan AWS CloudTrail

Amazon Polly terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di Amazon Polly. CloudTrail menangkap semua panggilan API untuk Amazon Polly sebagai acara. Panggilan yang diambil termasuk panggilan dari konsol Amazon Polly dan panggilan kode ke operasi Amazon Polly API. Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara secara berkelanjutan ke bucket Amazon S3, termasuk acara untuk Amazon Polly. Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat peristiwa terbaru di CloudTrail konsol dalam Riwayat acara. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat ke Amazon Polly, alamat IP dari mana permintaan itu dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk mempelajari selengkapnya CloudTrail, termasuk cara mengonfigurasi dan mengaktifkannya, lihat [Panduan AWS CloudTrail Pengguna](#).

Informasi Amazon Polly di CloudTrail

CloudTrail diaktifkan di AWS akun Anda saat Anda membuat akun. Ketika aktivitas acara yang didukung terjadi di Amazon Polly, aktivitas tersebut direkam dalam suatu CloudTrail peristiwa bersama dengan peristiwa AWS layanan lainnya dalam riwayat Acara. Anda dapat melihat, mencari, dan mengunduh acara terbaru di AWS akun Anda. Untuk informasi selengkapnya, lihat [Melihat Acara dengan Riwayat CloudTrail Acara](#).

Untuk catatan peristiwa yang sedang berlangsung di AWS akun Anda, termasuk acara untuk Amazon Polly, buat jejak. Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Secara default, saat Anda membuat jejak di konsol, jejak tersebut berlaku untuk semua AWS Wilayah. Jejak mencatat peristiwa dari semua Wilayah di AWS partisi dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi AWS layanan lain untuk menganalisis lebih lanjut dan menindaklanjuti data peristiwa yang dikumpulkan dalam CloudTrail log. Untuk informasi selengkapnya, lihat berikut:

- [Gambaran Umum untuk Membuat Jejak](#)
- [CloudTrail Layanan dan Integrasi yang Didukung](#)
- [Mengkonfigurasi Notifikasi Amazon SNS untuk CloudTrail](#)

- [Menerima File CloudTrail Log dari Beberapa Wilayah](#) dan [Menerima File CloudTrail Log dari Beberapa Akun](#)

Amazon Polly mendukung pencatatan tindakan berikut sebagai peristiwa dalam file CloudTrail log:

- [DeleteLexicon](#)
- [DescribeVoices](#)
- [GetLexicon](#)
- [GetSpeechSynthesisTask](#)
- [ListLexicons](#)
- [ListSpeechSynthesisTasks](#)
- [PutLexicon](#)
- [StartSpeechSynthesisTask](#)
- [SynthesizeSpeech](#)

Setiap entri peristiwa atau log berisi informasi tentang entitas yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan hal berikut ini:

- Apakah permintaan dibuat dengan pengguna root atau kredensial pengguna AWS Identity and Access Management (IAM).
- Apakah permintaan tersebut dibuat dengan kredensial keamanan sementara untuk satu peran atau pengguna gabungan.
- Apakah permintaan itu dibuat oleh AWS layanan lain.

Untuk informasi lain, lihat [Elemen userIdentity CloudTrail](#).

Contoh: Entri File Log Amazon Polly

Trail adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai file log ke bucket Amazon S3 yang Anda tentukan. CloudTrail file log berisi satu atau lebih entri log. Peristiwa mewakili permintaan tunggal dari sumber manapun dan mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. CloudTrail file log bukanlah jejak tumpukan yang diurutkan dari panggilan API publik, jadi file tersebut tidak muncul dalam urutan tertentu.

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan. SynthesizeSpeech

```
{
  "Records": [
    {
      "awsRegion": "us-east-2",
      "eventID": "19bd70f7-5e60-4cdc-9825-936c552278ae",
      "eventName": "SynthesizeSpeech",
      "eventSource": "polly.amazonaws.com",
      "eventTime": "2016-11-02T03:49:39Z",
      "eventType": "AwsApiCall",
      "eventVersion": "1.05",
      "recipientAccountId": "123456789012",
      "requestID": "414288c2-a1af-11e6-b17f-d7cfc06cb461",
      "requestParameters": {
        "lexiconNames": [
          "SampleLexicon"
        ],
        "engine": "neural",
        "outputFormat": "mp3",
        "sampleRate": "22050",
        "text": "*****",
        "textType": "text",
        "voiceId": "Kendra"
      },
      "responseElements": null,
      "sourceIPAddress": "1.2.3.4",
      "userAgent": "Amazon CLI/Polly 1.10 API 2016-06-10",
      "userIdentity": {
        "accessKeyId": "EXAMPLE_KEY_ID",
        "accountId": "123456789012",
        "arn": "arn:aws:iam::123456789012:user/Alice",
        "principalId": "EX_PRINCIPAL_ID",
        "type": "IAMUser",
        "userName": "Alice"
      }
    }
  ]
}
```

Integrasi CloudWatch dengan Amazon Polly

Saat Anda berinteraksi dengan Amazon Polly, Amazon mengirimkan metrik dan dimensi berikut ke CloudWatch setiap menit. Anda dapat menggunakan prosedur berikut untuk melihat metrik Amazon Polly.

Anda dapat memantau penggunaan Amazon Polly CloudWatch, yang mengumpulkan dan memproses data mentah dari Amazon Polly menjadi metrik yang hampir real-time yang dapat dibaca. Statistik ini dicatat untuk jangka waktu dua minggu, sehingga Anda dapat mengakses `historical information` dan mendapatkan perspektif yang lebih baik tentang kinerja aplikasi atau layanan web Anda. Secara default, data metrik Amazon Polly dikirim ke CloudWatch dalam interval 1 menit. Untuk informasi selengkapnya, lihat [Apa itu Amazon CloudWatch](#) di Panduan CloudWatch Pengguna Amazon.

Mendapatkan CloudWatch Metrik (Konsol)

1. Buka CloudWatch konsol di <https://console.aws.amazon.com/cloudwatch/>.
2. Pada panel navigasi, silakan pilih Metrik.
3. Di panel CloudWatch Metrik menurut Kategori, di bawah kategori metrik untuk Amazon Polly, pilih kategori metrik, lalu di panel atas, gulir ke bawah untuk melihat daftar lengkap metrik.

Mendapatkan CloudWatch metrik di AWS CLI

Kode berikut menampilkan metrik yang tersedia untuk Amazon Polly.

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

Perintah sebelumnya mengembalikan daftar metrik Amazon Polly yang mirip dengan yang berikut ini. `MetricNameElement` mengidentifikasi apa metrik itu.

```
{
  "Metrics": [
    {
      "Namespace": "AWS/Polly",
      "Dimensions": [
        {
          "Name": "Operation",
```

```
        "Value": "SynthesizeSpeech"
      },
    ],
    "MetricName": "ResponseLatency"
  },
  {
    "Namespace": "AWS/Polly",
    "Dimensions": [
      {
        "Name": "Operation",
        "Value": "SynthesizeSpeech"
      }
    ],
    "MetricName": "RequestCharacters"
  }
}
```

Untuk informasi selengkapnya, lihat [GetMetricStatistics](#) di Referensi Amazon CloudWatch API.

Metrik Amazon Polly

Amazon Polly menghasilkan metrik berikut untuk setiap permintaan. Metrik ini dikumpulkan dan dalam interval satu menit dikirim ke CloudWatch tempat yang tersedia.

Metrik	Deskripsi
RequestCharacters	Jumlah karakter dalam permintaan. Ini hanya karakter yang dapat ditagih dan tidak termasuk tag SSML. Dimensi yang Valid: Operasi Statistik yang Valid: Minimum, Maksimum, Rata-rata SampleCount,, Jumlah Unit: Jumlah
ResponseLatency	Latensi antara saat permintaan dibuat dan dimulainya respons streaming. Dimensi yang Valid: Operasi

Metrik	Deskripsi
	Statistik yang Valid: Minimum, Maksimum, Rata-rata, SampleCount Satuan: mikrodetik
2XXCount	Kode tingkat HTTP 200 dikembalikan setelah respons yang berhasil. Dimensi yang Valid: Operasi Statistik yang Valid: Rata-rata SampleCount,, Jumlah Unit: Jumlah
4XXCount	Kode kesalahan tingkat HTTP 400 dikembalikan pada kesalahan. Untuk setiap respons yang berhasil, nol (0) dipancarkan. Dimensi yang Valid: Operasi Statistik yang Valid: Rata-rata SampleCount,, Jumlah Unit: Jumlah
5XXCount	Kode kesalahan tingkat HTTP 500 dikembalikan pada kesalahan. Untuk setiap respons yang berhasil, nol (0) dipancarkan. Dimensi yang Valid: Operasi Statistik yang Valid: Rata-rata SampleCount,, Jumlah Unit: Jumlah

Dimensi untuk Metrik Amazon Polly

Metrik Amazon Polly menggunakan namespace AWS/Polly dan menyediakan metrik untuk dimensi berikut:

Dimensi	Deskripsi
Operation	Metrik dikelompokkan berdasarkan metode API yang mereka rujuk. Nilai yang mungkin adalah <code>SynthesizeSpeech</code> , <code>PutLexicon</code> , <code>DescribeVoices</code> , dll.

Referensi API Amazon Polly

Amazon Polly adalah layanan web yang memudahkan untuk mensintesis ucapan dari teks.

Layanan Amazon Polly menyediakan operasi API untuk mensintesis ucapan berkualitas tinggi dari teks biasa dan Speech Synthesis Markup Language (SSML), bersama dengan mengelola leksikon pengucapan yang memungkinkan Anda mendapatkan hasil terbaik untuk domain aplikasi Anda.

Panggilan API yang diautentikasi harus ditandatangani menggunakan Proses Penandatanganan Versi Tanda Tangan 4. Untuk informasi selengkapnya, lihat [Menandatangani Permintaan AWS API](#) di bagian Referensi Umum Amazon Web Services.

Topik

- [Tindakan](#)
- [Tipe Data](#)
- [Jenis Kesalahan Umum](#)
- [Parameter Umum](#)

Tindakan

Tindakan berikut didukung:

- [DeleteLexicon](#)
- [DescribeVoices](#)
- [GetLexicon](#)
- [GetSpeechSynthesisTask](#)
- [ListLexicons](#)
- [ListSpeechSynthesisTasks](#)
- [PutLexicon](#)
- [StartSpeechSynthesisStream](#)
- [StartSpeechSynthesisTask](#)
- [SynthesizeSpeech](#)

DeleteLexicon

Menghapus leksikon pengucapan tertentu yang disimpan dalam file. Wilayah AWS Sebuah leksikon yang telah dihapus tidak tersedia untuk sintesis ucapan, juga tidak mungkin untuk mengambilnya menggunakan salah satu atau. GetLexicon ListLexicon APIs

Untuk informasi selengkapnya, lihat [Mengelola Leksikon](#).

Minta Sintaks

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

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

LexiconName

Nama leksikon yang akan dihapus. Harus menjadi leksikon yang ada di wilayah tersebut.

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

Wajib: Ya

Isi Permintaan

Permintaan tidak memiliki isi permintaan.

Sintaks Respons

```
HTTP/1.1 200
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200 dengan isi HTTP kosong.

Kesalahan

LexiconNotFoundException

Amazon Polly tidak dapat menemukan leksikon yang ditentukan. Ini bisa disebabkan oleh leksikon yang hilang, namanya salah eja atau menentukan leksikon yang ada di wilayah berbeda.

Verifikasi bahwa leksikon ada, ada di wilayah (lihat [ListLexicons](#)) dan Anda mengeja namanya dieja dengan benar. Kemudian coba lagi.

Kode Status HTTP: 404

ServiceFailureException

Kondisi yang tidak diketahui telah menyebabkan kegagalan layanan.

Kode Status HTTP: 500

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)
- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

DescribeVoices

Mengembalikan daftar suara yang tersedia untuk digunakan saat meminta sintesis ucapan. Setiap suara berbicara bahasa tertentu, baik laki-laki atau perempuan, dan diidentifikasi oleh ID, yang merupakan versi ASCII dari nama suara.

Saat mensintesis speech (`SynthesizeSpeech`), Anda memberikan ID suara untuk suara yang Anda inginkan dari daftar suara yang dikembalikan oleh `DescribeVoices`.

Misalnya, Anda ingin aplikasi pembaca berita Anda membaca berita dalam bahasa tertentu, tetapi memberi pengguna opsi untuk memilih suara. Dengan menggunakan `DescribeVoices` operasi ini, Anda dapat memberi pengguna daftar suara yang tersedia untuk dipilih.

Anda dapat secara opsional menentukan kode bahasa untuk memfilter suara yang tersedia. Misalnya, jika Anda menentukan `en-US`, operasi mengembalikan daftar semua suara bahasa Inggris AS yang tersedia.

Operasi ini memerlukan izin untuk menjalankan tindakan `polly:DescribeVoices`.

Minta Sintaks

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

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

[Engine](#)

Menentukan mesin (`standard`, `neural`, `long-form` atau `generative`) yang digunakan oleh Amazon Polly saat memproses teks input untuk sintesis ucapan.

Nilai Valid: `standard` | `neural` | `long-form` | `generative`

[IncludeAdditionalLanguageCodes](#)

Nilai Boolean menunjukkan apakah akan mengembalikan suara bilingual yang menggunakan bahasa tertentu sebagai bahasa tambahan. Misalnya, jika Anda meminta semua bahasa yang menggunakan bahasa Inggris AS (`en-US`), dan ada suara Italia yang berbicara bahasa Italia (`IT-`

IT) dan Inggris AS, suara itu akan disertakan jika Anda menentukan `yes` tetapi tidak jika Anda menentukan `no`.

[LanguageCode](#)

Tag identifikasi bahasa (kode ISO 639 untuk nama bahasa-ISO 3166 kode negara) untuk memfilter daftar suara yang dikembalikan. Jika Anda tidak menentukan parameter opsional ini, semua suara yang tersedia akan dikembalikan.

Nilai Valid: `arb` | `cmn-CN` | `cy-GB` | `da-DK` | `de-DE` | `en-AU` | `en-GB` | `en-GB-WLS` | `en-IN` | `en-US` | `es-ES` | `es-MX` | `es-US` | `fr-CA` | `fr-FR` | `is-IS` | `it-IT` | `ja-JP` | `hi-IN` | `ko-KR` | `nb-NO` | `nl-NL` | `pl-PL` | `pt-BR` | `pt-PT` | `ro-RO` | `ru-RU` | `sv-SE` | `tr-TR` | `en-NZ` | `en-ZA` | `ca-ES` | `de-AT` | `yue-CN` | `ar-AE` | `fi-FI` | `en-IE` | `nl-BE` | `fr-BE` | `cs-CZ` | `de-CH` | `en-SG`

[NextToken](#)

Token pagination buram dikembalikan dari operasi sebelumnya `DescribeVoices`. Jika ada, ini menunjukkan di mana harus melanjutkan daftar.

Batasan Panjang: Panjang minimum sebesar 0. Panjang maksimum 4096.

Isi Permintaan

Permintaan tidak memiliki isi permintaan.

Sintaks Respons

```
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" ]
    }
  ]
}
```

```
}  
]  
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

[NextToken](#)

Token pagination yang akan digunakan dalam permintaan berikutnya untuk melanjutkan daftar suara. `NextToken` dikembalikan hanya jika respons terpotong.

Tipe: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 4096.

[Voices](#)

Daftar suara dengan propertinya.

Tipe: Array objek [Voice](#)

Kesalahan

`InvalidNextTokenException`

`NextToken` itu tidak valid. Verifikasi bahwa itu dieja dengan benar, dan kemudian coba lagi.

Kode Status HTTP: 400

`ServiceFailureException`

Kondisi yang tidak diketahui telah menyebabkan kegagalan layanan.

Kode Status HTTP: 500

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)
- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

GetLexicon

Mengembalikan isi dari leksikon pengucapan tertentu yang disimpan dalam file Wilayah AWS. Untuk informasi selengkapnya, lihat [Mengelola Leksikon](#).

Minta Sintaks

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

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

[LexiconName](#)

Nama leksikon.

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

Wajib: Ya

Isi Permintaan

Permintaan tidak memiliki isi permintaan.

Sintaks Respons

```
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  
  }  
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

[Lexicon](#)

Objek leksikon yang menyediakan nama dan konten string leksikon.

Tipe: Objek [Lexicon](#)

[LexiconAttributes](#)

Metadata leksikon, termasuk alfabet fonetik yang digunakan, kode bahasa, leksikon ARN, jumlah leksem yang ditentukan dalam leksikon, dan ukuran leksikon dalam byte.

Tipe: Objek [LexiconAttributes](#)

Kesalahan

LexiconNotFoundException

Amazon Polly tidak dapat menemukan leksikon yang ditentukan. Ini bisa disebabkan oleh leksikon yang hilang, namanya salah eja atau menentukan leksikon yang ada di wilayah berbeda.

Verifikasi bahwa leksikon ada, ada di wilayah (lihat [ListLexicons](#)) dan Anda mengeja namanya dieja dengan benar. Kemudian coba lagi.

Kode Status HTTP: 404

ServiceFailureException

Kondisi yang tidak diketahui telah menyebabkan kegagalan layanan.

Kode Status HTTP: 500

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)
- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

GetSpeechSynthesisTask

Mengambil SpeechSynthesisTask objek tertentu berdasarkan TaskId-nya. Objek ini berisi informasi tentang tugas sintesis ucapan yang diberikan, termasuk status tugas, dan tautan ke ember S3 yang berisi output tugas.

Minta Sintaks

```
GET /v1/synthesisTasks/TaskId HTTP/1.1
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

TaskId

Amazon Polly menghasilkan pengidentifikasi untuk tugas sintesis ucapan.

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

Wajib: Ya

Isi Permintaan

Permintaan tidak memiliki isi permintaan.

Sintaks Respons

```
HTTP/1.1 200
Content-type: application/json

{
  "SpeechSynthesisTask": {
    "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"
}
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

[SynthesisTask](#)

SynthesisTask objek yang menyediakan informasi dari tugas yang diminta, termasuk format output, waktu pembuatan, status tugas, dan sebagainya.

Tipe: Objek [SynthesisTask](#)

Kesalahan

InvalidTaskIdException

ID Tugas yang diberikan tidak valid. Harap berikan ID Tugas yang valid dan coba lagi.

Kode Status HTTP: 400

ServiceFailureException

Kondisi yang tidak diketahui telah menyebabkan kegagalan layanan.

Kode Status HTTP: 500

SynthesisTaskNotFoundException

Tugas Sintesis Pidato dengan ID Tugas yang diminta tidak dapat ditemukan.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)
- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

ListLexicons

Mengembalikan daftar leksikon pengucapan yang disimpan dalam file Wilayah AWS. Untuk informasi selengkapnya, lihat [Mengelola Leksikon](#).

Minta Sintaks

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

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

[NextToken](#)

Token pagination buram dikembalikan dari operasi sebelumnya `ListLexicons`. Jika ada, tunjukkan di mana harus melanjutkan daftar leksikon.

Batasan Panjang: Panjang minimum sebesar 0. Panjang maksimum 4096.

Isi Permintaan

Permintaan tidak memiliki isi permintaan.

Sintaks Respons

```
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"  
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

[Lexicons](#)

Daftar nama dan atribut leksikon.

Tipe: Array objek [LexiconDescription](#)

[NextToken](#)

Token pagination yang akan digunakan dalam permintaan berikutnya untuk melanjutkan daftar leksikon. NextToken dikembalikan hanya jika respons terpotong.

Tipe: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 4096.

Kesalahan

InvalidNextTokenException

NextToken itu tidak valid. Verifikasi bahwa itu dieja dengan benar, dan kemudian coba lagi.

Kode Status HTTP: 400

ServiceFailureException

Kondisi yang tidak diketahui telah menyebabkan kegagalan layanan.

Kode Status HTTP: 500

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)
- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

ListSpeechSynthesisTasks

Mengembalikan daftar SpeechSynthesisTask objek yang diurutkan berdasarkan tanggal pembuatannya. Operasi ini dapat memfilter tugas berdasarkan statusnya, misalnya, memungkinkan pengguna untuk membuat daftar hanya tugas yang selesai.

Minta Sintaks

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

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

MaxResults

Jumlah maksimum tugas sintesis ucapan yang dikembalikan dalam operasi Daftar.

Rentang yang Valid: Nilai minimum 1. Nilai maksimum 100.

NextToken

Token pagination yang akan digunakan dalam permintaan berikutnya untuk melanjutkan daftar tugas sintesis ucapan.

Batasan Panjang: Panjang minimum sebesar 0. Panjang maksimum 4096.

Status

Status tugas sintesis ucapan dikembalikan dalam operasi Daftar

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

Isi Permintaan

Permintaan tidak memiliki isi permintaan.

Sintaks Respons

```
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"
    }
  ]
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

NextToken

Token pagination buram dikembalikan dari operasi Daftar sebelumnya dalam permintaan ini. Jika ada, ini menunjukkan di mana harus melanjutkan daftar.

Tipe: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 4096.

SynthesisTasks

Daftar SynthesisTask objek yang menyediakan informasi dari tugas yang ditentukan dalam permintaan daftar, termasuk format output, waktu pembuatan, status tugas, dan sebagainya.

Tipe: Array objek [SynthesisTask](#)

Kesalahan

InvalidNextTokenException

NextToken Itu tidak valid. Verifikasi bahwa itu dieja dengan benar, dan kemudian coba lagi.

Kode Status HTTP: 400

ServiceFailureException

Kondisi yang tidak diketahui telah menyebabkan kegagalan layanan.

Kode Status HTTP: 500

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)
- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

PutLexicon

Menyimpan leksikon pengucapan dalam file Wilayah AWS. Jika leksikon dengan nama yang sama sudah ada di wilayah tersebut, itu ditimpa oleh leksikon baru. Operasi leksikon memiliki konsistensi akhirnya, oleh karena itu, mungkin perlu beberapa waktu sebelum leksikon tersedia untuk operasi SynthesizeSpeech.

Untuk informasi selengkapnya, lihat [Mengelola Leksikon](#).

Minta Sintaks

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

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

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

LexiconName

Nama leksikon. Nama harus mengikuti format ekspresi reguler [0-9A-Za-Z]{1,20}. Artinya, namanya adalah string alfanumerik peka huruf besar/kecil hingga 20 karakter.

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

Wajib: Ya

Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

Content

Isi leksikon PLS sebagai data string.

Tipe: String

Diperlukan: Ya

Sintaksis Respons

```
HTTP/1.1 200
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200 dengan isi HTTP kosong.

Kesalahan

InvalidLexiconException

Amazon Polly tidak dapat menemukan leksikon yang ditentukan. Verifikasi bahwa nama leksikon dieja dengan benar, lalu coba lagi.

Kode Status HTTP: 400

LexiconSizeExceededException

Ukuran maksimum leksikon yang ditentukan akan dilampaui oleh operasi ini.

Kode Status HTTP: 400

MaxLexemeLengthExceededException

Ukuran maksimum leksem akan dilampaui oleh operasi ini.

Kode Status HTTP: 400

MaxLexiconsNumberExceededException

Jumlah leksikon maksimum akan terlampaui oleh operasi ini.

Kode Status HTTP: 400

ServiceFailureException

Kondisi yang tidak diketahui telah menyebabkan kegagalan layanan.

Kode Status HTTP: 500

UnsupportedPlsAlphabetException

Alfabet yang ditentukan oleh leksikon bukanlah alfabet yang didukung. Nilai yang valid adalah x-sampa dan ipa.

Kode Status HTTP: 400

UnsupportedPlsLanguageException

Bahasa yang ditentukan dalam leksikon tidak didukung. Untuk daftar bahasa yang didukung, lihat [Atribut Leksikon](#).

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)
- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

StartSpeechSynthesisStream

Mensintesis input UTF-8, teks biasa, atau SSML melalui koneksi streaming dua arah. Tentukan parameter sintesis di header HTTP/2, kirim teks secara bertahap sebagai peristiwa pada aliran input, dan terima audio yang disintesis saat tersedia.

Operasi ini berfungsi sebagai mitra dua arah untuk: SynthesizeSpeech

- [SynthesizeSpeech](#)

Minta Sintaks

```
POST /v1/synthesisStream HTTP/1.1
x-amzn-Engine: Engine
x-amzn-LanguageCode: LanguageCode
x-amzn-LexiconNames: LexiconNames
x-amzn-OutputFormat: OutputFormat
x-amzn-SampleRate: SampleRate
x-amzn-VoiceId: VoiceId
Content-type: application/json

{
  "CloseStreamEvent": {
  },
  "TextEvent": {
    "FlushStreamConfiguration": {
      "Force": boolean
    },
    "Text": "string",
    "TextType": "string"
  }
}
```

Parameter Permintaan URI

Permintaan menggunakan parameter URI berikut.

Engine

Menentukan mesin untuk Amazon Polly untuk digunakan saat memproses teks input untuk sintesis ucapan. Saat ini, hanya *generative* mesin yang didukung. Jika Anda menentukan suara yang tidak didukung oleh mesin yang dipilih, Amazon Polly menampilkan kesalahan.

Nilai yang Valid: `standard` | `neural` | `long-form` | `generative`

Wajib: Ya

LanguageCode

Parameter opsional yang menetapkan kode bahasa untuk permintaan sintesis ucapan. Tentukan parameter ini hanya saat menggunakan suara bilingual. Jika suara dwibahasa digunakan dan tidak ada kode bahasa yang ditentukan, Amazon Polly menggunakan bahasa default suara dwibahasa.

Nilai Valid: `arb` | `cmn-CN` | `cy-GB` | `da-DK` | `de-DE` | `en-AU` | `en-GB` | `en-GB-WLS` | `en-IN` | `en-US` | `es-ES` | `es-MX` | `es-US` | `fr-CA` | `fr-FR` | `is-IS` | `it-IT` | `ja-JP` | `hi-IN` | `ko-KR` | `nb-NO` | `nl-NL` | `pl-PL` | `pt-BR` | `pt-PT` | `ro-RO` | `ru-RU` | `sv-SE` | `tr-TR` | `en-NZ` | `en-ZA` | `ca-ES` | `de-AT` | `yue-CN` | `ar-AE` | `fi-FI` | `en-IE` | `nl-BE` | `fr-BE` | `cs-CZ` | `de-CH` | `en-SG`

LexiconNames

Nama-nama satu atau lebih leksikon pengucapan untuk layanan yang akan diterapkan selama sintesis. Amazon Polly menerapkan leksikon hanya jika bahasa leksikon cocok dengan bahasa suara.

Anggota Array: Jumlah maksimum 5 item.

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

OutputFormat

Format audio untuk pidato yang disintesis. Saat ini, Amazon Polly tidak mendukung tanda bicara JSON.

Nilai yang Valid: `json` | `mp3` | `ogg_opus` | `ogg_vorbis` | `pcm` | `mulaw` | `alaw`

Wajib: Ya

SampleRate

Frekuensi audio, ditentukan dalam Hz.

VoiceId

Suara untuk digunakan dalam sintesis. Untuk mendapatkan daftar suara yang tersedia IDs, gunakan [DescribeVoices](#) operasi.

Nilai yang Valid: Aditi | Amy | Astrid | Bianca | Brian | Camila | Carla | Carmen | Celine | Chantal | Conchita | Cristiano | Dora | Emma | Enrique | Ewa | Filiz | Gabrielle | Geraint | Giorgio | Gwyneth | Hans | Ines | Ivy | Jacek | Jan | Joanna | Joey | Justin | Karl | Kendra | Kevin | Kimberly | Lea | Liv | Lotte | Lucia | Lupe | Mads | Maja | Marlene | Mathieu | Matthew | Maxim | Mia | Miguel | Mizuki | Naja | Nicole | Olivia | Penelope | Raveena | Ricardo | Ruben | Russell | Salli | Seoyeon | Takumi | Tatyana | Vicki | Vitoria | Zeina | Zhiyu | Aria | Ayanda | Arlet | Hannah | Arthur | Daniel | Liam | Pedro | Kajal | Hiujin | Laura | Elin | Ida | Suvi | Ola | Hala | Andres | Sergio | Remi | Adriano | Thiago | Ruth | Stephen | Kazuha | Tomoko | Niamh | Sofie | Lisa | Isabelle | Zayd | Danielle | Gregory | Burcu | Jitka | Sabrina | Jasmine | Jihye | Ambre | Beatrice | Florian | Lennart | Lorenzo | Tiffany

Wajib: Ya

Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

CloseStreamEvent

Peristiwa yang menunjukkan akhir aliran input.

Tipe: Objek [CloseStreamEvent](#)

Wajib: Tidak

TextEvent

Sebuah peristiwa teks yang berisi konten yang akan disintesis.

Tipe: Objek [TextEvent](#)

Wajib: Tidak

Sintaksis Respons

```
HTTP/1.1 200
Content-type: application/json

{
  "AudioEvent": {
    "AudioChunk": blob
  },
  "ServiceFailureException": {
  },
  "ServiceQuotaExceededException": {
  },
  "StreamClosedEvent": {
    "RequestCharacters": number
  },
  "ThrottlingException": {
  },
  "ValidationException": {
  }
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

AudioEvent

Acara audio yang berisi pidato yang disintesis.

Tipe: Objek AudioEvent

ServiceFailureException

Kondisi yang tidak diketahui telah menyebabkan kegagalan layanan.

Jenis: Pengecualian

Kode Status HTTP: 500

ServiceQuotaExceededException

Pengecualian yang menunjukkan kuota layanan akan terlampaui.

Jenis: Pengecualian

Kode Status HTTP: 402

[StreamClosedEvent](#)

Sebuah peristiwa, dengan informasi ringkasan, yang menunjukkan aliran telah ditutup.

Tipe: Objek [StreamClosedEvent](#)

[ThrottlingException](#)

Pengecualian yang menunjukkan permintaan dibatasi.

Jenis: Pengecualian

Kode Status HTTP: 400

[ValidationException](#)

Pengecualian yang menunjukkan validasi input gagal.

Jenis: Pengecualian

Kode Status HTTP: 400

Kesalahan

ServiceFailureException

Kondisi yang tidak diketahui telah menyebabkan kegagalan layanan.

Kode Status HTTP: 500

ServiceQuotaExceededException

Permintaan akan menyebabkan kuota layanan terlampaui.

quotaCode

Kode kuota mengidentifikasi kuota tertentu.

serviceCode

Kode layanan yang mengidentifikasi layanan asal.

Kode Status HTTP: 402

ThrottlingException

Permintaan ditolak karena pembatasan permintaan.

throttlingReasons

Daftar alasan yang menjelaskan mengapa permintaan itu dibatasi.

Kode Status HTTP: 400

ValidationException

Input gagal memenuhi kendala yang ditentukan oleh layanan.

fields

Bidang yang menyebabkan kesalahan validasi.

reason

Alasan permintaan gagal validasi.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)
- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

StartSpeechSynthesisTask

Memungkinkan pembuatan tugas sintesis asinkron, dengan memulai yang baru.

SpeechSynthesisTask Operasi ini memerlukan semua informasi standar yang diperlukan untuk sintesis ucapan, ditambah nama bucket Amazon S3 untuk layanan untuk menyimpan output tugas sintesis dan dua parameter opsional (OutputS3KeyPrefix dan SnsTopicArn). Setelah tugas sintesis dibuat, operasi ini akan mengembalikan SpeechSynthesisTask objek, yang akan mencakup pengidentifikasi tugas ini serta status saat ini. SpeechSynthesisTask Objek tersedia selama 72 jam setelah memulai tugas sintesis asinkron.

Minta Sintaks

```
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"
}
```

Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

Engine

Menentukan mesin (`standard`, `neural`, `long-form` atau `generative`) untuk Amazon Polly untuk digunakan saat memproses teks input untuk sintesis ucapan. Menggunakan suara yang tidak didukung untuk mesin yang dipilih akan mengakibatkan kesalahan.

Tipe: String

Nilai yang Valid: `standard` | `neural` | `long-form` | `generative`

Wajib: Tidak

LanguageCode

Kode bahasa opsional untuk permintaan Sintesis Pidato. Ini hanya diperlukan jika menggunakan suara bilingual, seperti Aditi, yang dapat digunakan untuk bahasa Inggris India (`En-in`) atau Hindi (`Hi-in`).

Jika suara dwibahasa digunakan dan tidak ada kode bahasa yang ditentukan, Amazon Polly menggunakan bahasa default suara dwibahasa. Bahasa default untuk suara apa pun adalah yang dikembalikan oleh [DescribeVoices](#) operasi untuk `LanguageCode` parameter. Misalnya, jika tidak ada kode bahasa yang ditentukan, Aditi akan menggunakan bahasa Inggris India daripada bahasa Hindi.

Tipe: String

Nilai yang Valid: `arb` | `cmn-CN` | `cy-GB` | `da-DK` | `de-DE` | `en-AU` | `en-GB` | `en-GB-WLS` | `en-IN` | `en-US` | `es-ES` | `es-MX` | `es-US` | `fr-CA` | `fr-FR` | `is-IS` | `it-IT` | `ja-JP` | `hi-IN` | `ko-KR` | `nb-NO` | `nl-NL` | `pl-PL` | `pt-BR` | `pt-PT` | `ro-RO` | `ru-RU` | `sv-SE` | `tr-TR` | `en-NZ` | `en-ZA` | `ca-ES` | `de-AT` | `yue-CN` | `ar-AE` | `fi-FI` | `en-IE` | `nl-BE` | `fr-BE` | `cs-CZ` | `de-CH` | `en-SG`

Wajib: Tidak

LexiconNames

Daftar satu atau lebih nama leksikon pengucapan yang Anda inginkan untuk diterapkan layanan selama sintesis. Leksikon diterapkan hanya jika bahasa leksikon sama dengan bahasa suara.

Tipe: Array string

Anggota Array: Jumlah maksimum 5 item.

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

Wajib: Tidak

OutputFormat

Format di mana output yang dikembalikan akan dikodekan. Untuk streaming audio, ini akan menjadi mp3, ogg_vorbis, ogg_opus, mu-law, a-law, atau pcm. Untuk tanda bicara, ini akan menjadi json.

Tipe: String

Nilai yang Valid: `json | mp3 | ogg_opus | ogg_vorbis | pcm | mulaw | alaw`

Wajib: Ya

OutputS3BucketName

Nama bucket Amazon S3 tempat file output akan disimpan.

Tipe: String

Pola: `^[a-z0-9][\.\-a-z0-9]{1,61}[a-z0-9]$`

Wajib: Ya

OutputS3KeyPrefix

Awalan kunci Amazon S3 untuk file ucapan keluaran.

Tipe: String

Pola: `^[0-9a-zA-Z\^\!\\-_\.\*\'\(\):;\$@=+\\,\?&]{0,800}$`

Wajib: Tidak

SampleRate

Frekuensi audio yang ditentukan dalam Hz.

Nilai yang valid untuk mp3 dan ogg_vorbis adalah "8000", "16000", "22050", dan "24000". Nilai default untuk suara standar adalah "22050". Nilai default untuk suara saraf adalah "24000". Nilai default untuk suara bentuk panjang adalah "24000". Nilai default untuk suara generatif adalah "24000".

Nilai yang valid untuk pcm adalah "8000" dan "16000" Nilai default adalah "16000".

Nilai valid untuk ogg_opus adalah "48000".

Nilai yang valid untuk mu-law dan a-law adalah "8000".

Tipe: String

Wajib: Tidak

[SnsTopicArn](#)

ARN untuk topik SNS secara opsional digunakan untuk memberikan pemberitahuan status untuk tugas sintesis ucapan.

Tipe: String

Pola: `^arn:aws(-(cn|iso(-b)?|us-gov))?:sns:[a-z0-9_-]{1,50}:\d{12}:[a-zA-Z0-9_-]{1,251}([a-zA-Z0-9_-]{0,5}|\.fifo)$`

Wajib: Tidak

[SpeechMarkTypes](#)

Jenis tanda bicara yang dikembalikan untuk teks input.

Tipe: Array string

Anggota Array: Jumlah maksimum 4 item.

Nilai yang Valid: sentence | ssm1 | viseme | word

Wajib: Tidak

[Text](#)

Teks masukan untuk mensintesis. Jika Anda menentukan ssm1 sebagai TextType, ikuti format SSML untuk teks input.

Tipe: String

Diperlukan: Ya

[TextType](#)

Menentukan apakah teks input adalah teks biasa atau SSML. Nilai defaultnya adalah teks biasa.

Tipe: String

Nilai yang Valid: ssm1 | text

Wajib: Tidak

[Voiceld](#)

ID suara untuk digunakan untuk sintesis.

Tipe: String

Nilai yang Valid: Aditi | Amy | Astrid | Bianca | Brian | Camila | Carla | Carmen | Celine | Chantal | Conchita | Cristiano | Dora | Emma | Enrique | Ewa | Filiz | Gabrielle | Geraint | Giorgio | Gwyneth | Hans | Ines | Ivy | Jacek | Jan | Joanna | Joey | Justin | Karl | Kendra | Kevin | Kimberly | Lea | Liv | Lotte | Lucia | Lupe | Mads | Maja | Marlene | Mathieu | Matthew | Maxim | Mia | Miguel | Mizuki | Naja | Nicole | Olivia | Penelope | Raveena | Ricardo | Ruben | Russell | Salli | Seoyeon | Takumi | Tatyana | Vicki | Vitoria | Zeina | Zhiyu | Aria | Ayanda | Arlet | Hannah | Arthur | Daniel | Liam | Pedro | Kajal | Hiujin | Laura | Elin | Ida | Suvi | Ola | Hala | Andres | Sergio | Remi | Adriano | Thiago | Ruth | Stephen | Kazuha | Tomoko | Niamh | Sofie | Lisa | Isabelle | Zayd | Danielle | Gregory | Burcu | Jitka | Sabrina | Jasmine | Jihye | Ambre | Beatrice | Florian | Lennart | Lorenzo | Tiffany

Wajib: Ya

Sintaksis Respons

```
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"
}
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

[SynthesisTask](#)

SynthesisTask objek yang memberikan informasi dan atribut tentang tugas sintesis ucapan yang baru dikirimkan.

Tipe: Objek [SynthesisTask](#)

Kesalahan

EngineNotSupportedException

Mesin ini tidak kompatibel dengan suara yang Anda tunjuk. Pilih suara baru yang kompatibel dengan mesin atau ganti mesin dan nyalakan kembali operasi.

Kode Status HTTP: 400

InvalidS3BucketException

Nama bucket Amazon S3 yang disediakan tidak valid. Silakan periksa masukan Anda dengan persyaratan penamaan bucket S3 dan coba lagi.

Kode Status HTTP: 400

InvalidS3KeyException

Awalan key Amazon S3 yang disediakan tidak valid. Harap berikan nama kunci objek S3 yang valid.

Kode Status HTTP: 400

InvalidSampleRateException

Tingkat sampel yang ditentukan tidak valid.

Kode Status HTTP: 400

InvalidSnsTopicArnException

Topik SNS yang disediakan ARN tidak valid. Harap berikan ARN topik SNS yang valid dan coba lagi.

Kode Status HTTP: 400

InvalidSsmlException

SSML yang Anda berikan tidak valid. Verifikasi sintaks SSML, ejaan tag dan nilai, lalu coba lagi.

Kode Status HTTP: 400

LanguageNotSupportedException

Bahasa yang ditentukan saat ini tidak didukung oleh Amazon Polly dalam kapasitas ini.

Kode Status HTTP: 400

LexiconNotFoundException

Amazon Polly tidak dapat menemukan leksikon yang ditentukan. Ini bisa disebabkan oleh leksikon yang hilang, namanya salah eja atau menentukan leksikon yang ada di wilayah berbeda.

Verifikasi bahwa leksikon ada, ada di wilayah (lihat [ListLexicons](#)) dan Anda mengeja namanya dieja dengan benar. Kemudian coba lagi.

Kode Status HTTP: 404

MarksNotSupportedForFormatException

Tanda ucapan tidak didukung untuk yang OutputFormat dipilih. Tanda ucapan hanya tersedia untuk konten dalam json format.

Kode Status HTTP: 400

ServiceFailureException

Kondisi yang tidak diketahui telah menyebabkan kegagalan layanan.

Kode Status HTTP: 500

SsmlMarksNotSupportedForTextTypeException

Tanda ucapan SSML tidak didukung untuk input tipe teks biasa.

Kode Status HTTP: 400

TextLengthExceededException

Nilai parameter “Teks” lebih panjang dari batas yang diterima. Untuk SynthesizeSpeech API, batas untuk teks input adalah maksimum 6000 karakter total, yang tidak lebih dari 3000 karakter dapat ditagih. Untuk StartSpeechSynthesisTask API, maksimum adalah 200.000 karakter, yang tidak lebih dari 100.000 karakter dapat ditagih. Tag SSML tidak dihitung sebagai karakter yang ditagih.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)
- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

SynthesizeSpeech

Mensintesis input UTF-8, teks biasa atau SSML, ke aliran byte. Masukan SSML harus valid, SSML yang terbentuk dengan baik. Beberapa huruf mungkin tidak tersedia dengan semua suara (misalnya, Cyrillic mungkin tidak dibaca sama sekali oleh suara bahasa Inggris) kecuali pemetaan fonem digunakan. Untuk informasi selengkapnya, lihat [Cara Kerjanya](#).

Minta Sintaks

```
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"
}
```

Parameter Permintaan URI

Permintaan tidak menggunakan parameter URI apa pun.

Isi Permintaan

Permintaan menerima data berikut dalam format JSON.

[Engine](#)

Menentukan mesin (standard,, neurallong-form, ataugenerative) untuk Amazon Polly untuk digunakan saat memproses teks input untuk sintesis ucapan. Sediakan mesin yang didukung oleh suara yang Anda pilih. Jika Anda tidak menyediakan mesin, mesin standar dipilih secara default. Jika suara yang dipilih tidak didukung oleh mesin standar, ini akan mengakibatkan kesalahan. [Untuk informasi tentang suara Amazon Polly dan suara mana yang tersedia untuk setiap mesin, lihat Suara yang Tersedia.](#)

Tipe: String

Nilai yang Valid: `standard` | `neural` | `long-form` | `generative`

Wajib: Tidak

LanguageCode

Kode bahasa opsional untuk permintaan Synthesize Speech. Ini hanya diperlukan jika menggunakan suara bilingual, seperti Aditi, yang dapat digunakan untuk bahasa Inggris India (En-in) atau Hindi (Hi-in).

Jika suara bilingual digunakan dan tidak ada kode bahasa yang ditentukan, Amazon Polly menggunakan bahasa default suara dwibahasa. Bahasa default untuk suara apa pun adalah yang dikembalikan oleh [DescribeVoices](#) operasi untuk LanguageCode parameter. Misalnya, jika tidak ada kode bahasa yang ditentukan, Aditi akan menggunakan bahasa Inggris India daripada bahasa Hindi.

Tipe: String

Nilai yang Valid: `arb` | `cmn-CN` | `cy-GB` | `da-DK` | `de-DE` | `en-AU` | `en-GB` | `en-GB-WLS` | `en-IN` | `en-US` | `es-ES` | `es-MX` | `es-US` | `fr-CA` | `fr-FR` | `is-IS` | `it-IT` | `ja-JP` | `hi-IN` | `ko-KR` | `nb-NO` | `nl-NL` | `pl-PL` | `pt-BR` | `pt-PT` | `ro-RO` | `ru-RU` | `sv-SE` | `tr-TR` | `en-NZ` | `en-ZA` | `ca-ES` | `de-AT` | `yue-CN` | `ar-AE` | `fi-FI` | `en-IE` | `nl-BE` | `fr-BE` | `cs-CZ` | `de-CH` | `en-SG`

Wajib: Tidak

LexiconNames

Daftar satu atau lebih nama leksikon pengucapan yang Anda inginkan untuk diterapkan layanan selama sintesis. Leksikon diterapkan hanya jika bahasa leksikon sama dengan bahasa suara. Untuk informasi tentang menyimpan leksikon, lihat [PutLexicon](#).

Tipe: Array string

Anggota Array: Jumlah maksimum 5 item.

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

Wajib: Tidak

OutputFormat

Format di mana output yang dikembalikan akan dikodekan. Untuk streaming audio, ini akan menjadi mp3, ogg_vorbis, ogg_opus, mu-law, a-law atau pcm. Untuk tanda bicara, ini akan menjadi json.

Saat pcm digunakan, konten yang dikembalikan audio/pcm dalam format 16-bit, 1 saluran (mono), endian kecil yang ditandatangani.

Tipe: String

Nilai yang Valid: json | mp3 | ogg_opus | ogg_vorbis | pcm | mulaw | alaw

Wajib: Ya

SampleRate

Frekuensi audio yang ditentukan dalam Hz.

Nilai yang valid untuk mp3 dan ogg_vorbis adalah "8000", "16000", "22050", "24000", "44100" dan "48000". Nilai default untuk suara standar adalah "22050". Nilai default untuk suara saraf adalah "24000". Nilai default untuk suara bentuk panjang adalah "24000". Nilai default untuk suara generatif adalah "24000".

Nilai yang valid untuk pcm adalah "8000" dan "16000" Nilai default adalah "16000".

Nilai valid untuk ogg_opus adalah "48000".

Nilai yang valid untuk mu-law dan a-law adalah "8000".

Tipe: String

Wajib: Tidak

SpeechMarkTypes

Jenis tanda bicara yang dikembalikan untuk teks input.

Tipe: Array string

Anggota Array: Jumlah maksimum 4 item.

Nilai yang Valid: sentence | ssm1 | viseme | word

Wajib: Tidak

Text

Masukan teks untuk mensintesis. Jika Anda menentukan `ssml` sebagai `TextType`, ikuti format SSML untuk teks input.

Tipe: String

Diperlukan: Ya

TextType

Menentukan apakah teks input adalah teks biasa atau SSML. Nilai default adalah teks biasa. Untuk informasi selengkapnya, lihat [Menggunakan SSML](#).

Tipe: String

Nilai yang Valid: `ssml` | `text`

Wajib: Tidak

VoiceId

ID suara untuk digunakan untuk sintesis. Anda bisa mendapatkan daftar suara yang tersedia IDs dengan memanggil [DescribeVoices](#) operasi.

Tipe: String

Nilai yang Valid: Aditi | Amy | Astrid | Bianca | Brian | Camila | Carla | Carmen | Celine | Chantal | Conchita | Cristiano | Dora | Emma | Enrique | Ewa | Filiz | Gabrielle | Geraint | Giorgio | Gwyneth | Hans | Ines | Ivy | Jacek | Jan | Joanna | Joey | Justin | Karl | Kendra | Kevin | Kimberly | Lea | Liv | Lotte | Lucia | Lupe | Mads | Maja | Marlene | Mathieu | Matthew | Maxim | Mia | Miguel | Mizuki | Naja | Nicole | Olivia | Penelope | Raveena | Ricardo | Ruben | Russell | Salli | Seoyeon | Takumi | Tatyana | Vicki | Vitoria | Zeina | Zhiyu | Aria | Ayanda | Arlet | Hannah | Arthur | Daniel | Liam | Pedro | Kajal | Hiujin | Laura | Elin | Ida | Suvi | Ola | Hala | Andres | Sergio | Remi | Adriano | Thiago | Ruth | Stephen | Kazuha | Tomoko | Niamh | Sofie | Lisa | Isabelle | Zayd | Danielle | Gregory | Burcu | Jitka | Sabrina | Jasmine | Jihye | Ambre | Beatrice | Florian | Lennart | Lorenzo | Tiffany

Wajib: Ya

Sintaksis Respons

```
HTTP/1.1 200
Content-Type: ContentType
x-amzn-RequestCharacters: RequestCharacters

AudioStream
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Respons mengembalikan header HTTP berikut.

ContentType

Menentukan jenis aliran audio. Ini harus mencerminkan `OutputFormat` parameter dalam permintaan Anda.

- Jika Anda meminta `mp3` sebagai `OutputFormat`, yang `ContentType` dikembalikan adalah `audio/mpeg`.
- Jika Anda meminta `ogg_vorbis` sebagai `OutputFormat`, yang `ContentType` dikembalikan adalah `audio/ogg`.
- Jika Anda meminta `ogg_opus` sebagai `OutputFormat`, yang `ContentType` dikembalikan adalah `audio/ogg`.
- Jika Anda meminta `pcm` sebagai `OutputFormat`, yang `ContentType` dikembalikan adalah `audio/pcm` dalam format 16-bit, 1 saluran (mono), endian kecil yang ditandatangani.
- Jika Anda meminta `mu-law` sebagai `OutputFormat`, yang `ContentType` dikembalikan adalah `audio/mulaw`.
- Jika Anda meminta `a-law` sebagai `OutputFormat`, yang `ContentType` dikembalikan adalah `audio/alaw`.
- Jika Anda meminta `json` sebagai `OutputFormat`, yang `ContentType` dikembalikan adalah `aplikasi/x-json-stream`.

RequestCharacters

Jumlah karakter yang disintesis.

Respons mengembalikan yang berikut sebagai isi HTTP.

AudioStream

Aliran yang berisi pidato yang disintesis.

Kesalahan

EngineNotSupportedException

Mesin ini tidak kompatibel dengan suara yang Anda tunjuk. Pilih suara baru yang kompatibel dengan mesin atau ganti mesin dan nyalakan kembali operasi.

Kode Status HTTP: 400

InvalidSampleRateException

Tingkat sampel yang ditentukan tidak valid.

Kode Status HTTP: 400

InvalidSsmlException

SSML yang Anda berikan tidak valid. Verifikasi sintaks SSML, ejaan tag dan nilai, lalu coba lagi.

Kode Status HTTP: 400

LanguageNotSupportedException

Bahasa yang ditentukan saat ini tidak didukung oleh Amazon Polly dalam kapasitas ini.

Kode Status HTTP: 400

LexiconNotFoundException

Amazon Polly tidak dapat menemukan leksikon yang ditentukan. Ini bisa disebabkan oleh leksikon yang hilang, namanya salah eja atau menentukan leksikon yang ada di wilayah berbeda.

Verifikasi bahwa leksikon ada, ada di wilayah (lihat [ListLexicons](#)) dan Anda mengeja namanya dieja dengan benar. Kemudian coba lagi.

Kode Status HTTP: 404

MarksNotSupportedForFormatException

Tanda ucapan tidak didukung untuk yang OutputFormat dipilih. Tanda ucapan hanya tersedia untuk konten dalam json format.

Kode Status HTTP: 400

ServiceFailureException

Kondisi yang tidak diketahui telah menyebabkan kegagalan layanan.

Kode Status HTTP: 500

SsmlMarksNotSupportedForTextTypeException

Tanda ucapan SSML tidak didukung untuk input tipe teks biasa.

Kode Status HTTP: 400

TextLengthExceededException

Nilai parameter “Teks” lebih panjang dari batas yang diterima. Untuk SynthesizeSpeech API, batas untuk teks input adalah maksimum 6000 karakter total, yang tidak lebih dari 3000 karakter dapat ditagih. Untuk StartSpeechSynthesisTask API, maksimum adalah 200.000 karakter, yang tidak lebih dari 100.000 karakter dapat ditagih. Tag SSML tidak dihitung sebagai karakter yang ditagih.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS Antarmuka Baris Perintah V2](#)
- [AWS SDK for .NET V4](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go v2](#)
- [AWS SDK for Java V2](#)
- [AWS SDK untuk V3 JavaScript](#)
- [AWS SDK para Kotlin](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

Tipe Data

tipe data berikut didukung:

- [AudioEvent](#)
- [CloseStreamEvent](#)
- [FlushStreamConfiguration](#)
- [Lexicon](#)
- [LexiconAttributes](#)
- [LexiconDescription](#)
- [StartSpeechSynthesisStreamActionStream](#)
- [StartSpeechSynthesisStreamEventStream](#)
- [StreamClosedEvent](#)
- [SynthesisTask](#)
- [TextEvent](#)
- [ThrottlingReason](#)
- [ValidationExceptionField](#)
- [Voice](#)

AudioEvent

Berisi sepotong data audio yang disintesis.

Daftar Isi

AudioChunk

Sepotong data audio yang disintesis dikodekan dalam format yang ditentukan oleh parameter.

OutputFormat

Tipe: Objek data biner berkode Base64

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

CloseStreamEvent

Menunjukkan akhir aliran input. Setelah mengirim acara ini, aliran input akan ditutup dan semua audio akan dikembalikan.

Daftar Isi

Anggota struktur pengecualian ini bergantung pada konteks.

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

FlushStreamConfiguration

Konfigurasi yang mengontrol saat data audio yang disintesis dikirim pada aliran output.

Daftar Isi

Force

Menentukan apakah akan memaksa mesin sintesis untuk segera menulis data audio buffer ke aliran output.

Tipe: Boolean

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

Lexicon

Menyediakan nama leksikon dan konten leksikon dalam format string. Untuk informasi selengkapnya, lihat [Spesifikasi Leksikon Pengucapan \(PLS\) Versi 1.0](#).

Daftar Isi

Content

Konten leksikon dalam format string. Isi leksikon harus dalam format PLS.

Tipe: String

Wajib: Tidak

Name

Nama leksikon.

Tipe: String

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

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

LexiconAttributes

Berisi metadata yang menggambarkan leksikon seperti jumlah leksem, kode bahasa, dan sebagainya. Untuk informasi selengkapnya, lihat [Mengelola Leksikon](#).

Daftar Isi

Alphabet

Alfabet fonetik yang digunakan dalam leksikon. Nilai yang valid adalah ipa dan x-sampa.

Tipe: String

Wajib: Tidak

LanguageCode

Kode bahasa yang berlaku untuk leksikon. Sebuah leksikon dengan kode bahasa seperti “en” akan diterapkan ke semua bahasa Inggris (en-GB, en-US, en-AUS, en-WLS, dan sebagainya).

Tipe: String

Nilai yang Valid: arb | cmn-CN | cy-GB | da-DK | de-DE | en-AU | en-GB | en-GB-WLS | en-IN | en-US | es-ES | es-MX | es-US | fr-CA | fr-FR | is-IS | it-IT | ja-JP | hi-IN | ko-KR | nb-NO | nl-NL | pl-PL | pt-BR | pt-PT | ro-RO | ru-RU | sv-SE | tr-TR | en-NZ | en-ZA | ca-ES | de-AT | yue-CN | ar-AE | fi-FI | en-IE | nl-BE | fr-BE | cs-CZ | de-CH | en-SG

Wajib: Tidak

LastModified

Leksikon tanggal terakhir diubah (nilai stempel waktu).

Tipe: Timestamp

Wajib: Tidak

LexemesCount

Jumlah leksem dalam leksikon.

Tipe: Integer

Wajib: Tidak

LexiconArn

Nama Sumber Daya Amazon (ARN) dari leksikon.

Tipe: String

Wajib: Tidak

Size

Ukuran total leksikon, dalam karakter.

Tipe: Integer

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

LexiconDescription

Menjelaskan isi leksikon.

Daftar Isi

Attributes

Menyediakan metadata leksikon.

Tipe: Objek [LexiconAttributes](#)

Wajib: Tidak

Name

Nama leksikon.

Tipe: String

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

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

StartSpeechSynthesisStreamActionStream

Aliran acara masuk untuk mengirim input dan acara kontrol untuk mengelola sintesis ucapan dua arah.

Daftar Isi

CloseStreamEvent

Peristiwa yang menunjukkan akhir aliran input.

Tipe: Objek [CloseStreamEvent](#)

Wajib: Tidak

TextEvent

Sebuah peristiwa teks yang berisi konten yang akan disintesis.

Tipe: Objek [TextEvent](#)

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

StartSpeechSynthesisStreamEventStream

Aliran peristiwa keluar yang berisi data audio yang disintesis dan peristiwa status aliran.

Daftar Isi

AudioEvent

Acara audio yang berisi pidato yang disintesis.

Tipe: Objek [AudioEvent](#)

Wajib: Tidak

ServiceFailureException

Kondisi yang tidak diketahui telah menyebabkan kegagalan layanan.

Jenis: Pengecualian

Kode Status HTTP: 500

Wajib: Tidak

ServiceQuotaExceededException

Pengecualian yang menunjukkan kuota layanan akan terlampaui.

Jenis: Pengecualian

Kode Status HTTP: 402

Wajib: Tidak

StreamClosedEvent

Sebuah peristiwa, dengan informasi ringkasan, yang menunjukkan aliran telah ditutup.

Tipe: Objek [StreamClosedEvent](#)

Wajib: Tidak

ThrottlingException

Pengecualian yang menunjukkan permintaan dibatasi.

Jenis: Pengecualian

Kode Status HTTP: 400

Wajib: Tidak

ValidationException

Pengecualian yang menunjukkan validasi input gagal.

Jenis: Pengecualian

Kode Status HTTP: 400

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

StreamClosedEvent

Menunjukkan bahwa aliran sintesis ditutup dan memberikan informasi ringkasan.

Daftar Isi

RequestCharacters

Jumlah total karakter yang disintesis selama sesi streaming.

Tipe: Integer

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

SynthesisTask

SynthesisTask objek yang memberikan informasi tentang tugas sintesis ucapan.

Daftar Isi

CreationTime

Stempel waktu untuk saat tugas sintesis dimulai.

Tipe: Timestamp

Wajib: Tidak

Engine

Menentukan mesin (`standard`, `neural`, `long-form` atau `generative`) untuk Amazon Polly untuk digunakan saat memproses teks input untuk sintesis ucapan. Menggunakan suara yang tidak didukung untuk mesin yang dipilih akan mengakibatkan kesalahan.

Tipe: String

Nilai yang Valid: `standard` | `neural` | `long-form` | `generative`

Wajib: Tidak

LanguageCode

Kode bahasa opsional untuk tugas sintesis. Ini hanya diperlukan jika menggunakan suara bilingual, seperti Aditi, yang dapat digunakan untuk bahasa Inggris India (`En-in`) atau Hindi (`Hi-in`).

Jika suara dwibahasa digunakan dan tidak ada kode bahasa yang ditentukan, Amazon Polly menggunakan bahasa default suara dwibahasa. Bahasa default untuk suara apa pun adalah yang dikembalikan oleh [DescribeVoices](#) operasi untuk `LanguageCode` parameter. Misalnya, jika tidak ada kode bahasa yang ditentukan, Aditi akan menggunakan bahasa Inggris India daripada bahasa Hindi.

Tipe: String

Nilai yang Valid: `arb` | `cmn-CN` | `cy-GB` | `da-DK` | `de-DE` | `en-AU` | `en-GB` | `en-GB-WLS` | `en-IN` | `en-US` | `es-ES` | `es-MX` | `es-US` | `fr-CA` | `fr-FR` | `is-IS` | `it-IT` | `ja-JP` | `hi-IN` | `ko-KR` | `nb-NO` | `nl-NL` | `pl-PL` | `pt-BR` | `pt-PT` |

ro-RO | ru-RU | sv-SE | tr-TR | en-NZ | en-ZA | ca-ES | de-AT | yue-CN |
ar-AE | fi-FI | en-IE | nl-BE | fr-BE | cs-CZ | de-CH | en-SG

Wajib: Tidak

LexiconNames

Daftar satu atau lebih nama leksikon pengucapan yang Anda inginkan untuk diterapkan layanan selama sintesis. Leksikon diterapkan hanya jika bahasa leksikon sama dengan bahasa suara.

Tipe: Array string

Anggota Array: Jumlah maksimum 5 item.

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

Wajib: Tidak

OutputFormat

Format di mana output yang dikembalikan akan dikodekan. Untuk streaming audio, ini akan menjadi mp3, ogg_vorbis, ogg_opus, mu-law, a-law, atau pcm. Untuk tanda bicara, ini akan menjadi json.

Tipe: String

Nilai yang Valid: json | mp3 | ogg_opus | ogg_vorbis | pcm | mulaw | alaw

Wajib: Tidak

OutputUri

Jalur untuk file ucapan keluaran.

Tipe: String

Wajib: Tidak

RequestCharacters

Jumlah karakter yang dapat ditagih disintesis.

Tipe: Integer

Wajib: Tidak

SampleRate

Frekuensi audio yang ditentukan dalam Hz.

Nilai yang valid untuk mp3 dan ogg_vorbis adalah "8000", "16000", "22050", dan "24000". Nilai default untuk suara standar adalah "22050". Nilai default untuk suara saraf adalah "24000". Nilai default untuk suara bentuk panjang adalah "24000". Nilai default untuk suara generatif adalah "24000".

Nilai yang valid untuk pcm adalah "8000" dan "16000" Nilai default adalah "16000".

Nilai valid untuk ogg_opus adalah "48000".

Nilai yang valid untuk mu-law dan a-law adalah "8000".

Tipe: String

Wajib: Tidak

SnsTopicArn

ARN untuk topik SNS secara opsional digunakan untuk memberikan pemberitahuan status untuk tugas sintesis ucapan.

Tipe: String

Pola: `^arn:aws(-(cn|iso(-b)?|us-gov))?:sns:[a-z0-9_-]{1,50}:\d{12}:[a-zA-Z0-9_-]{1,251}([a-zA-Z0-9_-]{0,5}|\.fifo)$`

Wajib: Tidak

SpeechMarkTypes

Jenis tanda bicara yang dikembalikan untuk teks input.

Tipe: Array string

Anggota Array: Jumlah maksimum 4 item.

Nilai yang Valid: `sentence | ssm1 | viseme | word`

Wajib: Tidak

TaskId

Amazon Polly menghasilkan pengidentifikasi untuk tugas sintesis ucapan.

Tipe: String

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

Wajib: Tidak

TaskStatus

Status saat ini dari tugas sintesis ucapan individu.

Tipe: String

Nilai yang Valid: `scheduled` | `inProgress` | `completed` | `failed`

Wajib: Tidak

TaskStatusReason

Alasan status saat ini dari tugas sintesis ucapan tertentu, termasuk kesalahan jika tugas gagal.

Tipe: String

Wajib: Tidak

TextType

Menentukan apakah teks input adalah teks biasa atau SSML. Nilai defaultnya adalah teks biasa.

Tipe: String

Nilai yang Valid: `ssml` | `text`

Wajib: Tidak

Voiceld

ID suara untuk digunakan untuk sintesis.

Tipe: String

Nilai yang Valid: `Aditi` | `Amy` | `Astrid` | `Bianca` | `Brian` | `Camila` | `Carla` | `Carmen` | `Celine` | `Chantal` | `Conchita` | `Cristiano` | `Dora` | `Emma` | `Enrique` | `Ewa` | `Filiz` | `Gabrielle` | `Geraint` | `Giorgio` | `Gwyneth` | `Hans` | `Ines` | `Ivy` | `Jacek` | `Jan` | `Joanna` | `Joey` | `Justin` | `Karl` | `Kendra` | `Kevin` | `Kimberly` | `Lea` | `Liv` | `Lotte` | `Lucia` | `Lupe` | `Mads` | `Maja` | `Marlene`

| Mathieu | Matthew | Maxim | Mia | Miguel | Mizuki | Naja | Nicole
| Olivia | Penelope | Raveena | Ricardo | Ruben | Russell | Salli |
Seoyeon | Takumi | Tatyana | Vicki | Vitoria | Zeina | Zhiyu | Aria
| Ayanda | Arlet | Hannah | Arthur | Daniel | Liam | Pedro | Kajal |
Hiujin | Laura | Elin | Ida | Suvi | Ola | Hala | Andres | Sergio | Remi
| Adriano | Thiago | Ruth | Stephen | Kazuha | Tomoko | Niamh | Sofie
| Lisa | Isabelle | Zayd | Danielle | Gregory | Burcu | Jitka | Sabrina
| Jasmine | Jihye | Ambre | Beatrice | Florian | Lennart | Lorenzo |
Tiffany

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

TextEvent

Berisi konten teks untuk disintesis menjadi ucapan.

Daftar Isi

Text

Konten teks untuk disintesis. Jika Anda menentukan `ssml` sebagai `TextType`, ikuti format SSML untuk teks input.

Tipe: String

Diperlukan: Ya

FlushStreamConfiguration

Konfigurasi untuk mengontrol saat audio yang disintesis disiram ke aliran output.

Tipe: Objek [FlushStreamConfiguration](#)

Wajib: Tidak

TextType

Menentukan apakah teks input adalah teks biasa atau SSML. Default: teks biasa.

Tipe: String

Nilai yang Valid: `ssml` | `text`

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ThrottlingReason

Memberikan informasi tentang alasan pelambatan tertentu.

Daftar Isi

reason

Kode alasan menjelaskan mengapa permintaan itu dibatasi.

Tipe: String

Wajib: Tidak

resource

Sumber daya yang menyebabkan pelambatan.

Tipe: String

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ValidationExceptionField

Informasi tentang bidang yang gagal validasi.

Daftar Isi

message

Pesan yang menjelaskan mengapa bidang gagal validasi.

Tipe: String

Diperlukan: Ya

name

Nama bidang yang gagal validasi.

Tipe: String

Wajib: Ya

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

Voice

Deskripsi suara.

Daftar Isi

AdditionalLanguageCodes

Kode tambahan untuk bahasa yang tersedia untuk suara yang ditentukan selain bahasa defaultnya.

Misalnya, bahasa default untuk Aditi adalah bahasa Inggris India (en-in) karena pertama kali digunakan untuk bahasa itu. Karena Aditi bilingual dan fasih dalam bahasa Inggris India dan Hindi, parameter ini akan menunjukkan kode. hi-IN

Tipe: Array string

Nilai yang Valid: arb | cmn-CN | cy-GB | da-DK | de-DE | en-AU | en-GB | en-GB-WLS | en-IN | en-US | es-ES | es-MX | es-US | fr-CA | fr-FR | is-IS | it-IT | ja-JP | hi-IN | ko-KR | nb-NO | nl-NL | pl-PL | pt-BR | pt-PT | ro-RO | ru-RU | sv-SE | tr-TR | en-NZ | en-ZA | ca-ES | de-AT | yue-CN | ar-AE | fi-FI | en-IE | nl-BE | fr-BE | cs-CZ | de-CH | en-SG

Wajib: Tidak

Gender

Jenis kelamin suara.

Tipe: String

Nilai yang Valid: Female | Male

Wajib: Tidak

Id

Amazon Polly menetapkan ID suara. Ini adalah ID yang Anda tentukan saat memanggil SynthesizeSpeech operasi.

Tipe: String

Nilai yang Valid: Aditi | Amy | Astrid | Bianca | Brian | Camila | Carla | Carmen | Celine | Chantal | Conchita | Cristiano | Dora | Emma | Enrique

| Ewa | Filiz | Gabrielle | Geraint | Giorgio | Gwyneth | Hans | Ines
 | Ivy | Jacek | Jan | Joanna | Joey | Justin | Karl | Kendra | Kevin
 | Kimberly | Lea | Liv | Lotte | Lucia | Lupe | Mads | Maja | Marlene
 | Mathieu | Matthew | Maxim | Mia | Miguel | Mizuki | Naja | Nicole
 | Olivia | Penelope | Raveena | Ricardo | Ruben | Russell | Salli |
 Seoyeon | Takumi | Tatyana | Vicki | Vitoria | Zeina | Zhiyu | Aria
 | Ayanda | Arlet | Hannah | Arthur | Daniel | Liam | Pedro | Kajal |
 Hiujin | Laura | Elin | Ida | Suvi | Ola | Hala | Andres | Sergio | Remi
 | Adriano | Thiago | Ruth | Stephen | Kazuha | Tomoko | Niamh | Sofie
 | Lisa | Isabelle | Zayd | Danielle | Gregory | Burcu | Jitka | Sabrina
 | Jasmine | Jihye | Ambre | Beatrice | Florian | Lennart | Lorenzo |
 Tiffany

Wajib: Tidak

LanguageCode

Kode bahasa suara.

Tipe: String

Nilai yang Valid: arb | cmn-CN | cy-GB | da-DK | de-DE | en-AU | en-GB | en-GB-WLS | en-IN | en-US | es-ES | es-MX | es-US | fr-CA | fr-FR | is-IS | it-IT | ja-JP | hi-IN | ko-KR | nb-NO | nl-NL | pl-PL | pt-BR | pt-PT | ro-RO | ru-RU | sv-SE | tr-TR | en-NZ | en-ZA | ca-ES | de-AT | yue-CN | ar-AE | fi-FI | en-IE | nl-BE | fr-BE | cs-CZ | de-CH | en-SG

Wajib: Tidak

LanguageName

Nama bahasa yang dapat dibaca manusia dalam bahasa Inggris.

Tipe: String

Wajib: Tidak

Name

Nama suara (misalnya, Salli, Kendra, dll.). Ini memberikan nama suara yang dapat dibaca manusia yang mungkin Anda tampilkan di aplikasi Anda.

Tipe: String

Wajib: Tidak

SupportedEngines

Menentukan mesin mana (standard,neural, long-form atau generative) yang didukung oleh suara yang diberikan.

Tipe: Array string

Nilai yang Valid: standard | neural | long-form | generative

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu bahasa khusus AWS SDKs, lihat berikut ini:

- [AWS SDK for C++](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

Jenis Kesalahan Umum

Bagian ini mencantumkan jenis kesalahan umum yang mungkin dikembalikan oleh AWS layanan ini. Tidak semua layanan mengembalikan semua jenis kesalahan yang tercantum di sini. Untuk kesalahan khusus pada tindakan API untuk layanan ini, lihat topik untuk tindakan API tersebut.

AccessDeniedException

Anda tidak memiliki izin untuk melakukan tindakan ini. Verifikasi bahwa kebijakan IAM Anda menyertakan izin yang diperlukan.

Kode Status HTTP: 403

ExpiredTokenException

Token keamanan yang termasuk dalam permintaan telah kedaluwarsa. Minta token keamanan baru dan coba lagi.

Kode Status HTTP: 403

IncompleteSignature

Tanda tangan permintaan tidak sesuai dengan AWS standar. Verifikasi bahwa Anda menggunakan AWS kredensi yang valid dan permintaan Anda diformat dengan benar. Jika Anda menggunakan SDK, pastikan itu mutakhir.

Kode Status HTTP: 403

InternalFailure

Permintaan tidak dapat diproses sekarang karena masalah server internal. Coba lagi nanti. Jika masalah berlanjut, hubungi AWS Support.

Kode Status HTTP: 500

MalformedHttpRequestException

Badan permintaan tidak dapat diproses. Ini biasanya terjadi ketika badan permintaan tidak dapat didekompresi menggunakan algoritme pengkodean konten yang ditentukan. Verifikasi bahwa header pengkodean konten cocok dengan format kompresi yang digunakan.

Kode Status HTTP: 400

NotAuthorized

Anda tidak memiliki izin untuk melakukan tindakan ini. Verifikasi bahwa kebijakan IAM Anda menyertakan izin yang diperlukan.

Kode Status HTTP: 401

OptInRequired

AWS Akun Anda memerlukan langganan untuk layanan ini. Verifikasi bahwa Anda telah mengaktifkan layanan di akun Anda.

Kode Status HTTP: 403

RequestAbortedException

Permintaan dibatalkan sebelum tanggapan dapat dikembalikan. Ini biasanya terjadi ketika klien menutup koneksi.

Kode Status HTTP: 400

RequestEntityTooLargeException

Entitas permintaan terlalu besar. Kurangi ukuran badan permintaan dan coba lagi.

Kode Status HTTP: 413

RequestTimeoutException

Permintaan habis waktunya. Server tidak menerima permintaan lengkap dalam jangka waktu yang diharapkan. Coba lagi.

Kode Status HTTP: 408

ServiceUnavailable

Layanan untuk sementara tidak tersedia. Coba lagi nanti.

Kode Status HTTP: 503

ThrottlingException

Tingkat permintaan Anda terlalu tinggi. Permintaan coba ulang AWS SDKs secara otomatis yang menerima pengecualian ini. Kurangi frekuensi permintaan.

Kode Status HTTP: 400

UnknownOperationException

Tindakan atau operasi tidak dikenali. Verifikasi bahwa nama tindakan dieja dengan benar dan didukung oleh versi API yang Anda gunakan.

Kode Status HTTP: 404

UnrecognizedClientException

Sertifikat X.509 atau ID kunci AWS akses yang Anda berikan tidak ada dalam catatan kami. Verifikasi bahwa Anda menggunakan kredensi yang valid dan belum kedaluwarsa.

Kode Status HTTP: 403

ValidationError

Input tidak memenuhi format atau kendala yang diperlukan. Periksa apakah semua parameter yang diperlukan disertakan dan nilainya valid.

Kode Status HTTP: 400

Parameter Umum

Daftar berikut berisi parameter yang digunakan semua tindakan untuk menandatangani permintaan Tanda Tangan Versi 4 dengan string kueri. Setiap parameter khusus tindakan tercantum dalam topik untuk tindakan tersebut. Untuk informasi selengkapnya tentang Tanda Tangan Versi 4, lihat [Menandatangani permintaan AWS API](#) di Panduan Pengguna IAM.

X-Amz-Algorithm

Algoritme hash yang Anda gunakan untuk membuat tanda tangan permintaan.

Syarat: Tentukan parameter ini ketika Anda menyertakan informasi autentikasi dalam string kueri alih-alih di header otorisasi HTTP.

Tipe: string

Nilai yang Valid: AWS4-HMAC-SHA256

Diperlukan: Kondisional

X-Amz-Credential

Nilai lingkup kredensial, yang merupakan string yang menyertakan access key Anda, tanggal, wilayah yang Anda targetkan, layanan yang Anda minta, dan string penghentian ("aws4_request"). Nilai dinyatakan dalam format berikut: access_key/HHBBTTTT/wilayah/layanan/aws4_request.

Untuk informasi selengkapnya, lihat [Membuat permintaan AWS API yang ditandatangani](#) di Panduan Pengguna IAM.

Syarat: Tentukan parameter ini ketika Anda menyertakan informasi autentikasi dalam string kueri alih-alih di header otorisasi HTTP.

Tipe: string

Wajib: Bersyarat

X-Amz-Date

Tanggal yang digunakan untuk membuat tanda tangan. Format harus berupa format dasar ISO 8601 (YYYYMMDD'T'HMMSS'Z'). Misalnya, waktu tanggal berikut adalah X-Amz-Date nilai yang valid:20120325T120000Z.

Kondisi: X-Amz-Date bersifat opsional untuk semua permintaan; dapat digunakan untuk mengganti tanggal yang digunakan untuk menandatangani permintaan. Jika header Tanggal

ditentukan dalam format dasar ISO 8601, tidak X-Amz-Date diperlukan. Ketika X-Amz-Date digunakan, itu selalu mengesampingkan nilai header Tanggal. Untuk informasi selengkapnya, lihat [Elemen tanda tangan permintaan AWS API](#) di Panduan Pengguna IAM.

Jenis: string

Wajib: Bersyarat

X-Amz-Security-Token

Token keamanan sementara yang diperoleh melalui panggilan ke AWS Security Token Service (AWS STS). Untuk daftar layanan yang mendukung kredensi keamanan sementara AWS STS, lihat layanan [Layanan AWS yang berfungsi dengan IAM di Panduan Pengguna IAM](#).

Kondisi: Jika Anda menggunakan kredensi keamanan sementara dari AWS STS, Anda harus menyertakan token keamanan.

Jenis: string

Wajib: Bersyarat

X-Amz-Signature

Menentukan tanda tangan yang dikodekan oleh hex yang dihitung dari string to sign dan kunci penandatanganan turunan.

Syarat: Tentukan parameter ini ketika Anda menyertakan informasi autentikasi dalam string kueri alih-alih di header otorisasi HTTP.

Tipe: string

Wajib: Bersyarat

X-Amz-SignedHeaders

Menentukan semua header HTTP yang disertakan sebagai bagian dari permintaan kanonik. Untuk informasi selengkapnya tentang menentukan header yang ditandatangani, lihat [Membuat permintaan AWS API yang ditandatangani](#) di Panduan Pengguna IAM.

Syarat: Tentukan parameter ini ketika Anda menyertakan informasi autentikasi dalam string kueri alih-alih di header otorisasi HTTP.

Tipe: string

Diperlukan: Kondisional

Sejarah Dokumen untuk Amazon Polly

Tabel berikut menjelaskan perubahan penting dalam setiap rilis Panduan Pengembang Amazon Polly. Untuk notifikasi tentang pembaruan dokumentasi ini, Anda dapat berlangganan ke umpan RSS.

- Pembaruan dokumentasi terbaru: 28 Mei 2026

Perubahan	Deskripsi	Tanggal
Wilayah baru ditambahkan untuk suara saraf	Amazon Polly sekarang tersedia di Wilayah Asia Pasifik (Thailand) AWS . Wilayah ini mendukung suara neural TTS (NTTS). Untuk informasi lebih lanjut, lihat Suara saraf .	28 Mei 2026
Wilayah baru ditambahkan untuk suara generatif	Suara generatif Amazon Polly dan streaming dua arah sekarang tersedia di AWS Wilayah tambahan: Eropa (Zurich). Untuk informasi lebih lanjut, lihat Suara generatif .	28 Mei 2026
Streaming dua arah tersedia di wilayah baru	Streaming dua arah Amazon Polly sekarang tersedia di dua AWS Wilayah tambahan: Eropa (London) dan Kanada (Tengah). Untuk informasi lebih lanjut, lihat Suara generatif .	April 20, 2026
Kebijakan terkelola yang diperbarui	Amazon Polly telah memperbarui kebijakan terkelola AmazonPollyReadOnlyAccess untuk menyertakan izin untuk	Maret 19, 2026

sintesis ucapan streaming dua arah. Untuk selengkapnya, lihat [Amazon Polly memperbarui kebijakan AWS terkelola](#).

[Suara baru ditambahkan untuk text-to-speech generatif](#)

Amazon Polly sekarang menyediakan sepuluh suara generatif tambahan: Brian, Aria, Jasmine, Tiffany, Ambre, Florian, Sabrina, Lennart, Beatrice, Lorenzo. Lihat Suara [generatif untuk daftar suara](#) TTS generatif.

Maret 19, 2026

[Wilayah baru ditambahkan untuk suara generatif](#)

Suara generatif Amazon Polly sekarang tersedia di dua AWS Wilayah tambahan: Eropa (London), dan Kanada (Tengah). Untuk informasi lebih lanjut, lihat [Suara generatif](#).

Maret 19, 2026

[Wilayah baru ditambahkan untuk suara generatif](#)

Suara generatif Amazon Polly sekarang tersedia di tiga AWS Wilayah tambahan: Asia Pasifik (Tokyo), Asia Pasifik (Seoul), dan Asia Pasifik (Singapura). Untuk informasi lebih lanjut, lihat [Suara generatif](#).

November 18, 2025

Suara baru ditambahkan untuk text-to-speech generatif	Amazon Polly sekarang menyediakan enam suara generatif tambahan: Seoyeon, Camila, Hannah, Niamh, Laura, dan Lisa. Lihat Suara generatif untuk daftar suara TTS generatif.	November 14, 2025
Wilayah baru ditambahkan untuk suara saraf	Amazon Polly sekarang tersedia di Wilayah Eropa (Zurich). AWS Wilayah ini mendukung suara neural TTS (NTTS). Untuk informasi lebih lanjut, lihat Suara saraf .	Oktober 20, 2025
Suara baru ditambahkan untuk text-to-speech generatif	Amazon Polly sekarang menyediakan tujuh suara generatif tambahan: Salli, Isabelle, Céline, Liam, Gabrielle, Ola, dan Ewa. Lihat Suara generatif untuk daftar suara TTS generatif.	Agustus 26, 2025
Wilayah baru ditambahkan untuk suara saraf dan standar	Amazon Polly sekarang tersedia di Wilayah Asia Pasifik (Malaysia) AWS . Wilayah ini mendukung neural TTS (NTTS) dan suara standar. Untuk informasi lebih lanjut, lihat Suara saraf dan Suara standar .	Maret 27, 2025
Suara baru ditambahkan untuk neural text-to-speech	Amazon Polly sekarang menyediakan satu suara Korea tambahan: Jihye. Lihat Suara saraf untuk daftar suara NTTS.	Maret 26, 2025

[Wilayah baru ditambahkan untuk suara saraf dan standar](#)

Amazon Polly sekarang tersedia di Wilayah Eropa (Spanyol). AWS Wilayah ini mendukung neural TTS (NTTS) dan suara standar. Untuk informasi lebih lanjut, lihat [Suara saraf dan Suara standar](#).

Februari 18, 2025

[Suara baru ditambahkan untuk neural text-to-speech](#)

Amazon Polly sekarang menyediakan satu suara bahasa Inggris (Singapura) tambahan: Jasmine. Lihat [Suara saraf](#) untuk daftar suara NTTS.

Februari 11, 2025

[Suara baru ditambahkan untuk text-to-speech generatif](#)

Amazon Polly sekarang menyediakan tujuh suara generatif baru: Pedro, Andrés, Sergio, Daniel, Kajal, Rémi, Bianca. Lihat Suara [generatif untuk daftar suara](#) TTS generatif.

November 21, 2024

[Suara bentuk panjang baru ditambahkan](#)

Amazon Polly sekarang menyediakan lebih banyak suara bentuk panjang. Satu tambahan bahasa Inggris (AS) dan dua suara bentuk panjang Spanyol (Spanyol) baru ditambahkan: Patrick, Alba, dan Raúl. Lihat [Long-formsuara](#) untuk daftar semua suara bentuk panjang.

November 14, 2024

Suara baru ditambahkan untuk text-to-speech generatif	Amazon Polly sekarang menyediakan enam suara generatif baru: Ayanda, Léa, Lucia, Mía, Lupe, dan Vicki. Lihat Suara generatif untuk daftar suara TTS generatif.	6 November 2024
Suara baru ditambahkan untuk text-to-speech generatif	Amazon Polly sekarang menyediakan empat suara generatif baru: Olivia, Danielle, Joanna, dan Stephen. Lihat Suara generatif untuk daftar suara TTS generatif.	Oktober 10, 2024
Bahasa baru ditambahkan untuk text-to-speech	Amazon Polly sekarang menyediakan dua bahasa TTS baru: Ceko (CS-Cz) dan Jerman (Swiss) (De-CH). Lihat Bahasa yang didukung untuk daftar bahasa.	September 26, 2024
Suara baru ditambahkan untuk NTTS	Amazon Polly sekarang menyediakan dua suara NTTS baru: Jitka dan Sabrina. Lihat Suara saraf untuk daftar suara NTTS.	Agustus 27, 2024
Mesin suara generatif baru ditambahkan	Amazon Polly sekarang menawarkan mesin suara generatif yang dirancang untuk konten yang lebih panjang, dengan tiga suara bahasa Inggris dalam varian generatif : Amy, Matthew, dan Ruth. Lihat Suara generatif untuk informasi lebih lanjut.	Maret 28, 2024

Suara baru ditambahkan untuk NTTS	Amazon Polly sekarang menyediakan suara Turki NTTS Burcu. Lihat Suara saraf untuk daftar suara NTTS.	Februari 14, 2024
Mesin suara bentuk panjang baru ditambahkan	Amazon Polly sekarang menawarkan mesin suara bentuk panjang yang dirancang untuk konten yang lebih panjang, dengan tiga suara en-AS: Danielle, Gregory, dan Ruth. Lihat Long-formsuara untuk informasi lebih lanjut.	16 November 2023
Suara baru ditambahkan untuk NTTS	Amazon Polly sekarang menyediakan dua suara bahasa Inggris NTTS AS baru: Danielle dan Gregory. Lihat Suara saraf untuk daftar suara NTTS.	5 Oktober 2023
Amazon Polly untuk Windows	Plugin Amazon Polly Windows Speech Application Programming Interface (SAPI) tidak akan lagi didukung.	26 September 2023
Panduan kuota yang diperbarui untuk Amazon Polly	Panduan kuota Amazon Polly yang diperbarui. Menambahkan contoh dan klarifikasi istilah. Lihat Kuota di Amazon Polly untuk pembaruan.	17 Agustus 2023
Suara baru ditambahkan untuk NTTS	Amazon Polly sekarang menyediakan suara NTTS Arab Teluk Zayd. Lihat Suara saraf untuk daftar suara NTTS.	16 Agustus 2023

Suara baru ditambahkan untuk NTTS	Amazon Polly sekarang menyediakan suara NTTS Prancis Belgia Isabelle. Lihat Suara saraf untuk daftar suara NTTS.	1 Agustus 2023
Suara baru ditambahkan untuk NTTS	Amazon Polly sekarang menyediakan suara Belgian Dutch (Flemish) NTTS Lisa. Lihat Suara saraf untuk daftar suara NTTS.	Juni 7, 2023
Suara baru ditambahkan untuk NTTS	Amazon Polly sekarang menyediakan dua suara NTTS baru: Bahasa Inggris Irlandia (Niamh), dan Denmark (Sofie). Lihat Suara saraf untuk daftar suara NTTS.	30 Mei 2023
Memperbarui panduan IAM untuk Amazon Polly	Memperbarui panduan untuk menyelaraskan dengan praktik terbaik IAM. Untuk informasi lebih lanjut, lihat Praktik terbaik keamanan di IAM .	19 April 2023
WordPress memperbarui	WordPress Plugin Amazon Polly tidak akan lagi didukung.	6 April 2023
Wilayah Baru ditambahkan	Amazon Polly sekarang tersedia di Wilayah Asia Pasifik (Osaka) AWS . Wilayah ini mendukung neural TTS (NTTS). Untuk informasi selengkapnya, lihat Kompatibilitas Fitur dan Wilayah untuk daftar wilayah yang mendukung NTTS.	5 April 2023

Suara baru ditambahkan untuk NTTS	Amazon Polly sekarang menyediakan dua suara NTTS Jepang baru: Kazuha dan Tomoko. Lihat Suara saraf untuk daftar suara NTTS.	7 Februari 2023
Suara baru ditambahkan untuk NTTS	Amazon Polly sekarang menyediakan dua suara NTTS Inggris AS baru: Stephen dan Ruth. Lihat Suara saraf untuk daftar suara NTTS.	31 Januari 2023
Suara baru ditambahkan untuk NTTS	Amazon Polly sekarang menyediakan suara NTTS baru untuk: Portugis Brasil (Thiago), Spanyol Kastilia (Sergio), Prancis (Rémi), Italia (Adriano), dan Spanyol Meksiko (Andrés). Lihat Suara saraf untuk daftar suara NTTS.	Januari 24, 2023
Suara baru ditambahkan untuk NTTS	Amazon Polly sekarang menyediakan suara NTTS untuk bahasa Arab (Hala) dan Polandia (Ola). Lihat Suara saraf untuk daftar suara NTTS.	17 November 2022
AWS PrivateLink Dukungan rilis	Amazon Polly sekarang memberikan AWS PrivateLink dukungan. Lihat Menggunakan Amazon Polly dengan titik akhir VPC untuk mempelajari lebih lanjut.	9 November 2022

Suara dan bahasa baru ditambahkan untuk NTTS	Amazon Polly sekarang menyediakan suara NTTS untuk Finlandia (Suvi), Norwegia (Ida), dan Swedia (Elin). Lihat Suara saraf untuk daftar suara NTTS.	8 November 2022
Suara baru ditambahkan untuk NTTS	Amazon Polly sekarang menyediakan suara NTTS Belanda Laura. Lihat Suara saraf untuk daftar suara NTTS.	2 November 2022
Wilayah Baru ditambahkan	Amazon Polly sekarang tersedia di Wilayah Eropa (Paris) AWS . Wilayah ini mendukung neural TTS (NTTS). Untuk informasi selengkapnya, lihat Kompatibilitas Fitur dan Wilayah untuk daftar wilayah yang mendukung NTTS.	September 22, 2022
Suara dan bahasa baru ditambahkan untuk NTTS	Amazon Polly sekarang menyediakan suara NTTS Kanton Hiujin. Lihat Suara saraf untuk daftar suara NTTS.	September 20, 2022
Wilayah Baru ditambahkan	Amazon Polly sekarang tersedia di Wilayah Asia Pasifik (Mumbai). AWS Wilayah ini mendukung neural TTS (NTTS). Untuk informasi selengkapnya, lihat Kompatibilitas Fitur dan Wilayah untuk daftar wilayah yang mendukung NTTS.	September 1, 2022

Suara baru ditambahkan untuk NTTS	Amazon Polly sekarang menyediakan suara Mandarin Zhiyu sebagai suara NTTS. Lihat Suara saraf untuk daftar suara NTTS.	23 Agustus 2022
Suara baru ditambahkan untuk NTTS	Amazon Polly sekarang menyediakan suara Hindi NTTS Kajal. Lihat Suara saraf untuk daftar suara NTTS.	27 Juli 2022
Suara baru ditambahkan untuk NTTS	Amazon Polly sekarang menyediakan suara NTTS untuk bahasa Spanyol AS (Pedro), Jerman (Daniel), Prancis Kanada (Liam), dan Inggris Inggris (Arthur). Lihat Suara saraf untuk daftar suara NTTS.	28 Juni 2022
Suara baru ditambahkan untuk NTTS	Amazon Polly sekarang menyediakan suara Portugis (Brasil) Vitória sebagai suara NTTS. Lihat Suara saraf untuk daftar suara NTTS.	27 April 2022
Suara baru ditambahkan untuk NTTS	Amazon Polly sekarang menyediakan suara Portugis (Eropa) Inês sebagai suara NTTS. Lihat Suara saraf untuk daftar suara NTTS.	26 April 2022
Suara dan bahasa baru ditambahkan untuk NTTS	Amazon Polly sekarang menyediakan bahasa Jerman (Austria) dan suara NTTS Hannah. Lihat Suara saraf untuk daftar suara NTTS.	19 April 2022

[Suara dan bahasa baru ditambahkan untuk NTTS](#)

Amazon Polly sekarang menyediakan suara Spanyol (Meksiko) Mia sebagai suara NTTS. Bahasa baru, Catalan, ditambahkan bersama dengan Arlet suara NTTS. Lihat [Suara saraf](#) untuk daftar suara NTTS.

22 Maret 2022

[Suara baru ditambahkan untuk NTTS](#)

Amazon Polly sekarang menyediakan suara Jepang Takumi sebagai suara NTTS. Lihat [Suara saraf](#) untuk daftar suara NTTS.

Desember 6, 2021

[Suara baru ditambahkan untuk NTTS](#)

Amazon Polly sekarang menyediakan suara Prancis Léa sebagai suara NTTS. Lihat [Suara saraf](#) untuk daftar suara NTTS.

18 November 2021

[Suara baru ditambahkan untuk NTTS](#)

Amazon Polly sekarang menyediakan suara Italia Bianca dan suara Spanyol Eropa Lucia sebagai suara NTTS. Lihat [Suara saraf](#) untuk daftar suara NTTS.

November 8, 2021

[Suara baru ditambahkan untuk NTTS](#)

Amazon Polly sekarang menyediakan suara bahasa Inggris Afrika Selatan baru, Ayanda. Suara hanya tersedia sebagai suara NTTS. Lihat [Suara saraf](#) untuk daftar suara NTTS.

1 September 2021

Wilayah Baru ditambahkan	Amazon Polly sekarang tersedia di Wilayah Afrika (Cape Town) AWS . Wilayah ini mendukung neural TTS (NTTS). Untuk informasi selengkapnya, lihat Kompatibilitas Fitur dan Wilayah untuk daftar wilayah yang mendukung NTTS.	1 September 2021
Bahasa dan suara baru ditambahkan	Amazon Polly sekarang mendukung Bahasa Inggris Selandia Baru (en-NZ). Suara NTTS baru, Aria, berbicara bahasa Inggris Selandia Baru dan pilihan kata-kata Maori.	Agustus 24, 2021
Fitur baru	Amazon Polly menjadikan gaya berbicara percakapan sebagai versi default untuk suara saraf Matthew dan Joanna. Kami menghapus referensi ke gaya berbicara percakapan.	28 Juni 2021
Suara baru ditambahkan untuk NTTS	Amazon Polly sekarang menyediakan suara Jerman Vicki sebagai suara NTTS.	15 Juni 2021

Suara baru ditambahkan	Suara wanita baru, Gabrielle , telah ditambahkan ke lokal Prancis (Kanada) (FR-CA). Suara berkualitas tinggi dan hanya tersedia sebagai suara NTTS. Seperti semua suara saraf, ini hanya tersedia di wilayah tertentu. Untuk daftar wilayah, lihat Kompatibilitas fitur dan wilayah .	1 Juni 2021
Suara baru ditambahkan untuk NTTS	Amazon Polly sekarang menyediakan suara Korea Seoyeon sebagai suara NTTS.	11 Mei 2021
Wilayah Baru ditambahkan untuk NTTS	Amazon Polly sekarang mendukung neural TTS (NTTS) di Wilayah Kanada (Tengah). AWS Untuk informasi selengkapnya, lihat Kompatibilitas Fitur dan Wilayah untuk NTTS.	17 Maret 2021
Suara baru tersedia untuk gaya penyiar berita	Selain suara Matthew, Joanna, dan Lupe untuk gaya berbicara Newscaster, Amazon Polly sekarang menyediakan opsi tambahan untuk gaya berbicara ini. Menggunakan mesin saraf, Anda dapat menggunakan suara Amy dalam bahasa Inggris Inggris untuk gaya Newscaster. Untuk informasi selengkapnya, lihat Gaya Berbicara NTTS .	10 November 2020

Wilayah Baru ditambahkan untuk NTTS

Selain Wilayah yang ada untuk NTTS (us-east-1, us-west-2, eu-west-1, dan ap-southeast-2), suara saraf sekarang didukung di empat Wilayah tambahan: (ap-north-east-1 (Tokyo), ap-southeast-1 (Singapura), eu-central-1 (Frankfurt), dan eu-west-2 (London)). Untuk informasi selengkapnya, lihat [Kompatibilitas Fitur dan Wilayah](#) untuk NTTS.

3 September 2020

Suara baru ditambahkan

Selain suara anak Ivy dan Justin, suara anak laki-laki baru, Kevin, telah ditambahkan ke bahasa Inggris Amerika (en-US). Suara baru ini berkualitas sangat tinggi dan hanya tersedia sebagai suara NTTS. Seperti semua suara saraf, itu hanya didukung di empat Wilayah: us-east-1 (Virginia N.), us-west-2 (Oregon), eu-west-1 (Irlandia), dan ap-southeast-2 (Sydney). Untuk informasi selengkapnya, lihat Suara [NTTS](#).

16 Juni 2020

Suara baru tersedia untuk gaya penyiar berita

Selain suara Matthew dan Joanna untuk gaya berbicara Newscaster, Amazon Polly sekarang menyediakan opsi tambahan untuk gaya berbicara ini. Menggunakan mesin saraf, Anda dapat menggunakan suara Lupe dalam bahasa Spanyol (Amerika) untuk gaya Newscaster. Untuk informasi selengkapnya, lihat [Gaya Berbicara NTTS](#).

16 April 2020

Fitur baru

Selain gaya berbicara Newscaster, Amazon Polly sekarang menyediakan gaya berbicara NTTS kedua untuk membantu Anda mensintesis teks ke bagian ucapan yang lebih baik. Gaya percakapan menggunakan sistem saraf untuk menghasilkan ucapan dalam gaya percakapan yang lebih ramah dan ekspresif yang dapat digunakan dalam banyak kasus penggunaan. Untuk informasi selengkapnya, lihat [Gaya Berbicara NTTS](#).

25 November 2019

Suara baru ditambahkan

Dua suara baru ditambahkan: Camila (perempuan, Portuguese-Brazil) dan Lupe (perempuan,). Spanish-US

23 Oktober 2019

Fitur baru ditambahkan	Penambahan Amazon Polly untuk plugin Windows untuk menggabungkan berbagai suara Amazon Polly ke dalam aplikasi Windows. SAPI-compliant	26 September 2019
Fitur baru utama	Selain suara text-to-speech (TTS) standar yang didukung oleh Amazon Polly sejak diluncurkan, Amazon Polly sekarang menyediakan sistem Neural TTS (NTTS) yang ditingkatkan yang dapat memberikan suara berkualitas lebih tinggi, sehingga memberi Anda suara text-to-speech yang paling alami dan mirip manusia. Untuk informasi lebih lanjut, lihat Neural Text-to-Speech .	30 Juli 2019
Suara baru ditambahkan	Suara baru ditambahkan: Lucia (wanita, Spanyol), dan Bianca (wanita, Italia).	2 Agustus 2018
Bahasa baru ditambahkan	Bahasa baru ditambahkan: Spanyol Meksiko (es-MX). Bahasa ini menggunakan suara perempuan Mia.	2 Agustus 2018

Bahasa baru ditambahkan	Bahasa baru ditambahkan: Hindi (Hi-in). Suara ini menggunakan suara perempuan Aditi, yang juga digunakan untuk bahasa Inggris India, membuat suara bilingual pertama Aditi Amazon Polly.	2 Agustus 2018
Fitur baru ditambahkan	Penambahan sintesis Pidato dari bagian teks panjang (hingga 100.000 karakter yang ditagih).	17 Juli 2018
Fitur SSML baru ditambahkan	Penambahan Durasi Maksimum untuk Pidato yang Disintesis .	17 Juli 2018
Suara baru ditambahkan	Suara baru ditambahkan: Léa (wanita, Prancis).	5 Juni 2018
Perluasan wilayah	Perluasan layanan Amazon Polly ke semua wilayah komersial.	4 Juni 2018
Bahasa baru ditambahkan	Bahasa baru ditambahkan: Korea (KO-KR).	4 Juni 2018
Fitur yang diperluas	Fitur Amazon Polly WordPress Plugin, termasuk penambahan kemampuan Amazon Translate.	4 Juni 2018
Suara baru ditambahkan	Dua suara baru ditambahkan: Aditi (perempuan, bahasa Inggris India) dan Seoyeon (perempuan, Korea).	15 November 2017

[Fitur baru](#)

Penambahan fitur [Tanda
Pidato](#) baru, serta perluasan
kemampuan [SSML..](#)

19 April 2017

[Panduan baru](#)

Ini adalah rilis pertama dari
Panduan Pengembang
Amazon Polly.

30 November 2016

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.