Guida per gli sviluppatori

Amazon Polly

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon Polly: Guida per gli sviluppatori

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

I marchi e l'immagine commerciale di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in una qualsiasi modalità che possa causare confusione tra i clienti o in una qualsiasi modalità che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà delle rispettive aziende, che possono o meno essere associate, collegate o sponsorizzate da Amazon.

Table of Contents

Cos'è Amazon Polly? ... 1
Come funziona ... 1
Vantaggi ... 2
È il primo utilizzo? ... 3
Lavorare con AWS SDKs ... 3
Nozioni di base ... 5
Iscrizione a AWS ... 5
Registrati per un Account AWS ... 5
Crea un utente con accesso amministrativo ... 6
Configurazione del AWS CLI ... 8
Riconfigurazione di AWS CLI ... 9
Esempio di sintesi vocale ... 11
Voci in Amazon Polly ... 15
Voci disponibili ... 15
Voci del marchio ... 23
Voci bilingue ... 23
Voci bilingue accentate ... 23
Voci completamente bilingue ... 24
Applicare la voce del giornalista ... 25
Ascoltare le voci ... 28
Cronometraggio della velocità di una voce ... 28
Modifica della velocità di una voce ... 29
Lingue in Amazon Polly ... 31
Arabo (arb) ... 34
Arabo (Golfo) (ar-AE) ... 39
Catalano (CA-ES) ... 46
Cinese (cantonese) (yue-CN) ... 50
Cinese, mandarino (cmn-CN) ... 55
Ceco (cs-CZ) ... 59
Danese (da-DK) ... 64
Olandese (belga) (nl-BE) ... 69
Olandese (nl-NL) ... 73
Inglese (Stati Uniti) (en-US) ... 77
Inglese (Australia) (en-AU) ... 81

Inglese (Regno Unito) (en-GB) ... 86
Inglese (India) (en-IN) ... 90
Inglese (Irlanda) (en-IE) ... 95
Inglese (Nuova Zelanda) (en-NZ) ... 99
Inglese (singaporiano) (en-SG) ... 105
Inglese (Sud Africa) (en-ZA) ... 110
Inglese (Galles) (en-GB-WLS) ... 115
Finlandese (fi-FI) ... 119
Francese (fr-FR) ... 125
Francese (belga) (fr-BE) ... 128
Francese (Canada) (fr-CA) ... 132
Tedesco (de-DE) ... 136
Tedesco (austriaco) (de-AT) ... 141
Tedesco (standard svizzero) (de-CH) ... 146
Hindi (hi-IN) ... 151
Islandese (is-IS) ... 154
Italiano (it-IT) ... 160
Giapponese (ja-JP) ... 163
Coreano (ko-KR) ... 167
Norvegese (nb-NO) ... 170
Polacco (pl-PL) ... 175
Portoghese (pt-PT) ... 178
Portoghese (Brasile) (pt-BR) ... 182
Romeno (ro-RO) ... 185
Russo (ru-RU) ... 189
Spagnolo (es-ES) ... 193
Spagnolo (Messico) (es-MX) ... 196
Spagnolo (Stati Uniti) (es-US) ... 199
Svedese (sv-SE) ... 202
Turco (tr-TR) ... 207
Gallese (cy-GB) ... 211
Motori vocali ... 217
Motore generativo ... 217
Voci generative disponibili ... 218
Compatibilità tra funzionalità e aree geografiche ... 220
Motore a forma lunga ... 221

Voci lunghe disponibili ... 222
Compatibilità tra funzionalità e aree geografiche ... 222
Motore neurale ... 223
Voci neurali disponibili ... 224
Compatibilità tra funzionalità e regioni ... 228
Motore standard ... 229
Voci standard disponibili ... 230
Compatibilità tra funzionalità e aree geografiche ... 233
Scelta di un motore vocale ... 234
Segni vocali ... 236
Tipi di segni vocali ... 236
Visemi e Amazon Polly ... 237
Uscita del contrassegno vocale ... 238
Richiesta di segni vocali ... 239
Esempio di segni vocali senza SSML ... 241
Esempio di segni vocali con SSML ... 242
Uso di SSML ... 244
Personaggi riservati ... 245
Utilizzo di SSML sulla console ... 248
Utilizzo di SSML con il comando Synthesize-Speech ... 249
Sintetizzazione di un documento ottimizzato con SSML ... 250
Tag SSML supportati ... 251
Identificazione del testo migliorato con SSML ... 254
Aggiungere una pausa ... 254
Enfatizzare le parole ... 255
Specificare un'altra lingua per parole specifiche ... 256
Inserimento di un tag personalizzato nel testo ... 258
Aggiungere una pausa tra i paragrafi ... 258
Uso della pronuncia fonetica ... 259
Controllo del volume, della velocità di conversazione e dell'intonazione ... 260
Impostazione di una durata massima per il parlato sintetizzato ... 263
Aggiungere una pausa tra le frasi ... 267
Controllo del modo in cui vengono pronunciati tipi speciali di parole ... 267
Pronunciare acronimi e abbreviazioni ... 271
Migliorare la pronuncia specificando parti del discorso ... 271
Aggiungere il suono del respiro ... 273

Stile di pronuncia newscaster ... 277
Aggiungere la compressione della gamma dinamica ... 278
Parlando a bassa voce ... 280
Controllo del timbro ... 281
Sussurrare ... 282
Gestione dei lessici ... 284
Utilizzo di più lessici ... 285
Caricamento di un lessico ... 286
Applicazione dei lessici (sintesi vocale) ... 292
Filtrare l'elenco dei lessici sulla console ... 295
Scaricamento dei lessici sulla console ... 297
Eliminazione di un lessico ... 298
File audio lunghi ... 300
Configurazione della politica IAM per la sintesi asincrona ... 301
Creazione di file audio lunghi ... 302
Quote ... 307
Regioni supportate ... 308
Quote e velocità di limitazione ... 308
Richieste simultanee ... 309
Best practice per mitigare la limitazione ... 309
Lessici di pronuncia ... 310
SynthesizeSpeech Operazioni API ... 310
SpeechSynthesisTask Operazioni API ... 311
SSML (Speech Synthesis Markup Language) ... 311
Codice e applicazioni di esempio ... 313
Esempi di Java ... 313
DeleteLexicon ... 314
DescribeVoices ... 316
GetLexicon ... 317
ListLexicons ... 319
PutLexicon ... 320
StartSpeechSynthesisTask ... 323
Contrassegni vocali ... 326
SynthesizeSpeech ... 329
Esempi Python ... 331
DeleteLexicon ... 331

GetLexicon ... 332
ListLexicon ... 334
PutLexicon ... 334
StartSpeechSynthesisTask ... 336
SynthesizeSpeech ... 337
Esempio di Java ... 337
Esempio di Python ... 342
Esempio in Python: index.html ... 344
Esempio in Python: server.py ... 348
Esempio iOS ... 355
Esempio Android ... 357
Esempi di codice ... 360
Nozioni di base ... 361
Azioni ... 361
Scenari ... 403
Conversione di sintesi vocale e di nuovo in testo ... 403
Crea un'applicazione di sincronizzazione labiale ... 404
Crea un'applicazione per analizzare il feedback dei clienti ... 405
Sicurezza ... 412
Protezione dei dati ... 413
Crittografia dei dati inattivi ... 413
Crittografia in transito ... 414
Riservatezza del traffico Internet ... 414
Identity and Access Management ... 414
Destinatari ... 414
Autenticazione con identità ... 415
Gestione dell'accesso con policy ... 419
Come Amazon Polly funziona con IAM ... 421
Esempi di policy basate su identità ... 430
Informazioni di riferimento delle autorizzazioni API Amazon Polly ... 437
Risoluzione dei problemi ... 439
Registrazione e monitoraggio ... 441
Convalida della conformità ... 441
Resilienza ... 442
Sicurezza dell'infrastruttura ... 442
Best practice di sicurezza ... 443

Utilizzo di endpoint VPC dell'interfaccia 443
Disponibilità 444
Creazione di un endpoint VPC per Amazon Polly 444
Verifica della connessione tra il VPC e Amazon Polly 444
Controllo dell'accesso all'endpoint Amazon Polly 444
Supporto delle chiavi di contesto VPC 445
Registrazione delle chiamate API Amazon Polly con AWS CloudTrail 446
Informazioni su Amazon Polly in CloudTrail 446
Esempio: voci del file di log di Amazon Polly 447
CloudWatch integrazione 449
Ottenere CloudWatch metriche (console) 449
Ottenere le metriche su CloudWatch AWS CLI 449
Parametri di Amazon Polly 450
Dimensioni per i parametri Amazon Polly 451
Documentazione di riferimento delle API 453
Azioni 453
DeleteLexicon 454
DescribeVoices 456
GetLexicon 460
GetSpeechSynthesisTask 463
ListLexicons 466
ListSpeechSynthesisTasks 469
PutLexicon 472
StartSpeechSynthesisTask 475
SynthesizeSpeech 483
Tipi di dati 489
Lexicon 490
LexiconAttributes 491
LexiconDescription 493
SynthesisTask 494
Voice 499
Errori comuni 501
Parametri comuni 503
Cronologia dei documenti 506
..... dxxi

Cos'è Amazon Polly?

Amazon Polly è un servizio cloud che converte il testo in voce naturale. Puoi utilizzare Amazon Polly per sviluppare applicazioni che aumentano il coinvolgimento e l'accessibilità. Amazon Polly supporta più lingue e include una varietà di voci realistiche. Con Amazon Polly, puoi creare applicazioni vocali che funzionano in più sedi e utilizzano la voce ideale per i tuoi clienti. Inoltre, paghi solo per il testo che sintetizzi. Puoi anche memorizzare nella cache e riprodurre l'audio generato da Amazon Polly; senza costi aggiuntivi.

Amazon Polly offre molte opzioni vocali, tra cui opzioni generative, lunghe, neurali e standard text-to-speech (TTS). Queste voci offrono miglioramenti rivoluzionari nella qualità del parlato utilizzando una nuova tecnologia di apprendimento automatico per offrire le voci più naturali e simili a quelle umane possibili. text-to-speech La tecnologia TTS neurale supporta anche uno stile di conversazione Newscaster, adattato ai casi d'uso della narrazione di notizie.

I casi d'uso più comuni di Amazon Polly includono, ma non sono limitati a: applicazioni mobili come newsreader, giochi, piattaforme di eLearning, applicazioni di accessibilità per persone ipovedenti e il segmento in rapida crescita dell'Internet of Things (IoT).

Amazon Polly è certificato per l'uso con carichi di lavoro regolamentati per HIPAA (Health Insurance Portability and Accountability Act del 1996) e Payment Card Industry Data Security Standard (PCI DSS).

Come funziona Amazon Polly

Amazon Polly converte testo di input in voce naturale. Per usare una voce Amazon Polly, scegli un motore vocale, richiama un metodo di sintesi vocale, fornisci il testo che desideri sintetizzare, quindi specifica un formato di uscita audio. Amazon Polly sintetizza il testo fornito in un flusso audio vocale di alta qualità.

- Testo di input: fornisci il testo da sintetizzare e Amazon Polly restituisce un flusso audio. Puoi fornire l'input come testo semplice o in formato Speech Synthesis Markup Language (SSML). Con SSML è possibile controllare diversi aspetti della sintesi restituita, ad esempio pronuncia, volume, intonazione e velocità. Per ulteriori informazioni, consulta Generazione vocale da documenti SSML.
- Voci disponibili: Amazon Polly offre un portafoglio di lingue e un'ampia gamma di voci, tra cui una voce bilingue (per inglese e hindi). Per la maggior parte delle lingue, è possibile scegliere tra diverse voci, sia maschili sia femminili. Quando si avvia l'attività di sintesi vocale, specificare il nome ID della voce, e poi Amazon Polly utilizza tale voce per convertire il testo in voce naturale.

Amazon Polly non è un servizio di traduzione e la sintesi vocale viene restituita nella stessa lingua del testo. I numeri rappresentati come cifre (ad esempio 53, non cinquantatré) vengono sintetizzati nel linguaggio della voce e non nel testo. Per ulteriori informazioni, consulta Voices in Amazon Polly (Voci in Amazon Polly).

- Formato output: Amazon Polly può restituire la sintesi vocale in più formati. È possibile selezionare il formato audio più adatto alle proprie esigenze. Ad esempio, è possibile richiedere la voce nel formato Ogg Vorbis MP3 per l'utilizzo da parte di applicazioni Web e mobili. In alternativa, è possibile richiedere il formato di output PCM per l'utilizzo da parte di AWS IoT dispositivi e soluzioni di telefonia.

> **Note**
>
> Per ascoltare esempi di voci di Amazon Polly nel tuo browser, consulta la panoramica del prodotto Amazon Polly.

Vantaggi

Alcuni dei vantaggi derivanti dall'utilizzo di Amazon Polly sono:

- Alta qualità: Amazon Polly offre voci generative, di lunga durata, neurali e di alta qualità (TTS) ad alte prestazioni. text-to-speech Queste tecnologie sintetizzano il linguaggio naturale con un'elevata precisione di pronuncia (incluse abbreviazioni, espansioni di acronimi, interpretazioni di data/ora e disambiguazione omografica).
- Bassa latenza: Amazon Polly offre risposte rapide, il che lo rende un'opzione valida per casi d'uso a bassa latenza come i sistemi di dialogo.
- Supporto per un ampio portafoglio di lingue e voci: Amazon Polly supporta dozzine di voci e lingue, offrendo opzioni vocali maschili e femminili per la maggior parte delle lingue. Questo numero continuerà ad aumentare man mano che vengono rese disponibili più voci neurali online. Le voci in inglese Stati Uniti di Matthew e Joanna possono anche utilizzare lo stile neurale newscaster, analogamente a quanto si potrebbe sentire in un notiziario professionale.
- Conveniente: il pay-per-use modello di Amazon Polly significa che non ci sono costi di configurazione. Inizia in piccolo e amplia man mano che la tua applicazione cresce.
- Soluzione basata su cloud: le soluzioni TTS installate sui dispositivi richiedono un numero significativo di risorse di elaborazione, una CPU molto potente e una grande quantità di RAM

e spazio su disco. Ciò può comportare costi di sviluppo più elevati e un maggiore consumo energetico su dispositivi come tablet, smartphone e così via. Al contrario, la conversione TTS effettuata nel riduce Cloud AWS drasticamente il fabbisogno di risorse locali. Ciò consente il supporto di tutte le lingue e le voci disponibili con una qualità eccezionale. Inoltre, i miglioramenti vocali sono immediatamente disponibili per tutti gli utenti finali e non richiedono aggiornamenti aggiuntivi per i dispositivi.

> **Note**
>
> Per ascoltare esempi di voci di Amazon Polly nel tuo browser, consulta la panoramica del prodotto Amazon Polly.

È il primo utilizzo?

Se utilizzi Amazon Polly per la prima volta, ti consigliamo di leggere le seguenti sezioni nell'ordine elencato:

1. Come funziona Amazon Polly— Questa sezione presenta vari input e opzioni di Amazon Polly con cui puoi lavorare per creare un'esperienza semplice.
2. Guida introduttiva ad Amazon Polly: in questa sezione puoi impostare il tuo account e provare la sintesi vocale di Amazon Polly.
3. Codice di esempio e applicazioni per Amazon Polly: in questa sezione vengono forniti ulteriori esempi che puoi utilizzare per esplorare Amazon Polly.

Utilizzo di Amazon Polly con un SDK AWS

AWS i kit di sviluppo software (SDKs) sono disponibili per molti linguaggi di programmazione più diffusi. Ogni SDK fornisce un'API, esempi di codice, e documentazione che facilitano agli sviluppatori la creazione di applicazioni nel loro linguaggio preferito.

Documentazione sugli SDK	Esempi di codice
AWS SDK per C++	AWS SDK per C++ esempi di codice
AWS CLI	AWS CLI esempi di codice

Documentazione sugli SDK	Esempi di codice
AWS SDK per Go	AWS SDK per Go esempi di codice
AWS SDK per Java	AWS SDK per Java esempi di codice
AWS SDK per JavaScript	AWS SDK per JavaScript esempi di codice
AWS SDK per Kotlin	AWS SDK per Kotlin esempi di codice
AWS SDK per .NET	AWS SDK per .NET esempi di codice
AWS SDK per PHP	AWS SDK per PHP esempi di codice
AWS Strumenti per PowerShell	AWS Strumenti per PowerShell esempi di codice
AWS SDK per Python (Boto3)	AWS SDK per Python (Boto3) esempi di codice
AWS SDK per Ruby	AWS SDK per Ruby esempi di codice
AWS SDK for Rust	AWS SDK for Rust esempi di codice
SDK AWS per SAP ABAP	SDK AWS per SAP ABAP esempi di codice
SDK AWS per Swift	SDK AWS per Swift esempi di codice

Esempio di disponibilità

Non riesci a trovare quello che ti serve? Richiedi un esempio di codice utilizzando il link Provide feedback (Fornisci un feedback) nella parte inferiore di questa pagina.

Guida introduttiva ad Amazon Polly

Amazon Polly offre diverse operazioni API che puoi integrare facilmente con le tue applicazioni esistenti. Per un elenco delle operazioni supportate, consulta Azioni.

È possibile eseguire quasi tutte le stesse operazioni sulla console Amazon Polly e su. AWS CLI Tuttavia, non è possibile ascoltare la voce sintetizzata su. AWS CLI Per utilizzare l'audio su AWS CLI, salvate il testo in un file. Quindi apri il file in un'applicazione audio a tua scelta.

Puoi utilizzare una delle seguenti opzioni:

- AWS SDKs — Quando utilizzi SDKs, le tue richieste ad Amazon Polly vengono automaticamente firmate e autenticate utilizzando le credenziali fornite. Questa è la soluzione consigliata per la creazione di applicazioni.
- AWS CLI — Puoi AWS CLI usare Amazon Polly senza scrivere alcun codice.

Prima di utilizzare Amazon Polly per la prima volta, devi registrarti. AWS Quando ti registri ad Amazon Web Services (AWS), il tuo AWS account viene automaticamente registrato per tutti i servizi in AWS, incluso Amazon Polly. Ti vengono addebitati solo i servizi e le risorse che utilizzi. Se sei un nuovo AWS cliente, puoi iniziare a usare Amazon Polly gratuitamente. Per ulteriori informazioni, consulta Piano di utilizzo gratuito di AWS.

Le sezioni seguenti descrivono come iniziare a utilizzare Amazon Polly.

Argomenti

- Iscrizione a AWS
- Configurazione del AWS CLI
- Riconfigurazione del AWS CLI

Iscrizione a AWS

Prima di poter utilizzare qualsiasi AWS servizio, incluso Amazon Polly, devi registrarti. AWS

Registrati per un Account AWS

Se non ne hai uno Account AWS, completa i seguenti passaggi per crearne uno.

Per iscriverti a un Account AWS

1. Apri la https://portal.aws.amazon.com/billing/registrazione.
2. Segui le istruzioni online.

   Nel corso della procedura di registrazione riceverai una telefonata o un messaggio di testo e ti verrà chiesto di inserire un codice di verifica attraverso la tastiera del telefono.

   Quando ti iscrivi a un Account AWS, Utente root dell'account AWSviene creato un. L'utente root dispone dell'accesso a tutte le risorse e tutti i Servizi AWS nell'account. Come best practice di sicurezza, assegna l'accesso amministrativo a un utente e utilizza solo l'utente root per eseguire attività che richiedono l'accesso di un utente root.

AWS ti invia un'email di conferma dopo il completamento della procedura di registrazione. In qualsiasi momento, puoi visualizzare l'attività corrente del tuo account e gestirlo accedendo a https://aws.amazon.com/e scegliendo Il mio account.

Crea un utente con accesso amministrativo

Dopo esserti registrato Account AWS, proteggi Utente root dell'account AWS AWS IAM Identity Center, abilita e crea un utente amministrativo in modo da non utilizzare l'utente root per le attività quotidiane.

Proteggi i tuoi Utente root dell'account AWS

1. Accedi AWS Management Consolecome proprietario dell'account scegliendo Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.

   Per informazioni sull'accesso utilizzando un utente root, consulta la pagina Signing in as the root user della Guida per l'utente di Accedi ad AWS .
2. Abilita l'autenticazione a più fattori (MFA) per l'utente root.

   Per istruzioni, consulta Abilitare un dispositivo MFA virtuale per l'utente Account AWS root (console) nella Guida per l'utente IAM.

Crea un utente con accesso amministrativo

1. Abilita Centro identità IAM.

Per istruzioni, consulta Abilitazione di AWS IAM Identity Center nella Guida per l'utente di AWS IAM Identity Center .

2. In IAM Identity Center, assegna l'accesso amministrativo a un utente.

   Per un tutorial sull'utilizzo di IAM Identity Center directory come fonte di identità, consulta Configurare l'accesso utente con l'impostazione predefinita IAM Identity Center directory nella Guida per l'AWS IAM Identity Center utente.

Accesso come utente amministratore

- Per accedere con l'utente IAM Identity Center, utilizza l'URL di accesso che è stato inviato al tuo indirizzo e-mail quando hai creato l'utente IAM Identity Center.

  Per informazioni sull'accesso utilizzando un utente IAM Identity Center, consulta AWS Accedere al portale di accesso nella Guida per l'Accedi ad AWS utente.

Assegna l'accesso a ulteriori utenti

1. In IAM Identity Center, crea un set di autorizzazioni conforme alla best practice dell'applicazione di autorizzazioni con il privilegio minimo.

   Segui le istruzioni riportate nella pagina Creazione di un set di autorizzazioni nella Guida per l'utente di AWS IAM Identity Center .

2. Assegna al gruppo prima gli utenti e poi l'accesso con autenticazione unica (Single Sign-On).

   Per istruzioni, consulta Aggiungere gruppi nella Guida per l'utente di AWS IAM Identity Center .

Per ulteriori informazioni su IAM, consulta:

- AWS Identity and Access Management (IAM)
- Nozioni di base
- Guida per l'utente di IAM

> **Note**
>
> Annota l'ID AWS del tuo account. Ti servirà nei passaggi successivi.

Configurazione del AWS CLI

Segui questi passaggi per scaricarlo e configurarlo AWS CLI per funzionare con Amazon Polly.

Per configurare il AWS Command Line Interface

1. Scarica e configura la AWS CLI. Per le istruzioni, consulta i seguenti argomenti nella Guida per l'utente di AWS Command Line Interface :

   - Configurarsi con il AWS Command Line Interface
   - Configurazione di AWS Command Line Interface

2. Aggiungi un profilo denominato per l'utente amministratore nel AWS CLI AWS file Config. È possibile utilizzare questo profilo quando si eseguono i AWS CLI comandi. Per ulteriori informazioni sui profili denominati, consulta Profili denominati in Guida per l'utente dell'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
   ```

   Per un elenco delle AWS regioni disponibili e di quelle supportate da Amazon Polly, consulta Regioni ed endpoint nel. Riferimenti generali di Amazon Web Services

   > **Note**
   >
   > Se utilizzi una regione supportata da Amazon Polly che hai specificato al momento della configurazione AWS CLI, ometti la riga seguente dagli esempi di codice. AWS CLI
   >
   > ```
   > --region aws-region
   > ```

3. Verifica la configurazione digitando il seguente comando help al prompt dei comandi.

```
aws help
```

Nella finestra dovrebbe apparire un elenco di AWS comandi validi. AWS CLI

Riconfigurazione del AWS CLI

Se in precedenza hai scaricato e configurato Amazon Polly potrebbe non essere disponibile a meno che non riconfiguri il. AWS CLI AWS CLI La seguente procedura verifica se ciò è necessario.

Per riattivare Amazon Polly dal AWS CLI

1. Verifica la disponibilità di Amazon Polly digitando il seguente comando help al prompt dei AWS CLI comandi.

   ```
   aws polly help
   ```

   Se vedi una descrizione di Amazon Polly e nella AWS CLI finestra viene visualizzato un elenco di comandi validi, puoi utilizzare Amazon Polly immediatamente. AWS CLI In questo caso, è possibile ignorare il resto di questa procedura. Se non viene visualizzato, continuare con la Fase 2.

2. Attiva Amazon Polly utilizzando una delle due seguenti opzioni:

   a. Disinstalla e reinstalla il. AWS CLI

      Per le istruzioni, consulta Installazione di AWS Command Line Interface nella Guida per l'utente di AWS Command Line Interface .

      oppure

   b. Scarica il file service-2.json.

      Nel prompt dei comandi, eseguire il seguente comando .

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

3. Verificare di nuovo la disponibilità di Amazon Polly.

```
aws polly help
```

La descrizione di Amazon Polly dovrebbe essere visualizzata.

Esempio di sintesi vocale con Amazon Polly

Questa pagina presenta un breve esempio di sintesi vocale eseguito nella console, in e con Python. AWS CLI Questo esempio esegue la sintesi vocale da testo semplice, non da SSML.

Console

Sintetizza il parlato sulla console

1. Accedi a AWS Management Console e apri la console Amazon Polly all'indirizzo. https://console.aws.amazon.com/polly/
2. Scegli la scheda Text-to-Speech (Sintesi vocale). Il campo di testo verrà caricato con un testo di esempio in modo da poter provare rapidamente Amazon Polly.
3. Disattiva SSML.
4. Digita o incolla questo testo nella casella di input.

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

5. In Engine, scegli Generative, Long Form, Neural o Standard.
6. Scegli una lingua e una AWS regione, quindi scegli una voce. (Se selezioni Neural for Engine, sono disponibili solo le lingue e le voci che supportano NTTS. Tutte le voci Standard e Long Form sono disabilitate.)
7. Per ascoltare immediatamente la sintesi vocale, scegli Listen (Ascolta).
8. Per salvare la sintesi vocale in un file, esegui una delle operazioni seguenti:
   a. Scegli Download (Scarica).
   b. Per passare a un formato di file diverso, espandi Additional settings (Impostazioni aggiuntive) attiva Speech file format settings (Impostazioni del formato di file vocale), scegli il formato di file desiderato e successivamente scegli Download.

AWS CLI

In questo esercizio, richiamerete l'`SynthesizeSpeech`operazione passando il testo di input. Puoi salvare l'audio risultante come un file e verificarne il contenuto.

1. Eseguite il `synthesize-speech` AWS CLI comando per sintetizzare il testo di esempio in un file audio ()`hello.mp3`.

   L' AWS CLI esempio seguente è formattato per Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione Unix barra rovesciata (\) al termine di ogni riga con un accento circonflesso (^) e usa virgolette (") attorno al testo di input con virgolette singole (') per i tag interni.

   ```
   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
   ```

   Nella chiamata `asynthesize-speech`, fornisci un testo di esempio da sintetizzare con una voce a tua scelta. È necessario fornire un ID vocale (spiegato nel passaggio successivo) e un formato di output. Il comando consente di salvare l'audio risultante nel file `hello.mp3`. Oltre al MP3 file, l'operazione invia il seguente output alla console.

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

2. Riproduci il file `hello.mp3` risultante per verificare la sintesi vocale.

Python

Per eseguire il test del codice di esempio Python, è necessario AWS SDK for Python (Boto). Per istruzioni, consultare AWS SDK per Python (Boto3).

Il codice Python in questo esempio esegue le seguenti azioni:

- Lo richiama AWS SDK for Python (Boto) per inviare una `SynthesizeSpeech` richiesta ad Amazon Polly (fornendo del testo come input).
- Viene eseguito l'accesso al flusso audio risultante nella risposta e viene salvato l'audio in un file sul disco locale (`speech.mp3`).
- Viene riprodotto il file audio con il lettore audio di default per il sistema locale.

Salva il codice in un file (example.py) ed eseguilo.

```
"""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])
```

Per esempi più dettagliati, consulta gli argomenti seguenti:

- Utilizzo di SSML sulla console
- Applicazione dei lessici (sintesi vocale)
- Codice di esempio e applicazioni per Amazon Polly

Voci in Amazon Polly

Amazon Polly offre dozzine di voci realistiche in un'ampia gamma di lingue. Ogni voce viene creata utilizzando persone madrelingua, perciò possono esserci variazioni da voce a voce, anche all'interno della stessa lingua. Puoi anche usare il AWS Management Console per testare ogni voce con un testo a tua scelta. Nella maggior parte dei casi, vi sarà almeno una voce maschile e una femminile (spesso ce n'è più di una). Solo alcune lingue hanno una voce singola.

L'inventario di voci e il numero di lingue incluse viene costantemente aggiornato per includere opzioni aggiuntive. Per suggerire una nuova lingua o voce, non esitare a fornire un feedback su questa pagina. Purtroppo, non siamo in grado di dare informazioni sui piani di rilascio di nuove lingue.

> **Note**
>
> Per ascoltare esempi di voci di Amazon Polly nel tuo browser, consulta la panoramica del prodotto Amazon Polly.

Argomenti

- Voci disponibili
- Voci bilingue
- Applicare la voce del giornalista
- Ascoltare le voci
- Cronometraggio della velocità di una voce
- Modifica della velocità di una voce

Voci disponibili

Amazon Polly offre una varietà di voci realistiche in più lingue per la sintesi vocale dal testo. La tabella seguente mostra tutte le voci offerte da Amazon Polly.

	Lingua e varianti linguistiche	Codice lingua	Nome/ID	Gender	Voce generativa	Voce in forma lunga	Voce neurale	Voce standard
1	Arabo	arb	Zeina	Femmina	No	No	No	Sì
2	Arabo (Golfo)	ar-AE	Hala*	Femmina	No	No	Sì	No
			Zayd*	Maschio	No	No	Sì	No
3	Olandese (belga)	nl-BE	Lisa	Femmina	No	No	Sì	No
4	Catalano	ca-ES	Arlet	Femmina	No	No	Sì	No
5	Ceco	CS-cz	Jitka	Femmina	No	No	Sì	No
6	Cinese (Cantonese)	yue-CN	Hiujin	Femmina	No	No	Sì	No
7	Cinese (Mandarino)	cmn-CN	Zhiyu	Femmina	No	No	Sì	Sì
8	Danese	da-DK	Naja	Femmina	No	No	No	Sì
			Mads	Maschio	No	No	No	Sì
			Sofie	Femmina	No	No	Sì	No
9	Olandese	nl-NL	Laura	Femmina	No	No	Sì	No
			Lotte	Femmina	No	No	No	Sì
			Ruben	Maschio	No	No	No	Sì

	Lingua e varianti linguistiche	Codice lingua	Nome/ ID	Gender	Voce generativa	Voce in forma lunga	Voce neurale	Voce standard
10	Inglese (Australia)	en-AU	Nicole	Femmina	No	No	No	Sì
			Olivia	Femmina	Sì	No	Sì	No
			Russell	Maschio	No	No	No	Sì
11	Inglese (Regno Unito)	en-GB	Amy**	Femmina	Sì	No	Sì	Sì
			Emma	Femmina	No	No	Sì	Sì
			Brian	Maschio	No	No	Sì	Sì
			Arturo	Maschio	No	No	Sì	No
12	Inglese (India)	en-IN	Aditi*	Femmina	No	No	No	Sì
			Raveena	Femmina	No	No	No	Sì
			Kajal*	Femmina	Sì	No	Sì	No
13	Inglese (Irlanda)	en-IN	Niamh	Femmina	No	No	Sì	No
14	Inglese (Nuova Zelanda)	en-NZ	Aria	Femmina	No	No	Sì	No
15	Inglese (singaporiano)	it-SG	Gelsomin	Femmina	No	No	Sì	No
16	Inglese (Sud Africa)	en-ZA	Ayanda	Femmina	Sì	No	Sì	No

	Lingua e varianti linguistiche	Codice lingua	Nome/ID	Gender	Voce generativa	Voce in forma lunga	Voce neurale	Voce standard
17	Inglese (Stati Uniti)	it-IT	Danielle	Femmina	Sì	Sì	Sì	No
			Gregory	Maschio	No	Sì	Sì	No
			Ivy	Femmina (bambina)	No	No	Sì	Sì
			Joanna**		Sì	No	Sì	Sì
			Kendra	Femmina	No	No	Sì	Sì
			Kimberly	Femmina	No	No	Sì	Sì
			Salli	Femmina	No	No	Sì	Sì
			Joey	Femmina	No	No	Sì	Sì
			Justin	Maschio	No	No	Sì	No
			Kevin	Maschio (bambino)	No	No	Sì	Sì
			Matthew**	Maschio (bambino)	Sì	No	Sì	No
			Ruth**		Sì	Sì	Sì	No
			Stephen**	Maschio	Sì	No	Sì	No
			Patrizio	Femmina	No	Sì	No	No
				Maschio				
				Maschio				
18	Inglese (Galles)	en-GB-WLS	Geraint	Maschio	No	No	No	Sì

	Lingua e varianti linguistiche	Codice lingua	Nome/ID	Gender	Voce generativa	Voce in forma lunga	Voce neurale	Voce standard
19	Finlandese	fi-FI	Suvi	Femmina	No	No	Sì	No
20	Francese	fr-FR	Céline/Celine	Femmina	No	No	No	Sì
			Léa	Femmina	Sì	No	Sì	Sì
			Mathieu	Maschio	No	No	No	Sì
			Rémi**	Maschio	Sì	No	Sì	No
21	Francese (belga)	fr-BE	Isabelle	Femmina	No	No	Sì	No
22	Francese (Canada)	fr-CA	Chantal	Femmina	No	No	No	Sì
			Gabrielle	Femmina	No	No	Sì	No
			Liam	Maschio	No	No	Sì	No
23	Tedesco	de-DE	Marlene	Femmina	No	No	No	Sì
			Vicki	Femmina	Sì	No	Sì	Sì
			Hans	Maschio	No	No	No	Sì
			Daniele	Maschio	Sì	No	Sì	No
24	Tedesco (austriaco)	de-AT	Hannah	Femmina	No	No	Sì	No

	Lingua e varianti linguistiche	Codice lingua	Nome/ID	Gender	Voce generativa	Voce in forma lunga	Voce neurale	Voce standard
25	Tedesco (svizzero)	de-CH	Sabrina	Femmina	No	No	Sì	No
26	Hindi	hi-IN	Aditi*	Femmina	No	No	No	Sì
			Kajal*	Femmina	No	No	Sì	No
27	Islandese	is-IS	Dóra/Dora	Femmina	No	No	No	Sì
			Karl	Maschio	No	No	No	Sì
28	Italiano	it-IT	Carla	Femmina	No	No	No	Sì
			Bianca	Femmina	Sì	No	Sì	Sì
			Giorgio	Maschio	No	No	No	Sì
			Adriano**	Maschio	No	No	Sì	No
29	Giapponese	ja-JP	Mizuki	Femmina	No	No	No	Sì
			Takumi	Maschio	No	No	Sì	Sì
			Kazuha	Femmina	No	No	Sì	No
			Tomoko	Femmina	No	No	Sì	No
30	Coreano	ko-KR	Seoyeon	Femmina	No	No	Sì	Sì
			Jihye	Femmina	No	No	Sì	No

	Lingua e varianti linguistiche	Codice lingua	Nome/ID	Gender	Voce generativa	Voce in forma lunga	Voce neurale	Voce standard
31	Norvegese	nb-NO	Liv	Femmina	No	No	No	Sì
			Ida	Femmina	No	No	Sì	No
32	Polacco	pl-PL	Ewa	Femmina	No	No	No	Sì
			Maja	Femmina	No	No	No	Sì
			Jacek	Maschio	No	No	No	Sì
			Jan	Maschio	No	No	No	Sì
			Ola	Femmina	No	No	Sì	No
33	Portoghese (Brasile)	pt-BR	Camila	Femmina	No	No	Sì	Sì
			Vitória/Vitoria	Femmina	No	No	Sì	Sì
			Ricardo	Maschio	No	No	No	Sì
			Thiago**	Maschio	No	No	Sì	No
34	Portoghese (Europa)	pt-PT	Inês/Ines	Femmina	No	No	Sì	Sì
			Cristiano	Maschio	No	No	No	Sì
35	Romeno	ro-RO	Carmen	Femmina	No	No	No	Sì
36	Russo	ru-RU	Tatyana	Femmina	No	No	No	Sì
			Maxim	Maschio	No	No	No	Sì

	Lingua e varianti linguistiche	Codice lingua	Nome/ID	Gender	Voce generativa	Voce in forma lunga	Voce neurale	Voce standard
37	Spagnolo (Spagna)	es-ES	Conchita	Femmina	No	No	No	Sì
			Lucia	Femmina	Sì	No	Sì	Sì
			Alba	Femmina	No	Sì	No	No
			Enrique	Maschio	No	No	No	Sì
			Sergio**	Maschio	Sì	No	Sì	No
			Raúl	Maschio	No	Sì	No	No
38	Spagnolo (messicano)	es-MX	Mia	Femmina	Sì	No	Sì	Sì
			Andres**	Maschio	Sì	No	Sì	No
39	Spagnolo (Stati Uniti)	es-US	Lupe**	Femmina	Sì	No	Sì	Sì
			Penélope/Penelope	Femmina	No	No	No	Sì
			Miguel	Maschio	No	No	No	Sì
			Pedro	Maschio	Sì	No	Sì	No
40	Svedese	sv-SE	Astrid	Femmina	No	No	No	Sì
			Elin	Femmina	No	No	Sì	No
41	Turco	tr-TR	Filiz	Femmina	No	No	No	Sì
			Burcu	Femmina	No	No	Sì	No
42	Gallese	cy-GB	Gwyneth	Femmina	No	No	No	Sì

* Questa voce è bilingue. Per ulteriori informazioni, consulta Voci bilingue.

** Queste voci possono essere utilizzate con le stili di pronuncia newscaster con il formato neurale. Per ulteriori informazioni, consulta Applicare la voce del giornalista.

Ogni motore vocale Amazon Polly ha caratteristiche uniche. Scopri di più sulle funzionalità e sulla disponibilità regionale per i motori vocali offerti da Amazon Polly:

- Voci generative
- Voci di forma lunga
- Voci neurali
- Voci standard

Voci del marchio

Oltre alle voci disponibili elencate nella tabella precedente, puoi utilizzare Amazon Polly per creare una voce personalizzata per il personaggio del tuo marchio. Con la voce del marchio, puoi offrire voci uniche ed esclusive ai tuoi clienti. Per ulteriori informazioni sulle voci del marchio Amazon Polly, consulta Brand Voice.

Voci bilingue

Amazon Polly ha due modi per produrre voci bilingue:

- Voci bilingue accentate
- Voci completamente bilingue

Voci bilingue accentate

Le voci bilingue accentate possono essere create utilizzando qualsiasi voce Amazon Polly, ma solo quando si utilizzano tag SSML.

Normalmente, tutte le parole nel testo di input vengono pronunciate nella lingua predefinita della voce specificata in uso.

Ad esempio, se si sta utilizzando la voce di Joanna (che parla inglese Stati Uniti), Amazon Polly pronuncia quanto segue con la voce di Joanna senza un accento francese:

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

In questo caso, le parole Je ne parle pas français sono pronunciate come se fossero in inglese.

Tuttavia, se si utilizza la voce Joanna con il tag **<lang>**, Amazon Polly pronuncia la frase con la voce di Joanna in francese con accento americano:

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

Poiché Joanna non è madrelingua francese, la pronuncia si basa sulla sua lingua nativa, ovvero l'inglese degli Stati Uniti. Ad esempio, sebbene una perfetta pronuncia francese presenti una /R/ vibrata alveolare nella parola français, la voce di Joanna in inglese americano pronuncia questo fonema con il suono corrispondente /r/.

Se si utilizza la voce di Giorgio, che parla italiano, con il seguente testo, Amazon Polly pronuncia la frase con la voce di Giorgio e la pronuncia in italiano:

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

Voci completamente bilingue

Una voce bilingue come Aditi o Kajal (inglese indiano e hindi) è in grado di parlare due lingue fluentemente. Questo ti offre la possibilità di utilizzare parole e frasi di entrambe le lingue in un unico testo utilizzando la stessa voce.

Al momento, Aditi, Kajal, Hala e Zayd sono le uniche due voci completamente bilingue disponibili.

Utilizzo di una voce bilingue (ad esempio, Aditi)

Aditi parla sia inglese indiano (en-IN) sia hindi (hi-IN) in modo scorrevole. È possibile sintetizzare testo sia in inglese sia in hindi e la voce può passare da una lingua all'altra anche all'interno della stessa frase.

L'hindi può essere utilizzato in due diverse forme:

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

Inoltre, è possibile combinare inglese e una o entrambe le forme di hindi all'interno di una stessa frase:

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

Poiché Aditi è una voce bilingue, il testo riportato in tutti questi esempi sarà letto correttamente, perché Amazon Polly è in grado di distinguere lingue e scritture.

Amazon Polly supporta anche numeri, date, orari e valute sia in inglese (numeri arabi) sia in hindi (numeri devanagari). Per impostazione predefinita, i numeri arabi vengono letti in inglese indiano. Per fare in modo che Amazon Polly li legga in hindi, è necessario utilizzare il parametro di codice della lingua `hi-IN`.

Applicare la voce del giornalista

Le persone utilizzano stili di pronuncia diversi, a seconda del contesto. Le conversazioni occasionali, ad esempio, sembrano molto diverse dai notiziari televisivi o radiofonici. A causa del modo in cui voci standard sono composte, non possono produrre stili di pronuncia diversi. Tuttavia, le voci neurali possono. Possono essere addestrati per uno stile di pronuncia specifico, con le variazioni e l'enfasi su alcune parti del discorso inerenti a tale stile.

Oltre alle voci neurali predefinite, Amazon Polly fornisce uno stile di sintesi vocale newscaster che utilizza il sistema neurale per generare la sintesi vocale nello stile di un conduttore televisivo o radiofonico. Lo stile Newscaster è disponibile con le voci Matthew e Joanna, in inglese Stati Uniti (en-US), la voce Lupe in spagnolo Stati Uniti (es-US) e la voce Amy in inglese britannico (en-GB).

Per utilizzare lo stile Newscaster, scegliere innanzitutto il motore neurale e quindi utilizzare nel testo di input la sintassi descritta nelle fasi seguenti.

> **Note**
>
> - Per utilizzare qualsiasi stile di linguaggio neurale, è necessario utilizzare una delle regioni AWS che supportano le voci neurali. Questa opzione non è disponibile in tutte le regioni. Per ulteriori informazioni, consulta Compatibilità tra funzionalità e regioni.

Console

Per applicare lo stile Newscaster

1. Apri la console Amazon Polly all'indirizzo. https://console.aws.amazon.com/polly/
2. Assicurati di utilizzare una AWS regione in cui sono supportate le voci neurali.
3. Nella Text-to-Speech pagina, per Engine, scegli Neural.
4. Scegli la lingua e la voce che vuoi usare. Solo Matthew e Joanna per l'inglese Stati Uniti (en-US), Lupe per lo spagnolo Stati Uniti (es-US) e Amy per l'inglese britannico (en-GB) sono disponibili nella voce newscaster.
5. Attiva SSML.
6. Aggiungi testo di input alla tua text-to-speech richiesta utilizzando la sintassi SSML in stile Newscaster.

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

Ad esempio, è possibile utilizzare il tag newscaster come segue:

```
<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. Scegli Listen (Ascolta).

AWS CLI

Per applicare lo stile Newscaster

1. Nella richiesta API, includere il parametro del motore con il valore `neural`:

```
  --engine neural
```

2. Aggiungere il testo di input alla richiesta API utilizzando la sintassi SSML in stile Newscaster.

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

Ad esempio, è possibile utilizzare il tag newscaster come segue:

```
<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>
```

Per ulteriori informazioni su SSML, consulta Tag SSML supportati.

Ascoltare le voci

Dopo aver configurato Amazon Polly, puoi testare le voci utilizzando testo personalizzato sulla console.

Per ascoltare le voci di Amazon Polly sulla console

1. Accedi a AWS Management Console e apri la console Amazon Polly all'indirizzo. https://console.aws.amazon.com/polly/
2. Scegli la scheda Text-to-Speech (Sintesi vocale).
3. Per Engine, scegli Generative, Long Form, Neural o Standard.
4. Seleziona una lingua e una regione. Quindi scegli una voce.
5. Inserire il testo che si desidera venga pronunciato dalla voce o utilizzare la frase di default, quindi scegliere Listen (Ascolta).

Cronometraggio della velocità di una voce

A causa della naturale variazione tra le voci, ogni voce disponibile parla a velocità leggermente diverse. Ad esempio, con le voci in inglese americano, Ivy e Joanna sono leggermente più veloci di Matthew e notevolmente più veloci di Joey. Poiché le variazioni tra le voci sono così elevate, non è disponibile una velocità standard (parole al minuto) per le voci di Amazon Polly. Tuttavia, puoi scoprire quanto tempo impiega la tua voce a pronunciare il testo selezionato utilizzando Speech Marks.

Per cronometrare la durata di un passaggio di testo parlato

1. Aprire il AWS CLI.
2. Esegui il codice seguente, compilandolo se necessario.

```
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. Aprire `LengthOfText.txt`.

Se il testo fosse "Mary aveva un agnellino", le ultime righe restituite da Amazon Polly sarebbero le seguenti:

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

L'ultimo viseme, cioè il suono delle ultime lettere in "agnello", comincia 1082 millisecondi dopo l'inizio del discorso. Sebbene non sia esattamente la lunghezza dell'audio, è comunque prossima e può servire come base per il confronto tra le voci.

Modifica della velocità di una voce

In alcune applicazioni, potrebbe essere più indicato rallentare o velocizzare la voce scelta. Se la velocità della voce è un problema, Amazon Polly offre la possibilità di modificarla utilizzando i tag SSML. Ad esempio, se la tua organizzazione stava realizzando un'applicazione per leggere libri per un pubblico di immigrati, potresti voler variare la velocità della voce. Il tuo pubblico può parlare inglese, ma la loro padronanza è limitata. <prosody>Amazon Polly ti aiuta a rallentare la velocità della voce utilizzando il tag SSML.

Puoi usare una percentuale:

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

Oppure una velocità preimpostata:

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

Se si usa SSML con Amazon Polly, sono disponibili due opzioni di velocità:

- Velocità preimpostate:`x-slow`,`slow`, `mediumfast`, e. `x-fast` In questi casi, la velocità di ciascuna opzione è approssimativa, a seconda della voce preferita. L'opzione `medium` è la velocità normale della voce.

- n% della velocità vocale: è possibile utilizzare qualsiasi percentuale della velocità vocale, compresa tra il 20% e il 200%. In questi casi, è possibile scegliere esattamente la velocità desiderata. Tuttavia, la velocità reale della voce è approssimativa e dipende dalla voce scelta. 100% è consideratala velocità normale della voce.

> **Note**
>
> Metti alla prova la voce selezionata a varie velocità. La velocità di ciascuna opzione è approssimativa e dipende dalla voce scelta.

Per ulteriori informazioni sull'uso del `prosody` tag, consultaControllo del volume, della velocità di conversazione e dell'intonazione .

Lingue in Amazon Polly

Le lingue seguenti sono supportate da Amazon Polly e possono essere utilizzate per sintetizzare i discorsi. Ogni lingua ha un codice linguistico univoco. Questi codici di lingua sono tag di identificazione della lingua W3C (*ISO 639-3*per il nome della lingua e *ISO 3166* per il prefisso del paese).

Seleziona una lingua dalla tabella seguente per informazioni dettagliate sui fonemi e sui visemi forniti da Amazon Polly.

Lingua o	Codice lingua
Arabo	arb
Arabo (Golfo)	ar-AE
Catalan	ca-ES
Cinese (Cantone)	yue-CN
Cinese (Mandarino)	cmn-CN
Ceco	cs-CZ
Danese	da-DK
Olandese (belga)	nl-BE
Olandese	nl-NL

Lingua	Codice lingua
Inglese (Australia)	en-AU
Inglese (Regno Unito)	en-GB
Inglese (India)	en-IN
Inglese (Nuova Zelanda)	en-NZ
Inglese (singapiano)	it-SG
Inglese (Sud Africa)	en-ZA
Inglese (Stati Uniti)	it-IT
Inglese (Galles)	en-GB-WLS
Finlande	fi-FI
France	fr-FR

Lingua	Codice lingua
France (belga)	fr-BE
France (Canac	fr-CA
Hindi	hi-IN
Tedesc	de-DE
Tedesc (austria o)	de-AT
Tedesc (standa svizzer	de-CH
Islande	is-IS
Italiano	it-IT
Giappo e	ja-JP
Corear	ko-KR
Norveg	nb-NO
Polacc	pl-PL
Portogl e (Brasile	pt-BR

Lingua	Codice lingua
Portoghese (Europa)	pt-PT
Romeno	ro-RO
Russo	ru-RU
Spagnolo (Spagna)	es-ES
Spagnolo (messico)	es-MX
Spagnolo (Stati Uniti)	es-US
Svedese	sv-SE
Turco	tr-TR
Gallese	cy-GB

Arabo (arb)

La tabella seguente elenca i fonemi dell'alfabeto fonetico internazionale (International Phonetic Alphabet, IPA), i simboli X-SAMPA (Extended Speech Assessment Methods Phonetic Alphabet) e i corrispondenti visemi della voce araba di Zeina, supportata da Amazon Polly.

Tabella fonemi/visemi

IPA	X-SAMPA	Descrizione	Esempio	Visema
Consonanti				
ʔ	?	occlusiva glottidale sorda	أنا	
ʕ	?\	fricativa faringale sonora	عُمَر	k
b	b	occlusiva bilabiale sonora	بَلَد	p
d	d	occlusiva alveolare sonora	داري	t
dˤ	d_?\	occlusiva alveolare sonora enfatica	ضَوء	t
͡dʒ	dZ	affricata postalveolare sonora	جَميل	S
ð	D	fricativa dentale sonora	ذلِكَ	T
ðˤ	D_?\	fricativa dentale sonora enfatica	ظَلام	T
f	f	fricativa labiodentale sorda	فَصل	f
g	g	occlusiva velare sonora	إنترلج	k
ɣ	G	fricativa velare sonora	غَرب	k

IPA	X-SAMPA	Descrizione	Esempio	Visema
h	h	fricativa glottidale sorda	هذا	k
j	j	approssimante palatale	يَمشي	i
k	k	occlusiva velare sorda	كَلب	k
l	l	approssimante alveolare laterale	قال	t
lˤ	l_G	laterale alveolare approssimante enfatica	عبلاله	t
m	m	nasale bilabiale	ماذا	p
n	n	nasale alveolare	نور	t
p	p	occlusiva bilabiale sorda	حَبس	p
q	q	occlusiva uvulare sorda	قَريب	k
r	r	vibrante alveolare	رَمل	r
s	s	fricativa alveolare sorda	سُؤال	s
sˤ	s_?\	fricativa alveolare sorda enfatica	صاحب	s
ʃ	S	fricativa postalveo lare sorda	شُكر	S

IPA	X-SAMPA	Descrizione	Esempio	Visema
t	t	occlusiva alveolare sorda	تَمر	t
tˤ	t_?\	occlusiva alveolare sorda enfatica	طالِب	t
θ	T	fricativa dentale sorda	ثَلاث	T
v	v	fricativa labiodentale sonora	فيتامين	f
w	w	approssimante labiovelare	وَلَد	u
x	x	fricativa velare sorda	خَوْف	k
ħ	X\	fricativa faringale sorda	حَوْلَ	k
z	z	fricativa alveolare sonora	زُهور	s
Vocali				
a	a	vocale anteriore aperta non arrotondata	بَرد	a
aː	a:	vocale anteriore aperta non arrotondata allungata	دار	a

IPA	X-SAMPA	Descrizione	Esempio	Visema
ɑˤ	A_?\	vocale posteriore aperta non arrotondata enfatica	طَبل	a
ɑˤː	A_?\:	vocale posteriore aperta non arrotondata enfatica	ظالِم	a
u	u	vocale posteriore chiusa arrotondata	شُرب	u
uː	u:	vocale posteriore chiusa arrotondata allungata	سور	u
uˤ	u_?\	vocale posteriore chiusa arrotondata enfatica	بُدّ	u
uˤː	u_?\:	vocale posteriore chiusa arrotondata enfatica	طول	u
i	i	vocale anteriore chiusa non arrotondata	بِنت	i
iː	i:	vocale anteriore chiusa non arrotondata allungata	حَزين	i

IPA	X-SAMPA	Descrizione	Esempio	Visema
iˤ	i_?\	vocale anteriore chiusa non arrotondata enfatica	ضدّ	i
iˤː	i_?\:	vocale anteriore chiusa non arrotondata enfatica	ضامي	i
e	e	vocale anteriore semichiusa non arrotondata	تكرام	e
eː	e:	vocale anteriore semichiusa non arrotondata allungata	ليدوم	e
ɔ	O	vocale posterior e semiaperta arrotondata	تكونجي	O
ɔː	O:	vocale posterior e semiapert a arrotondata allungata	تلفيزون	O

Arabo (Golfo) (ar-AE)

La tabella seguente elenca i fonemi dell'alfabeto fonetico internazionale (International Phonetic Alphabet, IPA), i simboli X-SAMPA (Extended Speech Assessment Methods Phonetic Alphabet) e i corrispondenti visemi della voce araba di Hala, supportata da Amazon Polly.

Tabella fonemi/visemi

IPA	X-SAMPA	Descrizione	Esempio	Pronuncia	Visema
Consonanti					
b	b	occlusiva bilabiale sonora	دلب	/ " b a . l a d /	b
d	d	occlusiva alveolare sonora	رد	/ " r a d d /	d
dˤ	d_?\	Occlusiva alveolare sonora faringealizzata	ضوء	/ " d_?\ a w ? /	D
f	f	fricativa labiodentale sorda	نرف	/ " f I . r I n /	f
g	g	occlusiva velare sonora	لاق	/ " g a: l /	k
j	j	approssimante palatale sonora	يشمي	/ " j I m . S i: /	i
k	k	occlusiva velare sorda	لماك	/ " k a: . m i l /	k
l	l	laterale alveolare approssimante sonora	ليل	/ " l e: l /	t
lˤ	l_G	approssim ante alveolare	هللادبع	/ ?\ a b . " d A_?\ l_G . l_G A_?\ /	t

IPA	X-SAMPA	Descrizione	Esempio	Pronuncia	Visema
		sonora faringealizzata			
m	m	occlusiva nasale bilabiale	مئة	/ " m I j . j a /	p
n	n	occlusiva nasale alveolare	نور	/ " n u: r /	t
p	p	occlusiva bilabiale sorda	أوبرا	/ " ? O . p e . r a: /	p
q	q	occlusiva uvulare sorda	قرص	/ " q A_?\ s_?\ r /	k
r	r	vibrante alveolare	رمل	/ " r a . m I l /	r
s	s	fricativa alveolare sorda	سمسم	/ " s I m . s I m /	s
sˤ	s_?\	fricativa alveolare sorda faringealizzata	صاحب	/ " s_?\ A_?: . X \ I b /	s
t	t	occlusiva alveolare sorda	تمر	/ "t a . m a r /	t
tˤ	t_?\	fricativa alveolare sorda faringealizzata	طالب	/ " t_?\ A_?: . l I b /	t
v	v	fricativa labiodentale sonora	فيتامين	/ v i: . t A . " m i: n /	f

IPA	X-SAMPA	Descrizione	Esempio	Pronuncia	Visema
w	w	approssimante labiovelare sonora	وايد	/ " w a: . j I d /	u
x	x	fricativa velare sorda	خروف	/ x a . " r u: f /	k
z	z	fricativa velare sorda	زهور	/ " z h u: r /	s
ð	D	fricativa interdentale sonora	ذلك	/ " D a: . l I k /	D
ðˤ	D_?\	fricativa interdent ale sonora faringealizzata	ظام	/ D_?\ A_?\ . " I a: m /	D
ħ	X\	fricativa faringale sorda	الحين	/ ? a l . " X\ i: n /	k
ŋ	N	occlusiva nasale velare	هونغ كونغ	/ h O N . " k O N g /	k
ɣ	G	fricativa velare sonora	غريبة	/ G I . " r i: . b a /	k
ʃ	S	fricativa postalveolare sorda	شمس	/ " S a m s /	S
ʒ	Z	fricativa postalveolare sonora	جاكيت	/ Z a . " k e: t /	S

IPA	X-SAMPA	Descrizione	Esempio	Pronuncia	Visema
ʔ	?	occlusiva glottidale sorda	مؤسسة	/ m u . " ? a s . s a . s a /	
ʕ	?\	fricativa faringale sonora	عام	/ " ?\ a: m m /	k
dʒ	dZ	affricata postalveolare sonora	جامعة	/ " dZ a: m . ?\ a /	S
θ	T	fricativa interdentale sonora	ثلاثة	/ T a . " l a: . T a /	T
ɦ	h	fricativa glottale sonora	هلال	/ " h l a: l /	k
Vocali					
æ	a	vocale breve arrotonda ta anteriore semiaperta	سفر	/ " s a . f a r /	a
ɑˤ	A_?\	vocale breve non arrotonda ta posterior e aperta faringealizzata	صلب	/ " s_?\ A_?\ l b /	a
æː	a:	vocale lunga non arrotonda ta anteriore semiaperta	باب	/ " b a: b /	a

IPA	X-SAMPA	Descrizione	Esempio	Pronuncia	Visema
ɑˤː	A_?\:	vocale lunga non arrotondata posteriore aperta faringealizzata	جضان	/ " n A_?: . D_? \ i_?\ dZ /	a
a	A	vocale breve non arrotondata centrale aperta	wifi	/ " w A j . f A j /	a
i	i	vocale breve non arrotondata anteriore chiusa tesa (MSA)	إسحاق	/ ? i s . " X\ A_? \: q /	i
ɪ	I	vocale non arrotondata anteriore chiusa permissiva	تنب	/ " b I n t /	i
iˤ	i_?\	vocale breve non arrotondata anteriore chiusa faringealizzata	طفل	/ " t_?\ i_?\ f I l /	i
iː	i:	vocale lunga non arrotondata anteriore chiusa	سبيل	/ s a . " b i: l /	i

IPA	X-SAMPA	Descrizione	Esempio	Pronuncia	Visema
iˤː	i_?:	vocale lunga non arrotondata anteriore chiusa faringealizzata	رطيب	/ r A_?\ . " t_?\ i_?: b /	i
u	u	vocale breve arrotondata posteriore chiusa tesa (MSA)	مخترع	/ " m u x . t a . r i ?\ /	u
ʊ	U	vocale corta arrotondata posteriore chiusa permissiva	رسوم	/ r U . " s u: m /	u
uˤ	u_?\	vocale corta arrotondata posteriore chiusa faringealizzata	عصفور	/ ?\ u_?\ s_?\ . " f u: r /	u
u:	u:	vocale lunga arrotondata posteriore chiusa	توت	/ " t u: t /	u
uˤː	u_?\:	vocale lunga arrotondata posteriore chiusa faringealizzata	صور	/ " s_?\ u_?\: r /	u

IPA	X-SAMPA	Descrizione	Esempio	Pronuncia	Visema
e	e	vocale breve non arrotondata anteriore media	إِنْتَرْنِت	/ " s e n t /	e
e:	e:	vocale lunga non arrotondata anteriore media	إيش	/ " ? e: S /	e
ɔ	O	vocale corta arrotondata posteriore semiaperta	دولار	/ d O . " l A r /	O
ɔ:	O:	vocale lunga arrotondata posteriore semiaperta	لون	/ " l O: n /	O

Catalano (CA-ES)

La tabella seguente elenca i fonemi dell'alfabeto fonetico internazionale (International Phonetic Alphabet, IPA), i simboli X-SAMPA (Extended Speech Assessment Methods Phonetic Alphabet) e i corrispondenti visemi della voce catalana di Arlet, supportata da Amazon Polly.

Tabella fonemi/visemi

IPA	X-SAMPA	Descrizione	Esempio	Visema
Consonanti				
p	p	occlusiva bilabiale sorda	ploure	p

IPA	X-SAMPA	Descrizione	Esempio	Visema
t	t	occlusiva alveolare sorda	Tarragona	t
k	k	occlusiva velare sorda	com	k
b	b	occlusiva bilabiale sonora	bata	p
d	d	occlusiva alveolare sonora	endoll	t
g	g	occlusiva velare sonora	gros	k
m	m	nasale bilabiale sonora	manera	p
n	n	nasale alveolare sonora	donar	t
ɲ	J	nasale palatale sonora	any	J
ŋ	N	nasale velare sonora	pingüí	k
ɫ	5	sonora laterale alveolare approssimante velarizzata (dark l)	albercoc	l
ʎ	L	approssimante laterale palatale sonora	llop	J

IPA	X-SAMPA	Descrizione	Esempio	Visema
r	r	vibrante alveolare sonora	parra	r
ɾ	4	monovibrante alveolare sonora	para	t
f	f	fricativa labiodentale sorda	èmfasi	f
s	s	fricativa alveolare sorda	sac	s
z	z	fricativa alveolare sonora	calzes	s
ʃ	S	fricativa postalveolare sorda	guix	S
ʒ	Z	fricativa postalveolare sonora	col·legi	S
͡tʃ	tS	affricata postalveolare sorda	cotxe	S
͡dʒ	dZ	affricata postalveolare sonora	platja	S
β	B	approssimante bilabiale sonora	obert	B
ð	D	approssimante dentale sonora	bedoll	T
j	j	approssimante palatale sonora	noia	i
ɣ	G	approssimante velare sonora	pega	k

IPA	X-SAMPA	Descrizione	Esempio	Visema
v	v	fricativa labiodentale sonora	afgà	f
w	w	approssimante labiovelare sonora	aigua	u
x	x	fricativa velare sorda	Jiménez	k
ʝ	j\	fricativa palatale sonora	yeso	J
l	l	laterale alveolare approssimante sonora	alondra	t
θ	T	fricativa dentale sorda	González	T
Vocali				
a	a	vocale posteriore aperta	casa	a
e	e	vocale anteriore semichiusa non arrotondata	llenya	e
ɛ	E	vocale anteriore semiaperta non arrotondata	xec	E
i	i	vocale anteriore chiusa non arrotondata	visca	i

IPA	X-SAMPA	Descrizione	Esempio	Visema
o	o	vocale posterior e semichiusa arrotondata	gos	o
ɔ	O	vocale posterior e semiaperta arrotondata	joc	O
u	u	vocale posteriore chiusa arrotondata	un	u
ə	@	vocale centrale media	casa	@
Simboli aggiuntivi				
ˈ	"	enfasi principale	Alabama	
ˌ	%	enfasi secondaria	Alabama	
.	.	sillabazione	A.la.ba.ma	

Cinese (cantonese) (yue-CN)

La tabella seguente elenca i fonemi di Jyutping e dell'alfabeto fonetico internazionale (IPA) della voce cinese cantonese supportata da Amazon Polly. Il jyutping è un sistema di romanizzazione del cantonese comunemente usato nel mondo accademico e tra i parlanti cantonesi. I sistemi IPA e X-SAMPA non sono solitamente utilizzati, ma sono disponibili per il supporto inglese. I simboli IPA e X-SAMPA presenti nella tabella fungono solo da riferimento e non devono essere utilizzati per la trascrizione cinese. Vengono anche visualizzati esempi di Pinyin e i corrispondenti visemi.

Per fare in modo che Amazon Polly utilizzi la pronuncia fonetica con Jyutping, utilizza il tagphoneme `alphabet="x-amazon-jyutping"`.

I seguenti esempi mostrano questo per ogni standard.

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-SAMPA:

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

> **Note**
>
> Amazon Polly accetta solo l'input in cinese cantonese codificato in UTF-8.

Tabella fonemi/visemi

Jyutping	IPA	X-SAMPA	Descrizione	Esempio di jyutping	Visema
Consonanti					
b	p	p	occlusiva bilabiale sorda	巴, baa1	p
c	ts^h	ts_h	affricata alveolare sorda aspirata	叉, caa1	s
d	t	t	occlusiva alveolare sorda	打, daa2	t

Jyutping	IPA	X-SAMPA	Descrizione	Esempio di jyutping	Visema
f	f	f	fricativa labiodentale sorda	花, faa1	f
g	k	k	occlusiva velare sorda	家, gaa1	k
gw	k^w	k_w	occlusiva velare sorda labializzata	瓜, gwaa1	u
h	h	h	fricativa glottidale sorda	哈, haa1	k
k	k^h	k_h	Occlusiva velare sorda aspirata	卡, kaa1	k
kw	k^{wh}	k_wh	occlusiva velare labiale sorda aspirata	誇, kwaa1	u
l	l	l	approssimante alveolare laterale	啦, laa1	t
m	m	m	nasale bilabiale	媽, maa1	p
m	m	m=	nasale bilabiale sillabica	唔, m4	p
ng	ŋ	N	nasale velare	牙, ngaa4	k
ng	ŋ	N=	nasale velare sillabica	吳, ng4	k
n	n	n	nasale alveolare	拿, naa4	t
p	p^h	p_h	Occlusiva bilabiale sorda aspirata	趴, paa1	p
s	s	s	fricativa alveolare sorda	沙, saa1	s
t	t^h	t_h	Occlusiva alveolare sorda aspirata	他, taa1	t

Jyutping	IPA	X-SAMPA	Descrizione	Esempio di jyutping	Visema
w	w	w	approssimante labiovelare	娃, waa1	u
y	j	j	approssimante palatale	也, jaa5	i
z	ts	ts	affricata alveolare sorda	渣, zaa1	s
Vocali					
a	ɐ	6	vocale centrale quasi aperta	吉, gat1	a
aa	ɑ	A	vocale posteriore aperta non arrotondata	家, gaa1	a
aai	ɑi	Ai	dittongo	街, gaai1	a
aau	au	Au	dittongo	交, gaau1	a
ai	ɐi	6i	dittongo	雞, gai1	a
au	ɐu	6u	dittongo	溝, kau1	a
e	ɛ	E	vocale anteriore semiaperta non arrotondata	爹, de1	E
ei	ei	ei	dittongo	基, gei1	e
eo	ɵ	8	vocale centrale semichiusa arrotondata	春, ceon1	o
eoi	ey	8y	dittongo	居, geoi1	o
eu	ɛu	Eu	dittongo	掉 in 掉垃圾, deu6	E

Jyutping	IPA	X-SAMPA	Descrizione	Esempio di jyutping	Visema
i	i	i	vocale anteriore chiusa non arrotondata	斯, si1	i
i	I	I	vocale quasi anteriore quasi chiusa non arrotondata	激, gik1	i
iu	iu	iu	dittongo	驕, giu1	i
o	ɔ	O	vocale posteriore semiaperta arrotondata	哥, go1	O
oe	œ	9	vocale anteriore semiaperta arrotondata	鋸, goe3	O
oi	ɔi	Oi	dittongo	該, goi1	O
ou	ou	ou	dittongo	高, gou1	o
u	u	u	vocale posteriore chiusa arrotondata	姑, gu1	u
u	ʊ	U	vocale quasi posteriore quasi chiusa arrotondata	谷, guk5	u
ui	ui	ui	dittongo	劾, gui6	u
yu	y	y	vocale anteriore chiusa arrotondata	於, jyu1	u
Contrassegni tono e simboli aggiuntivi					
1			alto livello	詩, si1	
2			media ascesa	史, si2	
3			livello medio	試, si3	

Jyutping	IPA	X-SAMPA	Descrizione	Esempio di jyutping	Visema
4			livello molto basso	時, si4	
5			bassa ascesa	市, si5	
6			livello basso	是, si6	
-	.	.	sillabazione	語音 jyu5-jam1	

Cinese, mandarino (cmn-CN)

La tabella seguente elenca i fonemi di Pinyin e dell'alfabeto fonetico internazionale (IPA) della voce cinese mandarino supportata da Amazon Polly. Pinyin è lo standard internazionale per la romanizzazione standard del cinese. I sistemi IPA e X-SAMPA non sono solitamente utilizzati, ma sono disponibili per il supporto inglese. I simboli IPA e X-SAMPA presenti nella tabella fungono solo da riferimento e non devono essere utilizzati per la trascrizione cinese. Vengono anche visualizzati esempi di Pinyin e i corrispondenti visemi.

Per fare in modo che Amazon Polly utilizzi la pronuncia fonetica con Pinyin, utilizza il tag `phoneme alphabet="x-amazon-`*`phonetic standard used`*`"`.

I seguenti esempi mostrano questo per ogni standard.

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-SAMPA:

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

> **Note**
>
> Amazon Polly accetta solo l'input in cinese mandarino codificato in UTF-8. Lo standard di codifica GB 18030 non è attualmente supportato da Amazon Polly.

Tabella fonemi/visemi

Pinyin	IPA	X-SAMPA	Descrizione	Esempio di Pinyin	Visema
Consonanti					
f	f	f	fricativa labiodentale sorda	发, fa1	f
h	h	h	fricativa glottidale sorda	和, he2	k
g	k	k	occlusiva velare sorda	古, gu3	k
k	k^h	k_h	Occlusiva velare sorda aspirata	苦, ku3	k
l	l	l	approssimante alveolare laterale	拉, la1	t
m	m	m	nasale bilabiale	骂, ma4	p
n	n	n	nasale alveolare	那, na4	t
ng	ŋ	N	nasale velare	正, zheng4	k
b	p	p	occlusiva bilabiale sorda	爸, ba4	p

Pinyin	IPA	X-SAMPA	Descrizione	Esempio di Pinyin	Visema
p	pʰ	p_h	Occlusiva bilabiale sorda aspirata	怕, pa4	p
s	s	s	fricativa alveolare sorda	四, si4	s
x	ɕ	s\	fricativa alveolopalatale sorda	西, xi1	J
sh	ʂ	s`	fricativa retroflessa sorda	是, shi4	S
d	t	t	occlusiva alveolare sorda	打, da3	t
t	tʰ	t_h	Occlusiva alveolare sorda aspirata	他, ta1	t
zh	͡tʂ	t`s`	affricata retroflessa sorda	之, zhi1	S
ch	͡tʂʰ	t`s`_h	affricata retroflessa sorda aspirata	吃, chi1	S
s	͡ts	ts	affricata alveolare sorda	字, zi4	s
j	͡tɕ	ts\	affricata alveolopalatale sorda	鸡, ji1	J
q	͡tɕʰ	ts_h	affricata alveolopalatale sorda aspirata	七, qi1	J
c	͡tsʰ	ts_h	affricata alveolare sorda aspirata	次, ci4	s
w	w	w	approssimante labiovelare	我, wo3	u

Pinyin	IPA	X-SAMPA	Descrizione	Esempio di Pinyin	Visema
r	ʐ	z`	fricativa retroflessa sonora	日, ri4	S
sillabe rotiche "er" e "r"					
er	ɚ	@`	vocale centrale media variante rotica	二, er4	@
-r			sillaba rotica	馅儿, xianr4	@
Vocali					
e	ɤ	7	vocale posterior e semichiusa non arrotondata	恶, e4	e
e	ə	@	vocale centrale media	恩, en1	@
a	a	a	vocale anteriore aperta non arrotondata	安, an1	a
ai	aɪ	aI	dittongo	爱, ai4	a
ao	aʊ	aU	dittongo	奥, ao4	a
ei	eɪ	e	dittongo	诶, ei4	e
e	ɛ	E	vocale anteriore semiaperta non arrotondata	姐, jie3	E
i	i	i	vocale anteriore chiusa non arrotondata	鸡, ji1	i
ou	oʊ	oU	dittongo	欧, ou1	o

Pinyin	IPA	X-SAMPA	Descrizione	Esempio di Pinyin	Visema
o	ɔ	O	vocale posteriore semiaperta arrotondata	哦, o4	o
u	u	u	vocale posteriore chiusa arrotondata	主, zhu3	u
yu	y	y	vocale anteriore chiusa arrotondata	于, yu2	u
Contrassegni tono e simboli aggiuntivi					
1			tono alto	淤, yu1	
2			tono crescente	鱼, yu2	
3			tono basso (decrescente/crescente)	语, yu3	
4			tono decrescente	育, yu4	
0			tono neutro	的, de0	
-	.	.	sillabazione	语音 yu3-yin1	

Ceco (cs-CZ)

La tabella seguente elenca i fonemi dell'alfabeto fonetico internazionale (IPA), i simboli dell'alfabeto fonetico X-SAMPA (Extended Speech Assessment Methods) e i visemi corrispondenti per la voce ceca supportata da Amazon Polly.

Tabella fonemi/visemi

IPA	X-SAMPA	Descrizione	Esempio	Visema
Consonanti				

IPA	X-SAMPA	Descrizione	Esempio	Visema
p	p	occlusiva bilabiale sorda	p sì	p
t	t	occlusiva alveolare sorda	Va bene	t
c	c	Occlusiva palatale sorda	t'uk	J
k	k	occlusiva velare sorda	k è	k
b	b	occlusiva bilabiale sonora	di Kez	p
d	d	occlusiva alveolare sonora	d ok	t
ɟ	J\	occlusiva palatale sonora	d' era	J
g	g	occlusiva velare sonora	g um	k
f	f	fricativa labiodentale sorda	di un film	f
v	v	fricativa labiodentale sonora	v sì	f
s	s	fricativa alveolare sorda	è un	s
z	z	fricativa alveolare sonora	z el	s
ʃ	S	fricativa postalveolare sorda	s el	S

IPA	X-SAMPA	Descrizione	Esempio	Visema
ʒ	Z	fricativa postalveolare sonora	ž en	S
x	x	fricativa velare sorda	chat	k
ɦ	h	fricativa glottale sonora	h bus	k
ʦ	ts	affricata alveolare sorda	c o	s
ʧ	tS	affricata postalveolare sorda	č in	S
ʣ	dz	affricata alveolare sonora	špi c berský	s
ʤ	dZ	affricata alveolare sorda	dž in	S
m	m	nasale bilabiale	m o	p
n	n	nasale alveolare	in os	t
ɲ	J	nasale palatale	ň Wader	J
ŋ	N	nasale velare	bar n ka	k
r	r	vibrante alveolare sonora	O sì	r
r	r_r	trillo fricativo alveolare sonoro alzato	ř ez	r

IPA	X-SAMPA	Descrizione	Esempio	Visema
r	r_0_r	trillo fricativo alveolare rialzato silenzioso	ke ř	r
l	l	approssimante alveolare laterale	l sì	t
j	j	approssimante palatale	j en	i
w	w	approssimativo labiovelare	W Watson	u
r̩	r_=	trillo alveolare sillabico	k o k	r
l̩	l_ =	laterale alveolare approssimante sillabica	v l n	t
Vocali				
a	a	vocale anteriore aperta non arrotondata	l a n	a
a:	a:	vocale anteriore aperta non arrotondata allungata	l à n	a
ɛ	E	vocale anteriore semiaperta non arrotondata	l et	E

IPA	X-SAMPA	Descrizione	Esempio	Visema
ɛː	E:	vocale anteriore semiaperta non arrotondata allungata	l è t	E
ɪ	I	vocale quasi anteriore quasi chiusa non arrotondata	b i t	i
iː	i:	vocale anteriore chiusa non arrotondata allungata	b í t	i
o	o	vocale posterior e semichiusa arrotondata	h o l	o
oː	o:	vocale posterior e semichius a arrotondata allungata	g o l	o
u	u	vocale posteriore chiusa arrotondata	p u l	u
uː	u:	vocale posteriore chiusa arrotondata allungata	p u l	u
a#u	au	dittongo	au a	a
u	Eu	dittongo	eu ro	E
oru	ou	dittongo	Sto bene	o

IPA	X-SAMPA	Descrizione	Esempio	Visema
Simboli aggiuntivi				
ˈ	“	enfasi principale		
.	.	sillabazione		

Danese (da-DK)

La tabella seguente elenca i fonemi dell'alfabeto fonetico internazionale (International Phonetic Alphabet, IPA), i simboli X-SAMPA (Extended Speech Assessment Methods Phonetic Alphabet) e i corrispondenti visemi per la lingua danese supportati da Amazon Polly.

Tabella fonemi/visemi

IPA	X-SAMPA	Descrizione	Esempio	Visema
Consonanti				
b	b	occlusiva bilabiale sonora	bat	p
d	d	occlusiva alveolare sonora	da	t
ð	D	fricativa dentale sonora	mad, thriller	T
f	f	fricativa labiodentale sorda	fat	f
g	g	occlusiva velare sonora	gat	k
h	h	fricativa glottidale sorda	hat	k

IPA	X-SAMPA	Descrizione	Esempio	Visema
j	j	approssimante palatale	jo	i
k	k	occlusiva velare sorda	kat	k
l	l	approssimante alveolare laterale	ladt	t
m	m	nasale bilabiale	mat	p
n	n	nasale alveolare	nay	t
ŋ	N	nasale velare	lang	k
p	p	occlusiva bilabiale sorda	pande	p
r	r	vibrante alveolare	thriller, story	r
ʁ	R	fricativa uvulare sonora	rat	k
s	s	fricativa alveolare sorda	sat	s
t	t	occlusiva alveolare sorda	tal	t
v	v	fricativa labiodent ale sonora	vat	f
w	w	approssimante labiovelare sonora	hav, weekend	u
Vocali				

IPA	X-SAMPA	Descrizione	Esempio	Visema
ø	2	vocale anteriore semichiusa arrotondata	øst	o
ø:	2:	vocale anteriore semichiusa arrotondata allungata	øse	o
ɐ	6	vocale centrale quasi aperta	mor	a
œ	9	vocale anteriore semiaperta arrotondata	skøn, grønt	O
œ:	9:	vocale anteriore semiaperta arrotondata allungata	høne, gøre	O
ə	@	vocale centrale media	ane	@
æ:	{:	vocale anteriore quasi aperta non arrotondata allungata	male	a
a	a	vocale anteriore aperta non arrotondata	man	a
æ	{	vocale anteriore quasi aperta non arrotondata	adresse	a

IPA	X-SAMPA	Descrizione	Esempio	Visema
ɑ	A	vocale posterior e aperta non arrotondata	lak, tak	a
ɑ:	A:	vocale posterior e aperta non arrotondata allungata	rase	a
e	e	vocale anteriore semichiusa non arrotondata	midt	e
e:	e:	vocale anteriore semichiusa non arrotondata allungata	mele	e
ɛ	E	vocale anteriore semiaperta non arrotondata	mæt	E
ɛ:	E:	vocale anteriore semiaperta non arrotondata allungata	mæle	E
i	i	vocale anteriore chiusa non arrotondata	mit	i
i:	i:	vocale anteriore chiusa non arrotondata allungata	mile	i

IPA	X-SAMPA	Descrizione	Esempio	Visema
o	o	vocale posterior e semichiusa arrotondata	foto	o
oː	o:	vocale posterior e semichius a arrotondata allungata	mole	o
ɔ	O	vocale posterior e semiaperta arrotondata	mund	O
ɔː	O:	vocale posterior e semiapert a arrotondata allungata	måle	O
ɒː	D:	vocale posterior e semiapert a arrotondata allungata	morse	O
u	u	vocale posteriore chiusa arrotondata	lusk	u
uː	u:	vocale posteriore chiusa arrotondata allungata	mule	u
ʌ	V	posteriore semiaperta non arrotondata	kører	E
y	y	vocale anteriore chiusa arrotondata	yt	u

IPA	X-SAMPA	Descrizione	Esempio	Visema
y:	y:	vocale anteriore chiusa arrotondata allungata	hyle	u
Simboli aggiuntivi				
ˈ	"	enfasi principale	Alabama	
ˌ	%	enfasi secondaria	Alabama	
.	.	sillabazione	A.la.ba.ma	

Olandese (belga) (nl-BE)

La tabella seguente elenca i fonemi dell'alfabeto fonetico internazionale (International Phonetic Alphabet, IPA), i simboli X-SAMPA (Extended Speech Assessment Methods Phonetic Alphabet) e i corrispondenti visemi per la lingua olandese belga (fiamminga) supportati da Amazon Polly.

Tabella fonemi/visemi

IPA	X-SAMPA	Descrizione	Esempio	Visema
Consonanti				
b	b	occlusiva bilabiale sonora	bak	p
d	d	occlusiva alveolare sonora	dak	t
͡dʒ	dZ	affricata postalveolare sonora	manager	S
f	f	fricativa labiodentale sorda	fel	f

IPA	X-SAMPA	Descrizione	Esempio	Visema
g	g	occlusiva velare sonora	goal	k
ɣ	G	fricativa velare sonora	hoed	k
ɦ	h\	fricativa glottale sonora	hand	k
j	j	approssimante palatale	ja	i
k	k	occlusiva velare sorda	kap	k
l	l	approssimante alveolare laterale	land	t
m	m	nasale bilabiale	met	p
n	n	nasale alveolare	net	t
ŋ	N	nasale velare	bang	k
p	p	occlusiva bilabiale sorda	pak	p
r	r	vibrante alveolare	rand	r
s	s	fricativa alveolare sorda	sein	s
ʃ	S	fricativa postalveolare sorda	show	S
t	t	occlusiva alveolare sorda	tak	t

IPA	X-SAMPA	Descrizione	Esempio	Visema
v	v	fricativa labiodentale sonora	vel	f
ʋ	v\	approssimante labiodentale	wit	f
x	x	fricativa velare sorda	toch	k
z	z	fricativa alveolare sonora	ziin	s
ʒ	Z	fricativa postalveolare sonora	bagage	S
Vocali				
øː	2:	vocale anteriore semichiusa arrotondata allungata	neus	o
œy	9y	dittongo	buit	O
ə	@	vocale centrale media	de	@
aː	a:	vocale anteriore aperta non arrotondata allungata	baad	a
ɑː	A	vocale posteriore aperta non arrotondata	bad	a

IPA	X-SAMPA	Descrizione	Esempio	Visema
e:	e:	vocale anteriore semichiusa non arrotondata allungata	beet	e
ɜː	3:	vocale centrale semiaperta non arrotondata allungata	barrière	E
ɛ	E	vocale anteriore semiaperta non arrotondata	bed	E
ɛi	Ei	dittongo	beet	E
i	i	vocale anteriore chiusa non arrotondata	vier	i
ɪ	I	vocale quasi anteriore quasi chiusa non arrotondata	pit	i
o:	o:	vocale posterior e semichius a arrotondata allungata	boot	o
ɔ	O	vocale posterior e semiaperta arrotondata	pot	O
u	u	vocale posteriore chiusa arrotondata	hoed	u

IPA	X-SAMPA	Descrizione	Esempio	Visema
ʌu	Vu	dittongo	fout	E
yː	y:	vocale anteriore chiusa arrotondata allungata	fuut	u
ʏ	Y	vocale quasi anteriore quasi chiusa arrotondata	hut	u
Simboli aggiuntivi				
ˈ	"	enfasi principale	Alabama	
ˌ	%	enfasi secondaria	Alabama	
.	.	sillabazione	A.la.ba.ma	

Olandese (nl-NL)

La tabella seguente elenca i fonemi dell'alfabeto fonetico internazionale (International Phonetic Alphabet, IPA), i simboli X-SAMPA (Extended Speech Assessment Methods Phonetic Alphabet) e i corrispondenti visemi per la lingua olandese supportati da Amazon Polly.

Tabella fonemi/visemi

IPA	X-SAMPA	Descrizione	Esempio	Visema
Consonanti				
b	b	occlusiva bilabiale sonora	bak	p
d	d	occlusiva alveolare sonora	dak	t

IPA	X-SAMPA	Descrizione	Esempio	Visema
͡dʒ	dZ	affricata postalveolare sonora	manager	S
f	f	fricativa labiodentale sorda	fel	f
g	g	occlusiva velare sonora	goal	k
ɣ	G	fricativa velare sonora	hoed	k
ɦ	h\	fricativa glottale sonora	hand	k
j	j	approssimante palatale	ja	i
k	k	occlusiva velare sorda	kap	k
l	l	approssimante alveolare laterale	land	t
m	m	nasale bilabiale	met	p
n	n	nasale alveolare	net	t
ŋ	N	nasale velare	bang	k
p	p	occlusiva bilabiale sorda	pak	p
r	r	vibrante alveolare	rand	r
s	s	fricativa alveolare sorda	sein	s

IPA	X-SAMPA	Descrizione	Esempio	Visema
ʃ	S	fricativa postalveolare sorda	show	S
t	t	occlusiva alveolare sorda	tak	t
v	v	fricativa labiodentale sonora	vel	f
ʋ	v\	approssimante labiodentale	wit	f
x	x	fricativa velare sorda	toch	k
z	z	fricativa alveolare sonora	ziin	s
ʒ	Z	fricativa postalveolare sonora	bagage	S
Vocali				
øː	2:	vocale anteriore semichiusa arrotondata allungata	neus	o
œy	9y	dittongo	buit	O
ə	@	vocale centrale media	de	@
aː	a:	vocale anteriore aperta non arrotondata allungata	baad	a

IPA	X-SAMPA	Descrizione	Esempio	Visema
ɑː	A	vocale posterior e aperta non arrotondata	bad	a
eː	e:	vocale anteriore semichiusa non arrotondata allungata	beet	e
ɜː	3:	vocale centrale semiaperta non arrotondata allungata	barrière	E
ɛ	E	vocale anteriore semiaperta non arrotondata	bed	E
ɛi	Ei	dittongo	beet	E
i	i	vocale anteriore chiusa non arrotondata	vier	i
ɪ	I	vocale quasi anteriore quasi chiusa non arrotondata	pit	i
oː	o:	vocale posterior e semichius a arrotondata allungata	boot	o
ɔ	O	vocale posterior e semiaperta arrotondata	pot	O

IPA	X-SAMPA	Descrizione	Esempio	Visema
u	u	vocale posteriore chiusa arrotondata	hoed	u
ʌu	Vu	dittongo	fout	E
yː	y:	vocale anteriore chiusa arrotondata allungata	fuut	u
ʏ	Y	vocale quasi anteriore quasi chiusa arrotondata	hut	u
Simboli aggiuntivi				
ˈ	"	enfasi principale	Alabama	
ˌ	%	enfasi secondaria	Alabama	
.	.	sillabazione	A.la.ba.ma	

Inglese (Stati Uniti) (en-US)

La tabella seguente elenca i fonemi dell'alfabeto fonetico internazionale (International Phonetic Alphabet, IPA), i simboli X-SAMPA (Extended Speech Assessment Methods Phonetic Alphabet) e i corrispondenti visemi per la lingua inglese americano supportati da Amazon Polly.

Tabella fonemi/visemi

IPA	X-SAMPA	Descrizione	Esempio	Visema
Consonanti				
b	b	occlusiva bilabiale sonora	bed	p

IPA	X-SAMPA	Descrizione	Esempio	Visema
d	d	occlusiva alveolare sonora	dig	t
͡dʒ	dZ	affricata postalveolare sonora	jump	S
ð	D	fricativa dentale sonora	poi	T
f	f	fricativa labiodentale sorda	five	f
g	g	occlusiva velare sonora	game	k
h	h	fricativa glottidale sorda	house	k
j	j	approssimante palatale	sì	i
k	k	occlusiva velare sorda	cat	k
l	l	approssimante alveolare laterale	lay	l
m	m	nasale bilabiale	mouse	p
n	n	nasale alveolare	nap	t
ŋ	N	nasale velare	thing	k
p	p	occlusiva bilabiale sorda	speak	p
ɹ	r\	approssimante alveolare	red	r

IPA	X-SAMPA	Descrizione	Esempio	Visema
s	s	fricativa alveolare sorda	seem	s
ʃ	S	fricativa postalveolare sorda	ship	S
t	t	occlusiva alveolare sorda	trap	t
͡tʃ	tS	affricata postalveolare sorda	chart	S
θ	T	fricativa dentale sorda	thin	T
v	v	fricativa labiodentale sonora	vest	f
w	w	approssimante labiovelare sonora	west	u
z	z	fricativa alveolare sonora	zero	s
ʒ	Z	fricativa postalveolare sonora	vision	S
Vocali				
ə	@	vocale centrale media	arena	@
ɚ	@`	vocale centrale media variante rotica	reader	@

IPA	X-SAMPA	Descrizione	Esempio	Visema
æ	{	vocale anteriore quasi aperta non arrotondata	trap	a
aɪ	aI	dittongo	price	a
aʊ	aU	dittongo	mouth	a
ɑ	A	vocale posterior e aperta non arrotondata allungata	father	a
eɪ	eI	dittongo	face	e
ɝ	3`	vocale centrale semiaperta non arrotondata versione rotica	nurse	E
ɛ	E	vocale anteriore semiaperta non arrotondata	dress	E
i	i	vocale anteriore chiusa non arrotondata allungata	fleece	i
ɪ	I	vocale quasi anteriore quasi chiusa non arrotondata	kit	i
oʊ	oU	dittongo	goat	o

IPA	X-SAMPA	Descrizione	Esempio	Visema
ɔ	O	vocale posterior e semiapert a arrotondata allungata	thought	O
ɔɪ	OI	dittongo	choice	O
u	u	vocale posteriore chiusa arrotondata allungata	goose	u
ʊ	U	vocale quasi posteriore quasi chiusa arrotondata	foot	u
ʌ	V	open-mid-back vocale non arrotondata	strut	E
Simboli aggiuntivi				
ˈ	"	enfasi principale	Alabama	
ˌ	%	enfasi secondaria	Alabama	
.	.	sillabazione	A.la.ba.ma	

Inglese (Australia) (en-AU)

La tabella seguente elenca i fonemi dell'alfabeto fonetico internazionale (International Phonetic Alphabet, IPA), i simboli X-SAMPA (Extended Speech Assessment Methods Phonetic Alphabet) e i corrispondenti visemi per la lingua inglese australiano supportati da Amazon Polly.

Tabella fonemi/visemi

IPA	X-SAMPA	Descrizione	Esempio	Visema
Consonanti				
b	b	occlusiva bilabiale sonora	bed	p
d	d	occlusiva alveolare sonora	dig	t
͡dʒ	dZ	affricata postalveolare sonora	jump	S
ð	D	fricativa dentale sonora	poi	T
f	f	fricativa labiodentale sorda	five	f
g	g	occlusiva velare sonora	game	k
h	h	fricativa glottidale sorda	house	k
j	j	approssimante palatale	sì	i
k	k	occlusiva velare sorda	cat	k
l	l	approssimante alveolare laterale	lay	t
ˌl	l=	laterale alveolare approssimante sillabica	battle	t

IPA	X-SAMPA	Descrizione	Esempio	Visema
m	m	nasale bilabiale	mouse	p
m̩	m=	nasale bilabiale sillabica	anthem	p
n	n	nasale alveolare	nap	t
n̩	n=	nasale alveolare sillabica	button	t
ŋ	N	nasale velare	thing	k
p	p	occlusiva bilabiale sorda	pin	p
ɹ	r\	approssimante alveolare	red	r
s	s	fricativa alveolare sorda	seem	s
ʃ	S	fricativa postalveo lare sorda	ship	S
t	t	occlusiva alveolare sorda	task	t
͡tʃ	tS	affricata postalveo lare sorda	chart	S
θ	T	fricativa dentale sorda	thin	T
v	v	fricativa labiodent ale sonora	vest	f
w	w	approssimante labiovelare sonora	west	u

IPA	X-SAMPA	Descrizione	Esempio	Visema
z	z	fricativa alveolare sonora	zero	s
ʒ	Z	fricativa postalveolare sonora	vision	S
Vocali				
ə	@	vocale centrale media	arena	@
əʊ	@U	dittongo	goat	@
æ	{	vocale anteriore quasi aperta non arrotondata	trap	a
aɪ	aI	dittongo	price	a
aʊ	aU	dittongo	mouth	a
ɑː	A:	vocale posteriore aperta non arrotondata allungata	father	a
eɪ	eI	dittongo	face	e
ɜː	3:	vocale centrale semiaperta non arrotondata allungata	nurse	E
ɛ	E	vocale anteriore semiaperta non arrotondata	dress	E
ɛə	E@	dittongo	square	E

IPA	X-SAMPA	Descrizione	Esempio	Visema
iː	i	vocale anteriore chiusa non arrotondata allungata	fleece	i
ɪ	I	vocale quasi anteriore quasi chiusa non arrotondata	kit	i
ɪə	I@	dittongo	near	i
ɔː	OI	vocale posterior e semiapert a arrotondata allungata	thought	O
ɔɪ	OI	Dittongo	choice	O
ɒ	Q	vocale posteriore aperta arrotondata	lot	O
uː	uː	vocale posteriore chiusa arrotondata allungata	goose	u
ʊ	U	vocale quasi posteriore quasi chiusa arrotondata	foot	u
ʊə	U@	dittongo	cure	u
ʌ	V	Open-mid-back vocale non arrotondata	strut	E
Simboli aggiuntivi				

IPA	X-SAMPA	Descrizione	Esempio	Visema
ˈ	"	enfasi principale	Alabama	
ˌ	%	enfasi secondaria	Alabama	
.	.	sillabazione	A.la.ba.ma	

Inglese (Regno Unito) (en-GB)

La tabella seguente elenca i fonemi dell'alfabeto fonetico internazionale (International Phonetic Alphabet, IPA), i simboli X-SAMPA (Extended Speech Assessment Methods Phonetic Alphabet) e i corrispondenti visemi per la lingua inglese britannico supportati da Amazon Polly.

Tabella fonemi/visemi

IPA	X-SAMPA	Descrizione	Esempio	Visema
Consonanti				
b	b	occlusiva bilabiale sonora	bed	p
d	d	occlusiva alveolare sonora	dig	t
͡dʒ	dZ	affricata postalveolare sonora	jump	S
ð	D	fricativa dentale sonora	poi	T
f	f	fricativa labiodentale sorda	five	f
g	g	occlusiva velare sonora	game	k

IPA	X-SAMPA	Descrizione	Esempio	Visema
h	h	fricativa glottidale sorda	house	k
j	j	approssimante palatale	sì	i
k	k	occlusiva velare sorda	cat	k
l	l	approssimante alveolare laterale	lay	t
l̩	l=	laterale alveolare approssimante sillabica	battle	t
m	m	nasale bilabiale	mouse	p
m̩	m=	nasale bilabiale sillabica	anthem	p
n	n	nasale alveolare	nap	t
n̩	n=	nasale alveolare sillabica	button	t
ŋ	N	nasale velare	thing	k
p	p	occlusiva bilabiale sorda	pin	p
ɹ	r\	approssimante alveolare	red	r
s	s	fricativa alveolare sorda	seem	s

IPA	X-SAMPA	Descrizione	Esempio	Visema
ʃ	S	fricativa postalveolare sorda	ship	S
t	t	occlusiva alveolare sorda	task	t
͡tʃ	tS	affricata postalveolare sorda	chart	S
θ	T	fricativa dentale sorda	thin	T
v	v	fricativa labiodentale sonora	vest	f
w	w	approssimante labiovelare sonora	west	u
z	z	fricativa alveolare sonora	zero	s
ʒ	Z	fricativa postalveolare sonora	vision	S
Vocali				
ə	@	vocale centrale media	arena	@
əʊ	@U	dittongo	goat	@
æ	{	vocale anteriore quasi aperta non arrotondata	trap	a
aɪ	aI	dittongo	price	a
aʊ	aU	dittongo	mouth	a

IPA	X-SAMPA	Descrizione	Esempio	Visema
ɑː	A:	vocale posterior e aperta non arrotondata allungata	father	a
eɪ	eI	dittongo	face	e
ɜː	3:	vocale centrale semiaperta non arrotondata allungata	nurse	E
ɛ	E	vocale anteriore semiaperta non arrotondata	dress	E
ɛə	E@	dittongo	square	E
iː	i	vocale anteriore chiusa non arrotondata allungata	fleece	i
ɪ	I	vocale quasi anteriore quasi chiusa non arrotondata	kit	i
ɪə	I@	dittongo	near	i
ɔː	O:	vocale posterior e semiapert a arrotondata allungata	thought	O
ɔɪ	OI	Dittongo	choice	O

IPA	X-SAMPA	Descrizione	Esempio	Visema
ɒ	Q	vocale posteriore aperta arrotondata	lot	O
uː	u:	vocale posteriore chiusa arrotondata allungata	goose	u
ʊ	U	vocale quasi posteriore quasi chiusa arrotondata	foot	u
ʊə	U@	dittongo	cure	u
ʌ	V	Open-mid-back vocale non arrotondata	strut	E
Simboli aggiuntivi				
ˈ	"	enfasi principale	Alabama	
ˌ	%	enfasi secondaria	Alabama	
.	.	sillabazione	A.la.ba.ma	

Inglese (India) (en-IN)

La tabella seguente elenca i fonemi dell'alfabeto fonetico internazionale (International Phonetic Alphabet, IPA), i simboli X-SAMPA (Extended Speech Assessment Methods Phonetic Alphabet) e i corrispondenti visemi per la lingua inglese indiano supportati da Amazon Polly.

Per ulteriori fonemi utilizzati insieme all'inglese indiano, vedi Hindi (hi-IN).

Tabella fonemi/visemi

IPA	X-SAMPA	Descrizione	Esempio	Visema
Consonanti				
b	b	occlusiva bilabiale sonora	bed	p
d	d	occlusiva alveolare sonora	dig	t
͡dʒ	dZ	affricata postalveolare sonora	jump	S
ð	D	fricativa dentale sonora	poi	T
f	f	fricativa labiodentale sorda	five	f
g	g	occlusiva velare sonora	game	k
h	h	fricativa glottidale sorda	house	k
j	j	approssimante palatale	sì	i
k	k	occlusiva velare sorda	cat	k
l	l	approssimante alveolare laterale	lay	t
ˌl	l=	laterale alveolare approssimante sillabica	battle	t

IPA	X-SAMPA	Descrizione	Esempio	Visema
m	m	nasale bilabiale	mouse	p
m̩	m=	nasale bilabiale sillabica	anthem	p
n	n	nasale alveolare	nap	t
n̩	n=	nasale alveolare sillabica	nap	t
ŋ	N	nasale velare	thing	k
p	p	occlusiva bilabiale sorda	pin	p
ɹ	r\	approssimante alveolare	red	r
s	s	fricativa alveolare sorda	seem	s
ʃ	S	fricativa postalveolare sorda	ship	S
t	t	occlusiva alveolare sorda	task	t
͡tʃ	tS	affricata postalveolare sorda	chart	S
θ	T	fricativa dentale sorda	thin	T
v	v	fricativa labiodentale sonora	vest	f
w	w	approssimante labiovelare sonora	west	u

IPA	X-SAMPA	Descrizione	Esempio	Visema
z	z	fricativa alveolare sonora	zero	s
ʒ	Z	fricativa postalveolare sonora	vision	S
Vocali				
ə	@	vocale centrale media	arena	@
əʊ	@U	dittongo	goat	@
æ	{	vocale anteriore quasi aperta non arrotondata	trap	a
aɪ	aI	dittongo	price	a
aʊ	aU	dittongo	mouth	a
ɑː	A:	vocale posteriore aperta non arrotondata allungata	father	a
eɪ	eI	dittongo	face	e
ɜː	3:	vocale centrale semiaperta non arrotondata allungata	nurse	E
ɛ	E	vocale anteriore semiaperta non arrotondata	dress	E
ɛə	E@	dittongo	square	E

IPA	X-SAMPA	Descrizione	Esempio	Visema
iː	i	vocale anteriore chiusa non arrotondata allungata	fleece	i
ɪ	I	vocale quasi anteriore quasi chiusa non arrotondata	kit	i
ɪə	I@	dittongo	near	i
ɔː	OI	vocale posterior e semiapert a arrotondata allungata	thought	O
ɔɪ	OI	Dittongo	choice	O
ɒ	Q	vocale posteriore aperta arrotondata	lot	O
uː	uː	vocale posteriore chiusa arrotondata allungata	goose	u
ʊ	U	vocale quasi posteriore quasi chiusa arrotondata	foot	u
ʊə	U@	dittongo	cure	u
ʌ	V	Open-mid-back vocale non arrotondata	strut	E
Simboli aggiuntivi				

IPA	X-SAMPA	Descrizione	Esempio	Visema
ˈ	"	enfasi principale	Alabama	
ˌ	%	enfasi secondaria	Alabama	
.	.	sillabazione	A.la.ba.ma	

Inglese (Irlanda) (en-IE)

La tabella seguente elenca i fonemi dell'alfabeto fonetico internazionale (International Phonetic Alphabet, IPA), i simboli X-SAMPA (Extended Speech Assessment Methods Phonetic Alphabet) e i corrispondenti visemi per la lingua inglese irlandese supportati da Amazon Polly.

Tabella fonemi/visemi

IPA	X-SAMPA	Descrizione	Esempio	Visema
Consonanti				
b	b	occlusiva bilabiale sonora	bed	p
d	d	occlusiva alveolare sonora	dig	t
͡dʒ	dZ	affricata postalveolare sonora	jump	S
ð	D	fricativa dentale sonora	poi	T
f	f	fricativa labiodentale sorda	five	f
g	g	occlusiva velare sonora	game	k

IPA	X-SAMPA	Descrizione	Esempio	Visema
h	h	fricativa glottidale sorda	house	k
j	j	approssimante palatale	sì	i
k	k	occlusiva velare sorda	cat	k
l	l	approssimante alveolare laterale	lay	t
m	m	nasale bilabiale	mouse	p
n	n	nasale alveolare	nap	t
ŋ	N	nasale velare	thing	k
p	p	occlusiva bilabiale sorda	speak	p
ɹ	r\	approssimante alveolare	red	r
s	s	fricativa alveolare sorda	seem	s
ʃ	S	fricativa postalveolare sorda	ship	S
t	t	occlusiva alveolare sorda	trap	t
͡tʃ	tS	affricata postalveolare sorda	chart	S
θ	T	fricativa dentale sorda	thin	T

IPA	X-SAMPA	Descrizione	Esempio	Visema
v	v	fricativa labiodentale sonora	vest	f
w	w	approssimante labiovelare sonora	west	u
z	z	fricativa alveolare sonora	zero	s
ʒ	Z	fricativa postalveolare sonora	vision	S
Vocali				
ə	@	vocale centrale media	arena	@
ɚ	@`	vocale centrale media variante rotica	reader	@
æ	{	vocale anteriore quasi aperta non arrotondata	trap	a
aɪ	aI	dittongo	price	a
aʊ	aU	dittongo	mouth	a
ɑ	A	vocale posteriore aperta non arrotondata allungata	father	a
eɪ	eI	dittongo	face	e

IPA	X-SAMPA	Descrizione	Esempio	Visema
ɝ	3`	vocale centrale semiaperta non arrotondata versione rotica	nurse	E
ɛ	E	vocale anteriore semiaperta non arrotondata	dress	E
i	i	vocale anteriore chiusa non arrotondata allungata	fleece	i
ɪ	I	vocale quasi anteriore quasi chiusa non arrotondata	kit	i
oʊ	oU	dittongo	goat	o
ɔ	O	vocale posterior e semiapert a arrotondata allungata	thought	O
ɔɪ	OI	dittongo	choice	O
u	u	vocale posteriore chiusa arrotondata allungata	goose	u
ʊ	U	vocale quasi posteriore quasi chiusa arrotondata	foot	u

IPA	X-SAMPA	Descrizione	Esempio	Visema
ʌ	V	open-mid-back vocale non arrotondata	strut	E
Simboli aggiuntivi				
ˈ	"	enfasi principale	Alabama	
ˌ	%	enfasi secondaria	Alabama	
.	.	sillabazione	A.la.ba.ma	

Inglese (Nuova Zelanda) (en-NZ)

La tabella seguente elenca i fonemi dell'alfabeto fonetico internazionale (International Phonetic Alphabet, IPA), i simboli X-SAMPA (Extended Speech Assessment Methods Phonetic Alphabet) e i corrispondenti visemi per la lingua inglese neozelandese supportati da Amazon Polly.

Tabella fonemi/visemi

IPA	X-SAMPA	Descrizione	Esempio	Visema
Consonanti				
b	b	occlusiva bilabiale sonora	bed	p
d	d	occlusiva alveolare sonora	dig	t
͡dʒ	dZ	affricata postalveolare sonora	jump	S
ð	D	fricativa dentale sonora	poi	T

IPA	X-SAMPA	Descrizione	Esempio	Visema
f	f	fricativa labiodentale sorda	five	f
g	g	occlusiva velare sonora	game	k
h	h	fricativa glottidale sorda	house	k
j	j	approssimante palatale	sì	i
k	k	occlusiva velare sorda	cat	k
l	l	approssimante alveolare laterale	lay	t
l̩	l=	laterale alveolare approssimante sillabica	battle	t
m	m	nasale bilabiale	mouse	p
m̩	m=	nasale bilabiale sillabica	anthem	p
n	n	nasale alveolare	nap	t
n̩	n=	nasale alveolare sillabica	button	t
ŋ	N	nasale velare	thing	k
p	p	occlusiva bilabiale sorda	pin	p

IPA	X-SAMPA	Descrizione	Esempio	Visema
ɹ	r\	approssimante alveolare	red	r
s	s	fricativa alveolare sorda	seem	s
ʃ	S	fricativa postalveolare sorda	ship	S
t	t	occlusiva alveolare sorda	task	t
͡tʃ	tS	affricata postalveolare sorda	chart	S
Θ	T	fricativa dentale sorda	thin	T
v	v	fricativa labiodentale sonora	vest	f
w	w	approssimante labiovelare sonora	west	u
z	z	fricativa alveolare sonora	zero	s
ʒ	Z	fricativa postalveolare sonora	vision	S
Vocali				
ə	@	vocale centrale media	arena	@
əʊ	@U	dittongo	goat	@

IPA	X-SAMPA	Descrizione	Esempio	Visema
æ	{	vocale anteriore quasi aperta non arrotondata	trap	a
aɪ	aI	dittongo	price	a
aʊ	aU	dittongo	mouth	a
ɑː	A:	vocale posterior e aperta non arrotondata allungata	father	a
eɪ	eI	dittongo	face	e
ɜː	3:	vocale centrale semiaperta non arrotondata allungata	nurse	E
ɛ	E	vocale anteriore semiaperta non arrotondata	dress	E
ɛə	E@	dittongo	square	E
iː	i	vocale anteriore chiusa non arrotondata allungata	fleece	i
ɪ	I	vocale quasi anteriore quasi chiusa non arrotondata	kit	i
ɪə	I@	dittongo	near	i

IPA	X-SAMPA	Descrizione	Esempio	Visema
ɔː	O:	vocale posterior e semiapert a arrotondata allungata	thought	O
ɔɪ	OI	Dittongo	choice	O
ɒ	Q	vocale posteriore aperta arrotondata	lot	O
uː	u:	vocale posteriore chiusa arrotondata allungata	goose	u
ʊ	U	vocale quasi posteriore quasi chiusa arrotondata	foot	u
ʊə	U@	dittongo	cure	u
ʌ	V	Open-mid-back vocale non arrotondata	strut	E
Simboli aggiuntivi				
ˈ	"	enfasi principale	Alabama	
ˌ	%	enfasi secondaria	Alabama	
.	.	sillabazione	A.la.ba.ma	

La voce Aria parla inglese neozelandese e offre un supporto limitato per Maori. Può pronunciare le seguenti parole e frasi Maori. Le frasi Maori fanno distinzione tra maiuscole e minuscole.

Italiano	Maori
Ciao/saluti	Kia ora
Benvenuto in	Nau mai (ki)
Ciao (una persona)/grazie	Tēnā koe
Ciao (tre o più persone)/grazie	Tēnā koutou
Buongiorno	Ata mārie
Buongiorno	Mōrena
Grazie	Ngā mihi
Stai attento	Ngā manaakitanga
Ci vediamo	Ka kite
Ci vediamo dopo	Mā te wā
Buona giornata	Kia pai tō rā
Buon Natale	Meri Kirihimete
Maori	Māori
Lingua maori	te reo Māori
Settimana della lingua Maori	Te wiki o te reo Māori
Nuova Zelanda	Aotearoa
Capodanno Maori	Mātariki
Città in Nuova Zelanda/Il Waitangi Day è la giornata nazionale della Nuova Zelanda	Waitangi
One	tahi
Due	rua

Italiano	Maori
Tre	toru
Quattro	whā
Cinque	rima
Sei	ono
Sette	whitu
Otto	waru
Nove	iwa
Dieci	tekau
Venti	rua tekau
Trenta	Toru tekau

Inglese (singaporiano) (en-SG)

La tabella seguente elenca i fonemi dell'International Phonetic Alphabet (IPA), i simboli dell'Extended Speech Assessment Methods Phonetic Alphabet (X-SAMPA) e i visemi corrispondenti per le voci inglesi di Singapore supportate da Amazon Polly.

Tabella fonemi/visemi

IPA	X-SAMPA	Descrizione	Esempio	Visema
Consonanti				
b	b	occlusiva bilabiale sonora	bed	p
d	d	occlusiva alveolare sonora	dig	t

IPA	X-SAMPA	Descrizione	Esempio	Visema
͡dʒ	dZ	affricata postalveolare sonora	jump	S
ð	D	fricativa dentale sonora	poi	T
f	f	fricativa labiodentale sorda	five	f
g	g	occlusiva velare sonora	game	k
h	h	fricativa glottidale sorda	house	k
j	j	approssimante palatale	sì	i
k	k	occlusiva velare sorda	cat	k
l	l	approssimante alveolare laterale	lay	t
ˌl	l=	laterale alveolare approssimante sillabica	battle	t
m	m	nasale bilabiale	mouse	p
m̩	m=	nasale bilabiale sillabica	anthem	p
n	n	nasale alveolare	nap	t
n̩	n=	nasale alveolare sillabica	button	t

IPA	X-SAMPA	Descrizione	Esempio	Visema
ŋ	N	nasale velare	thing	k
p	p	occlusiva bilabiale sorda	pin	p
ɹ	r\	approssimante alveolare	red	r
s	s	fricativa alveolare sorda	seem	s
ʃ	S	fricativa postalveolare sorda	ship	S
t	t	occlusiva alveolare sorda	task	t
͡tʃ	tS	affricata postalveolare sorda	chart	S
Θ	T	fricativa dentale sorda	thin	T
v	v	fricativa labiodentale sonora	vest	f
w	w	approssimante labiovelare sonora	west	u
z	z	fricativa alveolare sonora	zero	s
ʒ	Z	fricativa postalveolare sonora	vision	S
Vocali				

IPA	X-SAMPA	Descrizione	Esempio	Visema
ə	@	vocale centrale media	arena	@
əʊ	@U	dittongo	goat	@
æ	{	vocale anteriore quasi aperta non arrotondata	trap	a
aɪ	aI	dittongo	price	a
aʊ	aU	dittongo	mouth	a
ɑː	A:	vocale posterior e aperta non arrotondata allungata	father	a
eɪ	eI	dittongo	face	e
ɜː	3:	vocale centrale semiaperta non arrotondata allungata	nurse	E
ɛ	E	vocale anteriore semiaperta non arrotondata	dress	E
ɛə	E@	dittongo	square	E
iː	i	vocale anteriore chiusa non arrotondata allungata	fleece	i

IPA	X-SAMPA	Descrizione	Esempio	Visema
ɪ	I	vocale quasi anteriore quasi chiusa non arrotondata	kit	i
ɪə	I@	dittongo	near	i
ɔː	O:	vocale posterior e semiapert a arrotondata allungata	thought	O
ɔɪ	OI	Dittongo	choice	O
ɒ	Q	vocale posteriore aperta arrotondata	lot	O
uː	u:	vocale posteriore chiusa arrotondata allungata	goose	u
ʊ	U	vocale quasi posteriore quasi chiusa arrotondata	foot	u
ʊə	U@	dittongo	cure	u
ʌ	V	Open-mid-back vocale non arrotondata	strut	E
Simboli aggiuntivi				
ˈ	"	enfasi principale	Alabama	
ˌ	%	enfasi secondaria	Alabama	
.	.	sillabazione	A.la.ba.ma	

Inglese (Sud Africa) (en-ZA)

La tabella seguente elenca i fonemi dell'alfabeto fonetico internazionale (International Phonetic Alphabet, IPA), i simboli X-SAMPA (Extended Speech Assessment Methods Phonetic Alphabet) e i corrispondenti visemi per la lingua inglese sudafricano supportati da Amazon Polly.

Tabella fonemi/visemi

IPA	X-SAMPA	Descrizione	Esempio	Visema
Consonanti				
b	b	occlusiva bilabiale sonora	bed	p
d	d	occlusiva alveolare sonora	dig	t
͡dʒ	dZ	affricata postalveolare sonora	jump	S
ð	D	fricativa dentale sonora	poi	T
f	f	fricativa labiodentale sorda	five	f
g	g	occlusiva velare sonora	game	k
h	h	fricativa glottidale sorda	house	k
j	j	approssimante palatale	sì	i
k	k	occlusiva velare sorda	cat	k

IPA	X-SAMPA	Descrizione	Esempio	Visema
l	l	approssimante alveolare laterale	lay	t
l̩	l=	laterale alveolare approssimante sillabica	battle	t
ɬ̩	K	fricativa laterale sorda	umhlanga	t
m	m	nasale bilabiale	mouse	p
m̩	m=	nasale bilabiale sillabica	anthem	p
n	n	nasale alveolare	nap	t
n̩	n=	nasale alveolare sillabica	button	t
ŋ	N	nasale velare	thing	k
p	p	occlusiva bilabiale sorda	pin	p
ɹ	r\	approssimante alveolare	red	r
r	r	vibrante alveolare	pareis	r
s	s	fricativa alveolare sorda	seem	s
ʃ	S	fricativa postalveo lare sorda	ship	S
t	t	occlusiva alveolare sorda	task	t

IPA	X-SAMPA	Descrizione	Esempio	Visema
͡tʃ	tS	affricata postalveolare sorda	chart	S
θ	T	fricativa dentale sorda	thin	T
v	v	fricativa labiodentale sonora	vest	f
w	w	approssimante labiovelare sonora	west	u
x	x	fricativa velare sorda	gauteng	k
z	z	fricativa alveolare sonora	zero	s
!	!\	clic post-alveolare	gqeberha	k
|	|\	clic dentale	nctubo	t
||	||\	clic laterale	xhosa	t
Vocali				
ə	@	vocale centrale media	arena	@
əi	@i	dittongo	nelspr uit	i
əʊ	@U	dittongo	goat	@
æ	{	vocale anteriore quasi aperta non arrotondata	trap	a
aɪ	aI	dittongo	price	a

IPA	X-SAMPA	Descrizione	Esempio	Visema
aʊ	aU	dittongo	mouth	a
ɑː	A:	vocale posterior e aperta non arrotondata allungata	father	a
eɪ	eI	dittongo	face	e
ɜː	3:	vocale centrale semiaperta non arrotondata allungata	nurse	E
ɛ	E	vocale anteriore semiaperta non arrotondata	dress	E
ɛə	E@	dittongo	square	E
iː	i	vocale anteriore chiusa non arrotondata allungata	fleece	i
iə	I@	dittongo	du preez	i
ɪ	I	vocale quasi anteriore quasi chiusa non arrotondata	kit	i
ɪə	I@	dittongo	near	i

IPA	X-SAMPA	Descrizione	Esempio	Visema
ɔː	O:	vocale posterior e semiapert a arrotondata allungata	thought	O
ɔɪ	OI	Dittongo	choice	O
ɒ	Q	vocale posteriore aperta arrotondata	lot	O
uː	u:	vocale posteriore chiusa arrotondata allungata	goose	u
ʊ	U	vocale quasi posteriore quasi chiusa arrotondata	foot	u
ʊə	U@	dittongo	cure	u
ʌ	V	Open-mid-back vocale non arrotondata	strut	E
y	y	vocale anteriore chiusa arrotondata	van vuuren	u
Simboli aggiuntivi				
ˈ	"	enfasi principale	Alabama	
ˌ	%	enfasi secondaria	Alabama	
.	.	sillabazione	A.la.ba.ma	

Inglese (Galles) (en-GB-WLS)

La tabella seguente elenca i fonemi dell'alfabeto fonetico internazionale (International Phonetic Alphabet, IPA), i simboli X-SAMPA (Extended Speech Assessment Methods Phonetic Alphabet) e i corrispondenti visemi per la lingua inglese gallese supportati da Amazon Polly.

Tabella fonemi/visemi

IPA	X-SAMPA	Descrizione	Esempio	Visema
Consonanti				
b	b	occlusiva bilabiale sonora	bed	p
d	d	occlusiva alveolare sonora	dig	t
͡dʒ	dZ	affricata postalveolare sonora	jump	S
ð	D	fricativa dentale sonora	poi	T
f	f	fricativa labiodentale sorda	five	f
g	g	occlusiva velare sonora	game	k
h	h	fricativa glottidale sorda	house	k
j	j	approssimante palatale	sì	i
k	k	occlusiva velare sorda	cat	k

IPA	X-SAMPA	Descrizione	Esempio	Visema
l	l	approssimante alveolare laterale	lay	t
l̩	l=	laterale alveolare approssimante sillabica	battle	t
m	m	nasale bilabiale	mouse	p
m̩	m=	nasale bilabiale sillabica	anthem	p
n	n	nasale alveolare	nap	t
n̩	n=	nasale alveolare sillabica	nap	t
ŋ	N	nasale velare	thing	k
p	p	occlusiva bilabiale sorda	pin	p
ɹ	r\	approssimante alveolare	red	r
s	s	fricativa alveolare sorda	seem	s
ʃ	S	fricativa postalveo lare sorda	ship	S
t	t	occlusiva alveolare sorda	task	t
͡tʃ	tS	affricata postalveo lare sorda	chart	S

IPA	X-SAMPA	Descrizione	Esempio	Visema
Θ	T	fricativa dentale sorda	thin	T
v	v	fricativa labiodentale sonora	vest	f
w	w	approssimante labiovelare sonora	west	u
z	z	fricativa alveolare sonora	zero	s
ʒ	Z	fricativa postalveolare sonora	vision	S
Vocali				
ə	@	vocale centrale media	arena	@
əʊ	@U	dittongo	goat	@
æ	{	vocale anteriore quasi aperta non arrotondata	trap	a
aɪ	aI	dittongo	price	a
aʊ	aU	dittongo	mouth	a
ɑː	A:	vocale posteriore aperta non arrotondata allungata	father	a
eɪ	eI	dittongo	face	e

IPA	X-SAMPA	Descrizione	Esempio	Visema
ɜː	3:	vocale centrale semiaperta non arrotondata allungata	nurse	E
ɛ	E	vocale anteriore semiaperta non arrotondata	dress	E
ɛə	E@	dittongo	square	E
iː	i	vocale anteriore chiusa non arrotondata allungata	fleece	i
ɪ	I	vocale quasi anteriore quasi chiusa non arrotondata	kit	i
ɪə	I@	dittongo	near	i
ɔː	OI	vocale posterior e semiapert a arrotondata allungata	thought	O
ɔɪ	OI	Dittongo	choice	O
ɒ	Q	vocale posteriore aperta arrotondata	lot	O
uː	u:	vocale posteriore chiusa arrotondata allungata	goose	u

IPA	X-SAMPA	Descrizione	Esempio	Visema
ʊ	U	vocale quasi posteriore quasi chiusa arrotondata	foot	u
ʊə	U@	dittongo	cure	u
ʌ	V	Open-mid-back vocale non arrotondata	strut	E
Simboli aggiuntivi				
ˈ	"	enfasi principale	Alabama	
ˌ	%	enfasi secondaria	Alabama	
.	.	sillabazione	A.la.ba.ma	

Finlandese (fi-FI)

La tabella seguente elenca i fonemi dell'alfabeto fonetico internazionale (International Phonetic Alphabet, IPA), i simboli X-SAMPA (Extended Speech Assessment Methods Phonetic Alphabet) e i corrispondenti visemi della voce finlandese supportata da Amazon Polly.

Tabella fonemi/visemi

IPA	X-SAMPA	Descrizione	Esempio	Visema
Consonanti finlandesi				
p	p	occlusiva bilabiale sorda	[p]ankki	p
t	t	occlusiva alveolare sorda	[t]alo	t

IPA	X-SAMPA	Descrizione	Esempio	Visema
k	k	occlusiva velare sorda	[k]aali	k
d	d	occlusiva alveolare sonora	[d]ata	t
s	s	fricativa alveolare sorda	[s]ali	s
h	h	fricativa glottidale sorda	[h]attu	k
ʋ	v\	approssimante labiodentale sonora	[v]aivá	v
j	j	approssimante palatale	[j]oki	i
l	l	approssimante alveolare laterale	[l]oma	t
r	r	vibrante alveolare sonora	[r]iita	r
m	m	nasale bilabiale	[m]ato	p
n	n	nasale alveolare	[n]enäa	t
ŋ	N	nasale velare	he[n]ki	k
Consonanti presenti nei prestiti linguistici				
b	b	occlusiva bilabiale sonora	[b]ussi	p
f	f	fricativa labiodent ale sorda	[f]irma	v

IPA	X-SAMPA	Descrizione	Esempio	Visema
w	w	approssimante labiovelare sonora	[w]iki	u
z	z	fricativa alveolare sonora	[z] ulu	s
g	g	occlusiva velare sonora	[g]aala	k
ʃ	S	fricativa postalveolare sorda	[sh] akki	S
ʒ	Z	fricativa postalveolare sonora	[g]enre	S
θ	T	fricativa dentale sorda	ear[th]	T
ð	D	fricativa dentale sonora	ei[th]er	T
Vocali brevi				
i	i	vocale anteriore chiusa non arrotondata	k[i]lo	i
ɛ	E	vocale anteriore semiaperta non arrotondata	k[e]sä	E
æ	{	vocale anteriore quasi aperta non arrotondata	[kä] ly	A
y	y	vocale anteriore chiusa arrotondata	k[y]lä	u

IPA	X-SAMPA	Descrizione	Esempio	Visema
ø	2	vocale arrotondata anteriore-media chiusa	p[ö]ly	O
u	u	vocale posteriore chiusa arrotondata	k[u]lo	u
ɔ	O	vocale arrotondata posteriore-media aperta	k[o]lo	O
ɑ	A	vocale posteriore aperta non arrotondata	k[a]la	A
Vocali lunghe				
iː	iː	vocale anteriore chiusa non arrotondata allungata	s[ii]li	i
ɛː	E:	vocale non arrotondata anteriore-media aperta lunga	[ee]tu	E
æː	{:	vocale non arrotondata anteriore-aperta vicina lunga	t[ää]llä	A
yː	y:	vocale anteriore chiusa non arrotondata allungata	t[yy]li	u

IPA	X-SAMPA	Descrizione	Esempio	Visema
øː	2:	vocale arrotondata anteriore-media chiusa lunga	t[öö]lö	O
uː	u:	vocale posteriore chiusa arrotondata allungata	t[uu]li	u
ɔː	O:	vocale posteriore semiaperta arrotondata allungata	r[oo]li	O
ɑː	A:	vocale posteriore aperta non arrotondata allungata	k[aa]su	A
Dittonghi				
ɛi	Ei	dittongo	l[ei]pä	E
æi	{i	dittongo	[äi]ti	A
ui	ui	dittongo	k[ui]n	u
ɑi	Ai	dittongo	k[ai]kki	A
ɔi	Oi	dittongo	p[oi]ka	O
øi	2i	dittongo	s[öi]n	O
yi	yi	dittongo	l[yi]jy	u
au	Au	dittongo	s[au]na	A
ɔu	Ou	dittongo	k[ou]lu	O

IPA	X-SAMPA	Descrizione	Esempio	Visema
ɛu	Eu	dittongo	r[eu]na	E
iu	iu	dittongo	v[iu]lu	i
æy	{y	dittongo	t[äy]nnä	A
øy	2y	dittongo	k[öy]hä	O
ɛy	Ey	dittongo	pes[ey]tyä	E
iy	iy	dittongo	käär[iy]tyä	i
iɛ	iE	dittongo	t[ie]	i
yø	y2	dittongo	[yö]	u
uɔ	uO	dittongo	t[uo]	u
Vocali trovate nei prestiti linguistici inglesi				
ɪ	I	vocale quasi anteriore quasi chiusa non arrotondata	b[i]t	i
ʊ	U	vocale quasi posteriore quasi chiusa arrotondata	b[oo]k	u
ə	@	vocale centrale media	[a]bout	@
ʌ	V	open-mid-back vocale non arrotondata	c[u]t	E

Francese (fr-FR)

La tabella seguente elenca i fonemi dell'alfabeto fonetico internazionale (International Phonetic Alphabet, IPA), i simboli X-SAMPA (Extended Speech Assessment Methods Phonetic Alphabet) e i corrispondenti visemi per la lingua francese supportati da Amazon Polly.

Tabella fonemi/visemi

IPA	X-SAMPA	Descrizione	Esempio	Visema
Consonanti				
b	b	occlusiva bilabiale sonora	boire	p
d	d	occlusiva alveolare sonora	madame	t
f	f	fricativa labiodentale sorda	femme	f
g	g	occlusiva velare sonora	grand	k
ɥ	H	approssimante labio-palatale	bruit	u
j	j	approssimante palatale	meilleur	i
k	k	occlusiva velare sorda	quatre	k
l	l	approssimante alveolare laterale	malade	t
m	m	nasale bilabiale	maison	p
n	n	nasale alveolare	astronome	t
ɲ	J	nasale palatale	baigner	J

IPA	X-SAMPA	Descrizione	Esempio	Visema
ŋ	N	nasale velare	parking	k
p	p	occlusiva bilabiale sorda	pomme	p
ʁ	R	fricativa uvulare sonora	amoureux	k
s	s	fricativa alveolare sorda	santé	s
ʃ	S	fricativa postalveolare sorda	chat	S
t	t	occlusiva alveolare sorda	téléphone	t
v	v	fricativa labiodentale sonora	vrai	f
w	w	approssimante labiovelare sonora	soir	u
z	z	fricativa alveolare sonora	raison	s
ʒ	Z	fricativa postalveolare sonora	aubergine	S
Vocali				
ø	2	vocale anteriore semichiusa arrotondata	deux	o
œ	9	vocale anteriore semiaperta arrotondata	neuf	O

IPA	X-SAMPA	Descrizione	Esempio	Visema
œ̃	9~	vocale anteriore semiaperta arrotondata nasale	brun	O
ə	@	vocale centrale media	je	@
a	a	vocale anteriore aperta non arrotondata	table	a
ɑ̃	A~	vocale posterior e aperta non arrotondata nasale	camembert	a
e	e	vocale anteriore semichiusa non arrotondata	marché	e
ɛ	E	vocale anteriore semiaperta non arrotondata	neige	E
ɛ̃	E~	vocale anteriore semiaperta non arrotondata nasale	sapin	E
i	i	vocale anteriore chiusa non arrotondata	mille	i
o	o	vocale posterior e semichiusa arrotondata	hôpital	o

IPA	X-SAMPA	Descrizione	Esempio	Visema
ɔ	O	vocale posterior e semiaperta arrotondata	homme	O
ɔ̃	O~	vocale posterior e semiaperta arrotondata nasale	bon	O
u	u	vocale posteriore chiusa arrotondata	sous	u
y	y	vocale anteriore chiusa arrotondata	dur	u
Simboli aggiuntivi				
ˈ	"	enfasi principale	Alabama	
ˌ	%	enfasi secondaria	Alabama	
.	.	sillabazione	A.la.ba.ma	

Francese (belga) (fr-BE)

La tabella seguente elenca i fonemi dell'alfabeto fonetico internazionale (International Phonetic Alphabet, IPA), i simboli X-SAMPA (Extended Speech Assessment Methods Phonetic Alphabet) e i corrispondenti visemi per la lingua francese belga supportati da Amazon Polly.

Tabella fonemi/visemi

IPA	X-SAMPA	Descrizione	Esempio	Visema
Consonanti				
b	b	occlusiva bilabiale sonora	boire	p

IPA	X-SAMPA	Descrizione	Esempio	Visema
d	d	occlusiva alveolare sonora	madame	t
f	f	fricativa labiodentale sorda	femme	f
g	g	occlusiva velare sonora	grand	k
ɥ	H	approssimante labio-palatale	bruit	u
j	j	approssimante palatale	meilleur	i
k	k	occlusiva velare sorda	quatre	k
l	l	approssimante alveolare laterale	malade	t
m	m	nasale bilabiale	maison	p
n	n	nasale alveolare	astronome	t
ɲ	J	nasale palatale	baigner	J
ŋ	N	nasale velare	parking	k
p	p	occlusiva bilabiale sorda	pomme	p
ʁ	R	fricativa uvulare sonora	amoureux	k
s	s	fricativa alveolare sorda	santé	s

IPA	X-SAMPA	Descrizione	Esempio	Visema
ʃ	S	fricativa postalveolare sorda	chat	S
t	t	occlusiva alveolare sorda	téléphone	t
v	v	fricativa labiodentale sonora	vrai	f
w	w	approssimante labiovelare sonora	soir	u
z	z	fricativa alveolare sonora	raison	s
ʒ	Z	fricativa postalveolare sonora	aubergine	S
Vocali				
ø	2	vocale anteriore semichiusa arrotondata	deux	o
œ	9	vocale anteriore semiaperta arrotondata	neuf	O
œ̃	9~	vocale anteriore semiaperta arrotondata nasale	brun	O
ə	@	vocale centrale media	je	@

IPA	X-SAMPA	Descrizione	Esempio	Visema
a	a	vocale anteriore aperta non arrotondata	table	a
ɑ̃	A~	vocale posterior e aperta non arrotondata nasale	camembert	a
e	e	vocale anteriore semichiusa non arrotondata	marché	e
ɛ	E	vocale anteriore semiaperta non arrotondata	neige	E
ɛ̃	E~	vocale anteriore semiaperta non arrotondata nasale	sapin	E
i	i	vocale anteriore chiusa non arrotondata	mille	i
o	o	vocale posterior e semichiusa arrotondata	hôpital	o
ɔ	O	vocale posterior e semiaperta arrotondata	homme	O
ɔ̃	O~	vocale posterior e semiaperta arrotondata nasale	bon	O

IPA	X-SAMPA	Descrizione	Esempio	Visema
u	u	vocale posteriore chiusa arrotondata	sous	u
y	y	vocale anteriore chiusa arrotondata	dur	u
Simboli aggiuntivi				
ˈ	"	enfasi principale	Alabama	
ˌ	%	enfasi secondaria	Alabama	
.	.	sillabazione	A.la.ba.ma	

Francese (Canada) (fr-CA)

La tabella seguente elenca i fonemi dell'alfabeto fonetico internazionale (International Phonetic Alphabet, IPA), i simboli X-SAMPA (Extended Speech Assessment Methods Phonetic Alphabet) e i corrispondenti visemi per la lingua francese canadese supportati da Amazon Polly.

Tabella fonemi/visemi

IPA	X-SAMPA	Descrizione	Esempio	Visema
Consonanti				
b	b	occlusiva bilabiale sonora	boire	p
d	d	occlusiva alveolare sonora	madame	t
f	f	fricativa labiodentale sorda	femme	f
g	g	occlusiva velare sonora	grand	k

IPA	X-SAMPA	Descrizione	Esempio	Visema
ɥ	H	approssimante labio-palatale	bruit	u
j	j	approssimante palatale	meilleur	i
k	k	occlusiva velare sorda	quatre	k
l	l	approssimante alveolare laterale	malade	t
m	m	nasale bilabiale	maison	p
n	n	nasale alveolare	astronome	t
ɲ	J	nasale palatale	baigner	J
ŋ	N	nasale velare	parking	k
p	p	occlusiva bilabiale sorda	pomme	p
ʁ	R	fricativa uvulare sonora	amoureux	k
s	s	fricativa alveolare sorda	santé	s
ʃ	S	fricativa postalveo lare sorda	chat	S
t	t	occlusiva alveolare sorda	téléphone	t
v	v	fricativa labiodent ale sonora	vrai	f

IPA	X-SAMPA	Descrizione	Esempio	Visema
w	w	approssimante labiovelare sonora	soir	u
z	z	fricativa alveolare sonora	raison	s
ʒ	Z	fricativa postalveolare sonora	aubergine	S
Vocali				
ø	2	vocale anteriore semichiusa arrotondata	deux	o
œ	9	vocale anteriore semiaperta arrotondata	neuf	O
œ̃	9~	vocale anteriore semiaperta arrotondata nasale	brun	O
ə	@	vocale centrale media	je	@
a	a	vocale anteriore aperta non arrotondata	table	a
ɑ̃	A~	vocale posteriore aperta non arrotondata nasale	camembert	a
e	e	vocale anteriore semichiusa non arrotondata	marché	e

IPA	X-SAMPA	Descrizione	Esempio	Visema
ɛ	E	vocale anteriore semiaperta non arrotondata	neige	E
ɛ̃	E~	vocale anteriore semiaperta non arrotondata nasale	sapin	E
i	i	vocale anteriore chiusa non arrotondata	mille	i
o	o	vocale posterior e semichiusa arrotondata	hôpital	o
ɔ	O	vocale posterior e semiaperta arrotondata	homme	O
ɔ̃	O~	vocale posterior e semiaperta arrotondata nasale	bon	O
u	u	vocale posteriore chiusa arrotondata	sous	u
y	y	vocale anteriore chiusa arrotondata	dur	u
Simboli aggiuntivi				
ˈ	"	enfasi principale	Alabama	
ˌ	%	enfasi secondaria	Alabama	
.	.	sillabazione	A.la.ba.ma	

Tedesco (de-DE)

La tabella seguente elenca i fonemi dell'alfabeto fonetico internazionale (International Phonetic Alphabet, IPA), i simboli X-SAMPA (Extended Speech Assessment Methods Phonetic Alphabet) e i corrispondenti visemi per la lingua tedesca supportati da Amazon Polly.

Tabella fonemi/visemi

IPA	X-SAMPA	Descrizione	Esempio	Visema
Consonanti				
ʔ	?	occlusiva glottidale sorda		
b	b	occlusiva bilabiale sonora	Bier	p
d	d	occlusiva alveolare sonora	Dach	t
ç	C	fricativa palatale sorda	ich	k
͡dʒ	dZ	affricata postalveolare sonora	Dschungel	S
f	f	Fricativa labiodentale sorda	Vogel	f
g	g	Occlusiva velare sonora	Gabel	k
h	h	Fricativa faringale sorda	Haus	k
j	j	Fricativa faringale sorda	jemand	i

IPA	X-SAMPA	Descrizione	Esempio	Visema
k	k	Occlusiva velare sorda	Kleid	k
l	l	Laterale alveolare approssimante	Loch	t
m	m	Nasale bilabiale	Milch	p
n	n	Nasale alveolare	Natur	t
ŋ	N	Nasale velare	klingen	k
p	p	Occlusiva bilabiale sorda	Parca	p
͡pf	pf	Affricata labiodentale sorda	Apfel	
ʀ	R	Vibrante uvulare	Regen	
s	s	fricativa alveolare sorda	Messer	s
ʃ	S	Fricativa postalveolare sorda	Fischer	S
t	t	Occlusiva alveolare sorda	Topf	T
͡ts	Ts	Affricata alveolare sorda	Zahl	
͡tʃ	tS	Affricata postalveolare sorda	deutsch	S
v	v	Fricativa labiodentale sonora	Wasser	f

IPA	X-SAMPA	Descrizione	Esempio	Visema
x	x	Fricativa velare sorda	kochen	k
z	z	Fricativa alveolare sonora	See	s
ʒ	Z	Fricativa postalveolare sonora	Orange	S
Vocali				
øː	2:	vocale anteriore semichiusa arrotondata allungata	böse	o
ɐ	6	vocale centrale quasi aperta	besser	a
ɐ̯	6_^	vocale centrale quasi aperta non sillabica	Klar	a
œ	9	vocale anteriore semiaperta arrotondata	können	O
ə	@	vocale centrale media	Rede	@
a	a	vocale anteriore aperta non arrotondata	Salz	a

IPA	X-SAMPA	Descrizione	Esempio	Visema
a:	a:	vocale anteriore aperta non arrotondata allungata	Sahne	a
aɪ	aI	dittongo	nein	a
aʊ	aU	dittongo	Augen	a
ɑ̃	A~	vocale posterior e aperta non arrotondata nasale	Restaurant	a
e:	e:	vocale anteriore semichiusa non arrotondata allungata	Rede	e
ɛ	E	vocale anteriore semiaperta non arrotondata	Keller	E
ɛ̃	E~	vocale anteriore semiaperta non arrotondata nasale	Terrain	E
i:	i:	vocale anteriore chiusa non arrotondata allungata	Lied	i
ɪ	I	vocale quasi anteriore quasi chiusa non arrotondata	bitte	i

IPA	X-SAMPA	Descrizione	Esempio	Visema
oː	o:	vocale posterior e semichius a arrotondata allungata	Kohl	o
ɔ	O	vocale posterior e semiaperta arrotondata	Koffer	O
ɔ̃	O~	vocale posterior e semiaperta arrotondata nasale	Annonce	O
ɔʏ	OY	dittongo	neu	O
uː	u:	vocale posteriore chiusa arrotondata allungata	Bruder	u
ʊ	U	vocale quasi posteriore quasi chiusa arrotondata	Wunder	u
yː	y:	vocale anteriore chiusa arrotondata allungata	kühl	u
ʏ	Y	vocale quasi anteriore quasi chiusa arrotondata	Küche	u
Simboli aggiuntivi				
ˈ	"	enfasi principale	Alabama	
ˌ	%	enfasi secondaria	Alabama	

IPA	X-SAMPA	Descrizione	Esempio	Visema
.	.	sillabazione	A.la.ba.ma	

Tedesco (austriaco) (de-AT)

La tabella seguente elenca i fonemi dell'alfabeto fonetico internazionale (International Phonetic Alphabet, IPA), i simboli X-SAMPA (Extended Speech Assessment Methods Phonetic Alphabet) e i corrispondenti visemi per il tedesco austriaco supportati da Amazon Polly.

Tabella fonemi/visemi

IPA	X-SAMPA	Descrizione	Esempio	Visema
Consonanti				
ʔ	?	occlusiva glottidale sorda		
b	b	occlusiva bilabiale sonora	Bier	p
d	d	occlusiva alveolare sonora	Dach	t
ç	C	fricativa palatale sorda	ich	k
͡dʒ	dZ	affricata postalveolare sonora	Dschungel	S
f	f	Fricativa labiodentale sorda	Vogel	f
g	g	Occlusiva velare sonora	Gabel	k
h	h	Fricativa faringale sorda	Haus	k

IPA	X-SAMPA	Descrizione	Esempio	Visema
j	j	Fricativa faringale sorda	jemand	i
k	k	Occlusiva velare sorda	Kleid	k
l	l	Laterale alveolare approssimante	Loch	t
m	m	Nasale bilabiale	Milch	p
n	n	Nasale alveolare	Natur	t
ŋ	N	Nasale velare	klingen	k
p	p	Occlusiva bilabiale sorda	Parca	p
͡pf	pf	Affricata labiodentale sorda	Apfel	
ʀ	R	Vibrante uvulare	Regen	
s	s	fricativa alveolare sorda	Messer	s
ʃ	S	Fricativa postalveolare sorda	Fischer	S
t	t	Occlusiva alveolare sorda	Topf	T
͡ts	Ts	Affricata alveolare sorda	Zahl	
͡tʃ	tS	Affricata postalveolare sorda	deutsch	S

IPA	X-SAMPA	Descrizione	Esempio	Visema
v	v	Fricativa labiodentale sonora	Wasser	f
x	x	Fricativa velare sorda	kochen	k
z	z	Fricativa alveolare sonora	See	s
ʒ	Z	Fricativa postalveolare sonora	Orange	S
Vocali				
øː	2:	vocale anteriore semichiusa arrotondata allungata	böse	o
ɐ	6	vocale centrale quasi aperta	besser	a
ɐ̯	6_^	vocale centrale quasi aperta non sillabica	Klar	a
œ	9	vocale anteriore semiaperta arrotondata	können	O
ə	@	vocale centrale media	Rede	@
a	a	vocale anteriore aperta non arrotondata	Salz	a

IPA	X-SAMPA	Descrizione	Esempio	Visema
a:	a:	vocale anteriore aperta non arrotondata allungata	Sahne	a
aɪ	aI	dittongo	nein	a
aʊ	aU	dittongo	Augen	a
ɑ̃	A~	vocale posterior e aperta non arrotondata nasale	Restaurant	a
e:	e:	vocale anteriore semichiusa non arrotondata allungata	Rede	e
ɛ	E	vocale anteriore semiaperta non arrotondata	Keller	E
ɛ̃	E~	vocale anteriore semiaperta non arrotondata nasale	Terrain	E
i:	i:	vocale anteriore chiusa non arrotondata allungata	Lied	i
ɪ	I	vocale quasi anteriore quasi chiusa non arrotondata	bitte	i

IPA	X-SAMPA	Descrizione	Esempio	Visema
oː	o:	vocale posterior e semichius a arrotondata allungata	Kohl	o
ɔ	O	vocale posterior e semiaperta arrotondata	Koffer	O
ɔ̃	O~	vocale posterior e semiaperta arrotondata nasale	Annonce	O
ɔʏ	OY	dittongo	neu	O
uː	u:	vocale posteriore chiusa arrotondata allungata	Bruder	u
ʊ	U	vocale quasi posteriore quasi chiusa arrotondata	Wunder	u
yː	y:	vocale anteriore chiusa arrotondata allungata	kühl	u
ʏ	Y	vocale quasi anteriore quasi chiusa arrotondata	Küche	u
Simboli aggiuntivi				
ˈ	"	enfasi principale	Alabama	
ˌ	%	enfasi secondaria	Alabama	

IPA	X-SAMPA	Descrizione	Esempio	Visema
.	.	sillabazione	A.la.ba.ma	

Tedesco (standard svizzero) (de-CH)

La tabella seguente elenca i fonemi dell'alfabeto fonetico internazionale (IPA), i simboli dell'alfabeto fonetico X-SAMPA (Extended Speech Assessment Methods) e i visemi corrispondenti per la voce tedesca (standard svizzero) supportata da Amazon Polly.

Tabella fonemi/visemi

IPA	X-SAMPA	Descrizione	Esempio	Visema
Consonanti				
ʔ	?	occlusiva glottidale sorda		
b	b	occlusiva bilabiale sonora	Bier	p
d	d	occlusiva alveolare sonora	Dach	t
ç	C	fricativa palatale sorda	ich	k
͡dʒ	dZ	affricata postalveolare sonora	Dschungel	S
f	f	Fricativa labiodentale sorda	Vogel	f
g	g	Occlusiva velare sonora	Gabel	k
h	h	Fricativa faringale sorda	Haus	k

IPA	X-SAMPA	Descrizione	Esempio	Visema
j	j	Fricativa faringale sorda	jemand	i
k	k	Occlusiva velare sorda	Kleid	k
l	l	Laterale alveolare approssimante	Loch	t
m	m	Nasale bilabiale	Milch	p
n	n	Nasale alveolare	Natur	t
ŋ	N	Nasale velare	klingen	k
p	p	Occlusiva bilabiale sorda	Parca	p
͡pf	pf	Affricata labiodentale sorda	Apfel	
ʀ	R	Vibrante uvulare	Regen	
s	s	fricativa alveolare sorda	Messer	s
ʃ	S	Fricativa postalveolare sorda	Fischer	S
t	t	Occlusiva alveolare sorda	Topf	T
͡ts	Ts	Affricata alveolare sorda	Zahl	
͡tʃ	tS	Affricata postalveolare sorda	deutsch	S

IPA	X-SAMPA	Descrizione	Esempio	Visema
v	v	Fricativa labiodentale sonora	Wasser	f
x	x	Fricativa velare sorda	kochen	k
z	z	Fricativa alveolare sonora	See	s
ʒ	Z	Fricativa postalveolare sonora	Orange	S
Vocali				
øː	2:	vocale anteriore semichiusa arrotondata allungata	böse	o
ɐ	6	vocale centrale quasi aperta	besser	a
ɐ̯	6_^	vocale centrale quasi aperta non sillabica	Klar	a
œ	9	vocale anteriore semiaperta arrotondata	können	O
ə	@	vocale centrale media	Rede	@
a	a	vocale anteriore aperta non arrotondata	Salz	a

IPA	X-SAMPA	Descrizione	Esempio	Visema
a:	a:	vocale anteriore aperta non arrotondata allungata	Sahne	a
aɪ	aI	dittongo	nein	a
aʊ	aU	dittongo	Augen	a
ɑ̃	A~	vocale posterior e aperta non arrotondata nasale	Restaurant	a
e:	e:	vocale anteriore semichiusa non arrotondata allungata	Rede	e
ɛ	E	vocale anteriore semiaperta non arrotondata	Keller	E
ɛ̃	E~	vocale anteriore semiaperta non arrotondata nasale	Terrain	E
i:	i:	vocale anteriore chiusa non arrotondata allungata	Lied	i
ɪ	I	vocale quasi anteriore quasi chiusa non arrotondata	bitte	i

IPA	X-SAMPA	Descrizione	Esempio	Visema
oː	o:	vocale posterior e semichius a arrotondata allungata	Kohl	o
ɔ	O	vocale posterior e semiaperta arrotondata	Koffer	O
ɔ̃	O~	vocale posterior e semiaperta arrotondata nasale	Annonce	O
ɔʏ	OY	dittongo	neu	O
uː	u:	vocale posteriore chiusa arrotondata allungata	Bruder	u
ʊ	U	vocale quasi posteriore quasi chiusa arrotondata	Wunder	u
yː	y:	vocale anteriore chiusa arrotondata allungata	kühl	u
ʏ	Y	vocale quasi anteriore quasi chiusa arrotondata	Küche	u
Simboli aggiuntivi				
ˈ	“	enfasi principale	Alabama	
ˌ	%	enfasi secondaria	Alabama	

IPA	X-SAMPA	Descrizione	Esempio	Visema
.	.	sillabazione	A.la.ba.ma	

Hindi (hi-IN)

La tabella seguente elenca i fonemi dell'alfabeto fonetico internazionale (International Phonetic Alphabet, IPA), i simboli X-SAMPA (Extended Speech Assessment Methods Phonetic Alphabet) e i tipi di suono del fonema per le voci hindi supportati da Amazon Polly.

Per ulteriori fonemi utilizzati insieme all'hindi, vedi Inglese (India) (en-IN).

Tabella fonemi/visemi

IPA	X-SAMPA	Descrizione	Esempio
Consonanti			
pʰ	p_h	occlusiva bilabiale sorda aspirata	फूल (phool)
bʰ	b_h	occlusiva bilabiale sonora aspirata	भारी (bhaari)
t̪	t_d	occlusiva dentale sorda	तापमान (taapmaan)
t̪ʰ	t_d_h	occlusiva dentale sorda aspirata	थोड़ा (thoda)
d̪	d_d	occlusiva dentale sonora	दल्ली (dilli)
d̪ʰ	d_d_h	occlusiva dentale sonora aspirata	धोबी (dhobi)
ʈ	t`	occlusiva retroflessa sorda	कटोरा (katora)
ʈʰ	t`_h	occlusiva retroflessa sorda aspirata	ठंड (thand)

IPA	X-SAMPA	Descrizione	Esempio
ɖ	d`	occlusiva retroflessa sonora	डर (darr)
ɖʰ	d`_h	occlusiva retroflessa sonora aspirata	ढाल (dhal)
tʃʰ	tS_h	affricata palatale sorda aspirata	छाल (chaal)
dʒʰ	dZ_h	affricata palatale sonora aspirata	झाल (jhaal)
kʰ	k_h	occlusiva velare sorda aspirata	खान (khan)
gʰ	g_h	occlusiva velare sonora aspirata	घान (ghaan)
ɳ	n`	nasale retroflessa	क्षण (kshan)
ɾ	4	monovibrante alveolare	राम (ram)
ɽ	r`	monovibrante retroflessa piana	बड़ा (bada)
ɽʰ	r`_h	monovibrante retroflessa sonora aspirata	बढ़ी (barhi)
ʋ	v\	approssimante bilabiale	वसूल (wasool)
Vocali			
ə	@_o	vocale centrale media	अच्छा (achhaa)
ə̃	@~	vocale centrale media nasalizzata	हँसना (hansnaa)

IPA	X-SAMPA	Descrizione	Esempio
a	A_o	vocale anteriore aperta non arrotondata	आग (aag)
ã	A~	vocale anteriore aperta non arrotondata nasalizzata	घ़डयिँ (ghariyaan)
ɪ	I_o	vocale quasi anteriore quasi chiusa non arrotondata	इक्कीस (ikkees)
ɪ̃	I~	vocale quasi anteriore quasi chiusa non arrotondata nasalizzata	संचिाई (sinchai)
i	i_o	vocale anteriore chiusa non arrotondata	बल्लिी (billee)
ĩ	i~	vocale anteriore chiusa non arrotondata nasalizzata	नही (nahin)
ʊ	U_o	vocale quasi posteriore quasi chiusa arrotondata	उलूल (ullu)
ʊ̃	U~	vocale quasi posteriore quasi chiusa nasalizzata	मुँह (munh)
u	u_o	vocale posteriore chiusa arrotondata	फूल (phool)
ũ	u~	vocale posteriore chiusa arrotondata nasalizzata	ऊँट (oont)
ɔ	O_o	vocale posteriore semiaperta arrotondata	कौन (kaun)

IPA	X-SAMPA	Descrizione	Esempio
ɔ̃	O~	vocale posteriore semiaperta arrotondata nasalizzata	भौ (bhaun)
o	o	vocale posteriore semichiusa arrotondata	सोना (sona)
õ	o~	vocale posteriore semichiusa arrotondata nasalizzata	क्यों (kyon)
ɛ	E_o	vocale anteriore semiaperta non arrotonda ta	पैसा (paisa)
ɛ̃	E~	vocale anteriore semiaperta non arrotonda ta nasalizzata	मैं (main)
e	e	vocale anteriore semichiusa non arrotonda ta	एक (ek)
ẽ	e~	vocale anteriore semichiusa non arrotonda ta nasalizzata	किताबें (kitabein)

Islandese (is-IS)

La tabella seguente elenca i fonemi dell'alfabeto fonetico internazionale (International Phonetic Alphabet, IPA), i simboli X-SAMPA (Extended Speech Assessment Methods Phonetic Alphabet) e i corrispondenti visemi per la lingua islandese supportati da Amazon Polly.

Tabella fonemi/visemi

IPA	X-SAMPA	Descrizione	Esempio	Visema
Consonanti				
b	b	occlusiva bilabiale sonora	grasbakkanum	0
c	c	Occlusiva palatale sorda	pakkin	k
c^h	c_h	Occlusiva palatale sorda aspirata	anarkistai	k
ç	C	fricativa palatale sorda	héðan	k
d	d	occlusiva alveolare sonora	bóndi	t
ð	D	fricativa dentale sonora	borð	T
f	f	fricativa labiodentale sorda	duft	f
g	g	occlusiva velare sonora	holgóma	k
ɣ	G	fricativa velare sonora	hugur	k
h	h	fricativa glottidale sorda	heili	k
j	j	approssimante palatale	jökull	i

IPA	X-SAMPA	Descrizione	Esempio	Visema
kʰ	k_h	Occlusiva velare sorda aspirata	ósköpunum	k
l	l	approssimante alveolare laterale	gólf	t
l̥	l_0	laterale alveolare approssimante sorda	fólk	t
m	m	nasale bilabiale	september	p
m̥	m_0	Nasale bilabiale sorda	kompa	p
n	n	nasale alveolare	númer	t
n̥	n_0	Nasale alveolare sorda	pöntun	t
ɲ	J	nasale palatale	pælingar	J
ŋ	N	nasale velare	söngvarann	k
ŋ̊	N_0	Nasale velare sorda	frænka	k
pʰ	p_h	Occlusiva bilabiale sorda aspirata	afplánun	p
r	r	vibrante alveolare	afskrifta	r
r̥	r_0	Vibrante alveolare sorda	andvörpum	r
s	s	fricativa alveolare sorda	baðhús	s

IPA	X-SAMPA	Descrizione	Esempio	Visema
t^h	t_h	Occlusiva alveolare sorda aspirata	tanki	t
θ	T	fricativa dentale sorda	þeldökki	T
v	v	fricativa labiodentale sonora	silfur	f
w	w	approssimante labiovelare sonora		u
x	x	fricativa velare sorda	samfélags	k
Vocali				
œ	9	vocale anteriore semiaperta arrotondata	þröskuldinum	O
œ:	9:	vocale anteriore semiaperta arrotondata allungata	tvö	O
a	a	vocale anteriore aperta non arrotondata	nefna	a
a:	a:	vocale anteriore aperta non arrotondata allungata	fara	a
au	au	dittongo	átta	a

IPA	X-SAMPA	Descrizione	Esempio	Visema
au:	au:	dittongo	átján	a
ɛ	E	vocale anteriore semiaperta non arrotondata	kennari	E
ɛ:	E:	vocale anteriore semiaperta non arrotondata allungata	dreka	E
i	i	vocale anteriore chiusa non arrotondata	Gúlíver	i
i:	i:	vocale anteriore chiusa non arrotondata allungata	þrír	i
ɪ	I	vocale quasi anteriore quasi chiusa non arrotondata	samspil	i
ɪ:	I:	vocale quasi anteriore quasi chiusa non arrotondata allungata	stig	i
ɔ	O	vocale posterior e semiaperta arrotondata	regndropar	O

IPA	X-SAMPA	Descrizione	Esempio	Visema
ɔː	O:	vocale posterior e semiapert a arrotondata allungata	ullarbolur	O
ɔu	Ou	dittongo	tólf	O
ɔuː	Ou:	dittongo	fjórir	O
u	u	vocale posteriore chiusa arrotondata	stúlkan	u
uː	u:	vocale posteriore chiusa arrotondata allungata	frú	u
ʏ	Y	vocale quasi anteriore quasi chiusa arrotondata	tíu	u
ʏː	Y	vocale quasi anteriore quasi chiusa arrotondata allungata	gruninn	u
Simboli aggiuntivi				
ˈ	"	enfasi principale	Alabama	
ˌ	%	enfasi secondaria	Alabama	
.	.	sillabazione	A.la.ba.ma	

Italiano (it-IT)

La tabella seguente elenca i fonemi dell'alfabeto fonetico internazionale (International Phonetic Alphabet, IPA), i simboli X-SAMPA (Extended Speech Assessment Methods Phonetic Alphabet) e i corrispondenti visemi per la lingua italiana supportati da Amazon Polly.

Tabella fonemi/visemi

IPA	X-SAMPA	Descrizione	Esempio	Visema
Consonanti				
b	b	occlusiva bilabiale sonora	bacca	p
d	d	occlusiva alveolare sonora	dama	t
͡dz	dz	affricata alveolare sonora	zero	s
͡dʒ	dZ	affricata postalveolare sonora	giro	S
f	f	fricativa labiodentale sorda	famiglia	f
g	g	occlusiva velare sonora	gatto	k
h	h	fricativa glottidale sorda	horror	k
j	j	approssimante palatale	dieci	i
k	k	occlusiva velare sorda	campo	k

IPA	X-SAMPA	Descrizione	Esempio	Visema
l	l	approssimante alveolare laterale	lido	t
ʎ	L	approssimante laterale palatale	aglio	J
m	m	nasale bilabiale	mille	p
n	n	nasale alveolare	nove	t
ɲ	J	nasale palatale	lasagne	J
p	p	occlusiva bilabiale sorda	pizza	p
r	r	vibrante alveolare	risata	r
s	s	fricativa alveolare sorda	sei	s
ʃ	S	fricativa postalveolare sorda	scienza	S
t	t	occlusiva alveolare sorda	tavola	t
͡ts	ts	affricata alveolare sorda	forza	s
͡tʃ	tS	affricata postalveolare sorda	cielo	S
v	v	fricativa labiodentale sonora	venti	f
w	w	approssimante labiovelare sonora	quattro	u

IPA	X-SAMPA	Descrizione	Esempio	Visema
z	z	fricativa alveolare sonora	bisogno	s
ʒ	Z	fricativa postalveolare sonora	bijou	S
Vocali				
a	a	vocale anteriore aperta non arrotondata	arco	a
e	e	vocale anteriore semichiusa non arrotondata	tre	e
ɛ	E	vocale anteriore semiaperta non arrotondata	ettaro	E
i	i	vocale anteriore chiusa non arrotondata	impero	i
o	o	vocale posteriore semichiusa arrotondata	cento	o
ɔ	O	vocale posteriore semiaperta arrotondata	otto	O
u	u	vocale posteriore chiusa arrotondata	uno	u
Simboli aggiuntivi				

IPA	X-SAMPA	Descrizione	Esempio	Visema
ˈ	"	enfasi principale	Alabama	
ˌ	%	enfasi secondaria	Alabama	
.	.	sillabazione	A.la.ba.ma	

Giapponese (ja-JP)

Amazon Polly supporta gli alfabeti Pronuncia Kana e Yomigana per il giapponese. Per fare in modo che Amazon Polly utilizzi la pronuncia fonetica con questi alfabeti, utilizzare l'attributo `alphabet="x-amazon-phonetic standard used"` del fonema.

- `x-amazon-pron-kana` - indica l'utilizzo della Pronuncia Kana. La Pronuncia Kana sono caratteri speciali Katakana utilizzati per la trascrizione fonetica e possono codificare l'accento dell'intonazione.
- `x-amazon-yomigana` - indica l'utilizzo di Yomigana. Yomigana può essere l'alfabeto tradizionale Katakana, Hiragana e latino interpretati come romanizzazione hepburn.

Il seguente esempio di codice mostra come vengono utilizzati:

Pronuncia 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>
```

La tabella seguente elenca i fonemi dell'alfabeto fonetico internazionale (International Phonetic Alphabet, IPA), i simboli X-SAMPA (Extended Speech Assessment Methods Phonetic Alphabet) e i corrispondenti visemi per la lingua giapponese supportati da Amazon Polly.

IPA	X-SAMPA	Descrizione	Esempio	Visema
Consonanti				
ɾ	4	monovibrante alveolare	練習, renshuu	t
ʔ	?	occlusiva glottidale sorda	あつっ, atsu'	
b	b	occlusiva bilabiale sonora	舞踊, buyou	p
β	B	fricativa bilabiale sonora	ヴィンテージ, vinteeji	B
c	c	Occlusiva palatale sorda	ききょう, kikyou	k
ç	C	fricativa palatale sorda	人, hito	k
d	d	occlusiva alveolare sonora	濁点, dakuten	t
͡dʑ	dz\	affricata alveopalatale sonora	純, jun	J
g	g	occlusiva velare sonora	ご飯, gohan	k
h	h	fricativa glottidale sorda	本, hon	k
j	j	approssimante palatale	屋根, yane	i

IPA	X-SAMPA	Descrizione	Esempio	Visema
ɟ	J\	occlusiva palatale sonora	行儀, gyougi	J
k	k	occlusiva velare sorda	漢字, kanji	k
ɺ	l\	monovibrante laterale alveolare	釣り, tsuri	r
ɺj	l\j	monovibrante laterale alveolare , approssimante palatale	流行, ryuukou	r
m	m	nasale bilabiale	飯, meshi	p
n	n	nasale alveolare	猫, neko	t
ɲ	J	nasale palatale	日本, nippon	J
ɴ	N\	nasale uvulare	缶, kan	k
p	p	occlusiva bilabiale sorda	パン, pan	p
ɸ	p\	fricativa bilabiale sorda	福, huku	f
s	s	fricativa alveolare sorda	層, sou	s
ɕ	s\	fricativa alveolopa latale sorda	書簡, shokan	J
t	t	occlusiva alveolare sorda	手紙, tegami	t

IPA	X-SAMPA	Descrizione	Esempio	Visema
͡ts	ts	affricata alveolare sorda	釣り, tsuri	s
͡tɕ	ts\	affricata alveolopalatale sorda	吉, kichi	J
w	w	approssimante labiovelare sonora	電話, denwa	u
z	z	fricativa alveolare sonora	座敷, zashiki	s
Vocali				
äː	a:_"	vocale centrale aperta non arrotondata allungata	羽蟻, haari	a
ä	a_"	vocale centrale aperta non arrotondata	仮名, kana	a
eː	e:_o	vocale anteriore media non arrotondata allungata	学生, gakusei	@
e	e_o	vocale anteriore media non arrotondata	歴, reki	@
i	i	vocale anteriore chiusa non arrotondata	気, ki	i

IPA	X-SAMPA	Descrizione	Esempio	Visema
iː	i:	vocale anteriore chiusa non arrotondata allungata	詩歌, shiika	i
ɯ	M	vocale posterior e chiusa non arrotondata	運, un	i
ɯː	M:	vocale posterior e chiusa non arrotondata allungata	宗教, shuukyou	i
oː	o:_o	vocale posteriore media arrotondata allungata	購読, koodoku	o
o	o_o	vocale posteriore media arrotondata	読者, dokusha	o

Coreano (ko-KR)

La tabella seguente elenca i fonemi dell'alfabeto fonetico internazionale (International Phonetic Alphabet, IPA), i simboli X-SAMPA (Extended Speech Assessment Methods Phonetic Alphabet) e i corrispondenti visemi per la lingua coreana supportati da Amazon Polly.

IPA	X-SAMPA	Descrizione	Esempio	Visema
Consonanti				
k	k	occlusiva velare sorda	강, [g]ang	k

IPA	X-SAMPA	Descrizione	Esempio	Visema
k#	k_t	Occlusiva velare sorda forte	깨, [kk]e	k
n	n	nasale alveolare	남, [n]am	t
t	t	occlusiva alveolare sorda	도, [d]o	t
t#	t_t	Occlusiva alveolare sorda forte	때, [tt]e	t
ɾ	4	monovibrante alveolare	사랑, sa[r]ang	t
l	l	approssimante alveolare laterale	돌, do[l]	t
m	m	nasale bilabiale	무, [m]u	p
p	p	occlusiva bilabiale sorda	봄, [b]om	p
p#	p_t	Occlusiva bilabiale sorda forte	뻘, [pp]eol	p
s	s	fricativa alveolare sorda	새, [s]e	s
s#	s_t	fricativa alveolare sorda forte	씨, [ss]i	s
ŋ	N	nasale velare	방, ba[ng]	k
͡tɕ	ts\	affricata alveolopalatale sorda	조, [j]o	J
͡t#ɕ	ts_t	affricata alveolopalatale sorda forte	찌, [jj]i	J

IPA	X-SAMPA	Descrizione	Esempio	Visema
͡tɕʰ	ts_h	affricata alveolopalatale sorda aspirata	차, [ch]a	J
kʰ	k_h	Occlusiva velare sorda aspirata	코, [k]o	k
tʰ	t_h	Occlusiva alveolare sorda aspirata	통, [t]ong	t
pʰ	p_h	Occlusiva bilabiale sorda aspirata	패, [p]e	p
h	h	fricativa glottidale sorda	힘, [h]im	k
j	j	approssimante palatale	양, [y]ang	i
w	w	approssimante labiovelare sonora	왕, [w]ang	u
ɥ	M\	approssimante velar>	의, [wj]i	i
Vocali				
a	a	vocale anteriore aperta non arrotondata	밥, b[a]b	a
ʌ	V	Vocale posterior e semiaperta non arrotondata	정, j[eo]ng	E

IPA	X-SAMPA	Descrizione	Esempio	Visema
ɛ	E	vocale anteriore semiaperta non arrotondata	배, b[e]	E
o	o	vocale posterior e semichiusa arrotondata	노, n[o]	o
u	u	vocale posteriore chiusa arrotondata	둘, d[u]l	u
ɯ	M	vocale posterior e chiusa non arrotondata	은, [eu]n	i
i	i	vocale anteriore chiusa non arrotondata	김, k[i]m	i

Norvegese (nb-NO)

Il grafico seguente elenca tutto l'insieme di fonemi dell'alfabeto fonetico internazionale (International Phonetic Alphabet, IPA) e i simboli X-SAMPA (Extended Speech Assessment Methods Phonetic Alphabet) e i corrispondenti visemi supportati da Amazon Polly per la lingua norvegese.

IPA	X-SAMPA	Descrizione	Esempio	Visema
Consonanti				
ɾ	4	monovibrante alveolare	prøv	t
b	b	occlusiva bilabiale sonora	labb	p

IPA	X-SAMPA	Descrizione	Esempio	Visema
ç	C	fricativa palatale sorda	kino	k
d	d	occlusiva alveolare sonora	ladd	t
ɖ	d`	occlusiva retroflessa sonora	verdi	t
f	f	fricativa labiodentale sorda	fot	f
g	g	occlusiva velare sonora	tagg	k
h	h	fricativa glottidale sorda	ha	k
j	j	approssimante palatale	gi	i
k	k	occlusiva velare sorda	takk	k
l	l	approssimante alveolare laterale	fall, ball	t
ɭ	l`	approssimante laterale retroflessa	ærlig	t
m	m	nasale bilabiale	lam	p
n	n	nasale alveolare	vann	t
ɳ	n`	nasale retroflessa	garn	t
ŋ	N	nasale velare	sang	k

IPA	X-SAMPA	Descrizione	Esempio	Visema
p	p	occlusiva bilabiale sorda	hopp	p
s	s	fricativa alveolare sorda	lass	s
ʂ	s`	fricativa retroflessa sorda	års	S
ʃ	S	fricativa postalveolare sorda	skyt	S
t	t	occlusiva alveolare sorda	Lat	t
ʈ	t`	occlusiva retroflessa sorda	hardt	t
ʋ	v\	approssimante labiodentale	vin	f
w	w	approssimante labiovelare sonora	will	x
Vocali				
øː	2:	vocale anteriore semichiusa arrotondata allungata	søt	o
œ	9	vocale anteriore semiaperta arrotondata	søtt	O
ə	@	vocale centrale media	ape	@

IPA	X-SAMPA	Descrizione	Esempio	Visema
æː	{:	vocale anteriore quasi aperta non arrotondata allungata	vær	a
ʉ	}	vocale centrale chiusa arrotondata	lund	u
ʉː	}:	vocale centrale chiusa arrotondata allungata	lun	u
æ	{	vocale anteriore quasi aperta non arrotondata	vært	a
ɑ	A	vocale posterior e aperta non arrotondata	hatt	a
ɑː	A:	vocale posterior e aperta non arrotondata allungata	hat	a
eː	e:	vocale anteriore semichiusa non arrotondata allungata	sen	e
ɛ	E	vocale anteriore semiaperta non arrotondata	send	E

IPA	X-SAMPA	Descrizione	Esempio	Visema
iː	i:	vocale anteriore chiusa non arrotondata allungata	vin	i
ɪ	I	vocale quasi anteriore quasi chiusa non arrotondata	vind	i
oː	o:	vocale posterior e semichius a arrotondata allungata	våt	o
ɔ	O	vocale posterior e semiaperta arrotondata	vått	O
uː	u:	vocale posteriore chiusa arrotondata allungata	bok	u
ʊ	U	vocale quasi posteriore quasi chiusa arrotondata	bukk	u
yː	y:	vocale anteriore chiusa arrotondata allungata	lyn	u
ʏ	Y	vocale quasi anteriore quasi chiusa arrotondata	lynne	u
Simboli aggiuntivi				

IPA	X-SAMPA	Descrizione	Esempio	Visema
ˈ	"	enfasi principale	Alabama	
ˌ	%	enfasi secondaria	Alabama	
.	.	sillabazione	A.la.ba.ma	

Polacco (pl-PL)

La tabella seguente elenca i fonemi dell'alfabeto fonetico internazionale (International Phonetic Alphabet, IPA), i simboli X-SAMPA (Extended Speech Assessment Methods Phonetic Alphabet) e i corrispondenti visemi per la lingua polacca supportati da Amazon Polly.

Tabella fonemi/visemi

IPA	X-SAMPA	Descrizione	Esempio	Visema
Consonanti				
b	b	occlusiva bilabiale sonora	bobas, belka	p
d	d	occlusiva alveolare sonora	dar, do	t
͡dz	dz	affricata alveolare sonora	dzwon, widzowie	s
͡dʑ	dz\	affricata alveopala tale sonora	dźwięk	J
͡dʐ	dz`	affricata retroflessa sonora	dżem, dżungla	S
f	f	fricativa labiodent ale sorda	furtka, film	f

IPA	X-SAMPA	Descrizione	Esempio	Visema
g	g	occlusiva velare sonora	gazeta, waga	k
h	h	fricativa glottidale sorda	chleb, handel	k
j	j	approssimante palatale	jak, maja	i
k	k	occlusiva velare sorda	kura, marek	k
l	l	approssimante alveolare laterale	lipa, alicja	t
m	m	nasale bilabiale	matka, molo	p
n	n	nasale alveolare	norka	t
ɲ	J	nasale palatale	koń, toruń	J
p	p	occlusiva bilabiale sorda	pora, stop	p
r	r	vibrante alveolare	rok, park	r
s	s	fricativa alveolare sorda	sum, pas	s
ɕ	s\	fricativa alveolopa latale sorda	śruba, śnieg	J
ʂ	s`	fricativa retroflessa sorda	szum, masz	S
t	t	occlusiva alveolare sorda	tok, stół	t

IPA	X-SAMPA	Descrizione	Esempio	Visema
͡ts	ts	affricata alveolare sorda	car, co	s
͡tɕ	ts\	affricata alveolopa latale sorda	ćma, mieć	J
͡tʂ	ts`	affricata retroflessa sorda	czas, raczej	S
v	v	fricativa labiodent ale sonora	worek, mewa	f
w	w	approssimante labiovelare sonora	łaska, mało	u
z	z	fricativa alveolare sonora	zero	s
ʑ	z\	fricativa alveolopa latale sonora	źrebię, bieliźnie	J
ʐ	z`	fricativa retroflessa sonora	żar, żona	S
Vocali				
a	a	vocale anteriore aperta non arrotondata	ja	a
ɛ	E	vocale anteriore semiaperta non arrotondata	echo	E
ɛ̃	E~	vocale anteriore semiaperta non arrotondata nasale	węże	E

IPA	X-SAMPA	Descrizione	Esempio	Visema
i	i	vocale anteriore chiusa non arrotondata	ile	i
ɔ	O	vocale posterior e semiaperta arrotondata	oczy	O
ɔ̃	O~	vocale posterior e semiaperta arrotondata nasale	wąż	O
u	u	vocale posteriore chiusa arrotondata	uczta	u
ɨ	1	vocale centrale chiusa non arrotondata	byk	i
Simboli aggiuntivi				
ˈ	"	enfasi principale	Alabama	
ˌ	%	enfasi secondaria	Alabama	
.	.	sillabazione	A.la.ba.ma	

Portoghese (pt-PT)

La tabella seguente elenca i fonemi dell'alfabeto fonetico internazionale (International Phonetic Alphabet, IPA), i simboli X-SAMPA (Extended Speech Assessment Methods Phonetic Alphabet) e i corrispondenti visemi per la lingua portoghese supportati da Amazon Polly.

Tabella fonemi/visemi

IPA	X-SAMPA	Descrizione	Esempio	Visema
Consonanti				
ɾ	4	monovibrante alveolare	pira	t
b	b	occlusiva bilabiale sonora	dato	p
d	d	occlusiva alveolare sonora	dato	t
f	f	fricativa labiodent ale sorda	facto	f
g	g	occlusiva velare sonora	gato	k
j	j	approssimante palatale	paraguay	i
k	k	occlusiva velare sorda	cacto	k
l	l	approssimante alveolare laterale	galo	t
ʎ	L	approssimante laterale palatale	galho	J
m	m	nasale bilabiale	mato	p
n	n	nasale alveolare	nato	t
ɲ	J	nasale palatale	pinha	J
p	p	occlusiva bilabiale sorda	pato	p

IPA	X-SAMPA	Descrizione	Esempio	Visema
ʀ	R\	vibrante uvulare	barroso	k
s	s	fricativa alveolare sorda	saca	s
ʃ	S	fricativa postalveolare sorda	chato	S
t	t	occlusiva alveolare sorda	tacto	t
v	v	fricativa labiodentale sonora	vaca	f
w	w	approssimante labiovelare sonora	mau	u
z	z	fricativa alveolare sonora	zaca	s
ʒ	Z	fricativa postalveolare sonora	jacto	S
Vocali				
a	a	vocale anteriore aperta non arrotondata	parto	a
ã	a~	vocale anteriore aperta non arrotondata nasale	pega	a
e	e	vocale anteriore semichiusa non arrotondata	pega	e

IPA	X-SAMPA	Descrizione	Esempio	Visema
ẽ	e~	vocale anteriore semichiusa non arrotondata nasale	movem	e
ɛ	E	vocale anteriore semiaperta non arrotondata	café	E
i	i	vocale anteriore chiusa non arrotondata	lingueta	i
ĩ	i~	vocale anteriore chiusa non arrotondata nasale	cinto	i
o	o	vocale posterior e semichiusa arrotondata	poder	o
õ	o~	vocale posterior e semichiusa arrotondata nasale	compra	o
ɔ	O	vocale posterior e semiaperta arrotondata	cotó	O
u	u	vocale posteriore chiusa arrotondata	fui	u
ũ	u~	vocale posteriore chiusa arrotondata nasale	sunto	u
Simboli aggiuntivi				

IPA	X-SAMPA	Descrizione	Esempio	Visema
ˈ	"	enfasi principale	Alabama	
ˌ	%	enfasi secondaria	Alabama	
.	.	sillabazione	A.la.ba.ma	

Portoghese (Brasile) (pt-BR)

La tabella seguente elenca i fonemi dell'alfabeto fonetico internazionale (International Phonetic Alphabet, IPA), i simboli X-SAMPA (Extended Speech Assessment Methods Phonetic Alphabet) e i corrispondenti visemi per la lingua brasiliana supportati da Amazon Polly.

Tabella fonemi/visemi

IPA	X-SAMPA	Descrizione	Esempio	Visema
Consonanti				
ɾ	4	monovibrante alveolare	pira	t
b	b	occlusiva bilabiale sonora	bato	p
d	d	occlusiva alveolare sonora	dato	t
͡dʒ	dZ	affricata postalveolare sonora	idade	S
f	f	fricativa labiodentale sorda	facto	f
g	g	occlusiva velare sonora	gato	k

IPA	X-SAMPA	Descrizione	Esempio	Visema
j	j	approssimante palatale	paraguay	i
k	k	occlusiva velare sorda	cacto	k
l	l	approssimante alveolare laterale	galo	t
ʎ	L	approssimante laterale palatale	galho	J
m	m	nasale bilabiale	mato	p
n	n	nasale alveolare	nato	t
ɲ	J	nasale palatale	pinha	J
p	p	occlusiva bilabiale sorda	pato	p
s	s	fricativa alveolare sorda	saca	s
ʃ	S	fricativa postalveolare sorda	chato	S
t	t	occlusiva alveolare sorda	tacto	t
͡tʃ	tS	affricata postalveolare sorda	noite	S
v	v	fricativa labiodentale sonora	vaca	f
w	w	approssimante labiovelare sonora	mau	u

IPA	X-SAMPA	Descrizione	Esempio	Visema
χ	X	fricativa uvulare sorda	carro	k
z	z	fricativa alveolare sonora	zaca	s
ʒ	Z	fricativa postalveolare sonora	jacto	S
Vocali				
a	a	vocale anteriore aperta non arrotondata	parto	a
ã	a~	vocale anteriore aperta non arrotondata nasale	pensamos	a
e	e	vocale anteriore semichiusa non arrotondata	pega	e
ẽ	e~	vocale anteriore semichiusa non arrotondata nasale	movem	e
ɛ	E	vocale anteriore semiaperta non arrotondata	café	E
i	i	vocale anteriore chiusa non arrotondata	lingueta	i

IPA	X-SAMPA	Descrizione	Esempio	Visema
ĩ	i~	vocale anteriore chiusa non arrotondata nasale	cinto	i
o	o	vocale posterior e semichiusa arrotondata	poder	o
õ	o~	vocale posterior e semichiusa arrotondata nasale	compra	o
ɔ	O	vocale posterior e semiaperta arrotondata	cotó	O
u	u	vocale posteriore chiusa arrotondata	fui	u
ũ	u~	vocale posteriore chiusa arrotondata nasale	sunto	u
Simboli aggiuntivi				
ˈ	"	enfasi principale	Alabama	
ˌ	%	enfasi secondaria	Alabama	
.	.	sillabazione	A.la.ba.ma	

Romeno (ro-RO)

La tabella seguente elenca i fonemi dell'alfabeto fonetico internazionale (International Phonetic Alphabet, IPA), i simboli X-SAMPA (Extended Speech Assessment Methods Phonetic Alphabet) e i corrispondenti visemi per la lingua rumena supportati da Amazon Polly.

Tabella fonemi/visemi

IPA	X-SAMPA	Descrizione	Esempio	Visema
Consonanti				
b	b	occlusiva bilabiale sonora	bubă	p
d	d	occlusiva alveolare sonora	după	t
͡dʒ	dZ	affricata postalveolare sonora	george	S
f	f	fricativa labiodentale sorda	afacere	f
g	g	occlusiva velare sonora	agri#	k
h	h	fricativa glottidale sorda	harpă	k
j	j	approssimante palatale	baie	i
k	k	occlusiva velare sorda	co#	k
l	l	approssimante alveolare laterale	lampa	t
m	m	nasale bilabiale	mama	p
n	n	nasale alveolare	nor	t
p	p	occlusiva bilabiale sorda	pilă	p
r	r	vibrante alveolare	rampă	r

IPA	X-SAMPA	Descrizione	Esempio	Visema
s	s	fricativa alveolare sorda	soare	s
ʃ	S	fricativa postalveolare sorda	ma#ină	S
t	t	occlusiva alveolare sorda	tata	t
͡ts	ts	affricata alveolare sorda	#ară	s
͡tʃ	tS	affricata postalveolare sorda	ceai	S
v	v	fricativa labiodentale sonora	via#ă	f
w	w	approssimante labiovelare sonora	beau	u
z	z	fricativa alveolare sonora	mozol	s
ʒ	Z	fricativa postalveolare sonora	joacă	S
Vocali				
ə	@	vocale centrale media	babă	@
a	a	vocale anteriore aperta non arrotondata	casa	a

IPA	X-SAMPA	Descrizione	Esempio	Visema
e	e	vocale anteriore semichiusa non arrotondata	elan	e
e̯	e_^	vocale anteriore semichiusa non arrotondata non sillabica	beau	e
i	i	vocale anteriore chiusa non arrotondata	mie	i
o	o	vocale posterior e semichiusa arrotondata	oră	o
oa	o_^a	dittongo	oare	o
u	u	vocale posteriore chiusa arrotondata	unde	u
ɨ	1	vocale centrale chiusa non arrotondata	România	i
Simboli aggiuntivi				
ˈ	"	enfasi principale	Alabama	
ˌ	%	enfasi secondaria	Alabama	
.	.	sillabazione	A.la.ba.ma	

Russo (ru-RU)

La tabella seguente elenca i fonemi dell'alfabeto fonetico internazionale (International Phonetic Alphabet, IPA), i simboli X-SAMPA (Extended Speech Assessment Methods Phonetic Alphabet) e i corrispondenti visemi per la lingua russa supportati da Amazon Polly.

Tabella fonemi/visemi

IPA	X-SAMPA	Descrizione	Esempio	Visema
Consonanti				
b	b	occlusiva bilabiale sonora	борт	p
b^j	b'	occlusiva bilabiale sonora palataliz zata	бюро	p
d	d	occlusiva alveolare sonora	дом	t
d^j	d'	occlusiva alveolare sonora palataliz zata	дядя	t
f	f	fricativa labiodent ale sorda	флаг	f
f^j	f'	fricativa labiodent ale palatalizzata sorda	февраль	f
g	g	occlusiva velare sonora	нога	k
g^j	g'	occlusiva velare sonora palataliz zata	герой	k

IPA	X-SAMPA	Descrizione	Esempio	Visema
j	j	approssimante palatale	дизайн, ящик	i
k	k	occlusiva velare sorda	кот	k
k^j	k'	occlusiva velare sorda palatalizzata	кино	k
l	l	approssimante alveolare laterale	лампа	t
l^j	l'	laterale alveolare approssimante palatalizzata	лес	t
m	m	nasale bilabiale	мама	p
m^j	m'	nasale bilabiale palatalizzata	мяч	p
n	n	nasale alveolare	нос	t
n^j	n'	nasale alveolare palatalizzata	няня	t
p	p	occlusiva bilabiale sorda	папа	p
p^j	p'	occlusiva bilabiale sorda palatalizzata	перо	p
r	r	vibrante alveolare	роза	r
r^j	r'	vibrante alveolare palatalizzata	рюмка	r

IPA	X-SAMPA	Descrizione	Esempio	Visema
s	s	fricativa alveolare sorda	сыр	s
sʲ	s'	fricativa alveolare sorda palatalizzata	сердце, русь	s
ɕː	s\:	fricativa alveolopalatale sorda allungata	щека	J
ʂ	s`	fricativa retroflessa sorda	шум	S
t	t	occlusiva alveolare sorda	точка	t
tʲ	t'	occlusiva alveolare sorda palatalizzata	тётя	t
͡ts	ts	affricata alveolare sorda	царь	s
͡tɕ	ts\	affricata alveolopalatale sorda	час	J
v	v	fricativa labiodentale sonora	вор	f
vʲ	v'	fricativa labiodentale palatalizzata sonora	верфь	f
x	x	fricativa velare sorda	хор	k
xʲ	x'	fricativa velare sorda palatalizzata	химия	k

IPA	X-SAMPA	Descrizione	Esempio	Visema
z	z	fricativa alveolare sonora	зуб	s
zʲ	z'	fricativa alveolare sonora palataliz zata	зима	s
ʑː	z\:	fricativa alveolopa latale sonora allungata	уезжать	J
ʐ	z`	fricativa retroflessa sonora	жена	S
Vocali				
ə	@	vocale centrale media	канарейка	@
a	a	vocale anteriore aperta non arrotondata	два, яблоко	a
e	e	vocale anteriore semichiusa non arrotondata	печь	e
ɛ	E	vocale anteriore semiaperta non arrotondata	это	E
i	i	vocale anteriore chiusa non arrotondata	один, четыре	i

IPA	X-SAMPA	Descrizione	Esempio	Visema
o	o	vocale posterior e semichiusa arrotondata	кот	o
u	u	vocale posteriore chiusa arrotondata	муж, вьюга	u
ɨ	1	vocale centrale chiusa non arrotondata	мышь	i

Spagnolo (es-ES)

La tabella seguente elenca i fonemi dell'alfabeto fonetico internazionale (International Phonetic Alphabet, IPA), i simboli X-SAMPA (Extended Speech Assessment Methods Phonetic Alphabet) e i corrispondenti visemi per la lingua spagnola supportati da Amazon Polly.

Tabella fonemi/visemi

IPA	X-SAMPA	Descrizione	Esempio	Visema
Consonanti				
ɾ	4	monovibrante alveolare	pero, bravo, amor, eterno	t
b	b	occlusiva bilabiale sonora	bestia	p
β	B	fricativa bilabiale sonora	bebé	B
d	d	occlusiva alveolare sonora	cuando	t
ð	D	fricativa dentale sonora	arder	T

IPA	X-SAMPA	Descrizione	Esempio	Visema
f	f	fricativa labiodentale sorda	fase, café	f
g	g	occlusiva velare sonora	gato, lengua, guerra	k
ɣ	G	fricativa velare sonora	trigo, Argos	k
j	j	approssimante palatale	hacia, tierra, radio, viuda	i
ʝ	j\	fricativa palatale sonora	enhielar, sayo, inyectado, desyerba	J
k	k	occlusiva velare sorda	caña, laca, quisimos	k
l	l	approssimante alveolare laterale	lino, calor, principal	t
ʎ	L	approssimante laterale palatale	llave, pollo	J
m	m	nasale bilabiale	madre, comer, anfibio	p
n	n	nasale alveolare	nido, anillo, sin	t
ɲ	J	nasale palatale	cabaña, ñoquis	J
ŋ	N	nasale velare	cinco, venga	k
p	p	occlusiva bilabiale sorda	pozo, topo	p
r	r	vibrante alveolare	perro, enrachado	r

IPA	X-SAMPA	Descrizione	Esempio	Visema
s	s	fricativa alveolare sorda	saco, casa, puertas	s
t	t	occlusiva alveolare sorda	tamiz, átomo	t
͡tʃ	tS	affricata postalveolare sorda	chubasco	S
θ	T	fricativa dentale sorda	cereza, zorro, lacero, paz	T
w	w	approssimante labiovelare sonora	fuego, fuimos, cuota, cuadro	u
x	x	fricativa velare sorda	jamón, general, suje, reloj	k
z	z	fricativa alveolare sonora	rasgo, mismo	s
Vocali				
a	a	vocale anteriore aperta non arrotondata	tanque	a
e	e	vocale anteriore semichiusa non arrotondata	peso	e
i	i	vocale anteriore chiusa non arrotondata	cinco	i

IPA	X-SAMPA	Descrizione	Esempio	Visema
o	o	vocale posterior e semichiusa arrotondata	bosque	o
u	u	vocale anteriore semichiusa non arrotondata	publicar	u
Simboli aggiuntivi				
ˈ	"	enfasi principale	Alabama	
ˌ	%	enfasi secondaria	Alabama	
.	.	sillabazione	A.la.ba.ma	

Spagnolo (Messico) (es-MX)

La tabella seguente elenca i fonemi dell'alfabeto fonetico internazionale (International Phonetic Alphabet, IPA), i simboli X-SAMPA (Extended Speech Assessment Methods Phonetic Alphabet) e i corrispondenti visemi per la lingua spagnola, Messico supportata da Amazon Polly.

Tabella fonemi/visemi

IPA	X-SAMPA	Descrizione	Esempio	Visema
Consonanti				
ɾ	4	monovibrante alveolare	pero, bravo, amor, eterno	t
b	b	occlusiva bilabiale sonora	bestia	p
β	B	fricativa bilabiale sonora	bebé	B

IPA	X-SAMPA	Descrizione	Esempio	Visema
d	d	occlusiva alveolare sonora	cuando	t
ð	D	fricativa dentale sonora	arder	T
f	f	fricativa labiodentale sorda	fase, café	f
g	g	occlusiva velare sonora	gato, lengua, guerra	k
ɣ	G	fricativa velare sonora	trigo, Argos	k
j	j	approssimante palatale	hacia, tierra, radio, viuda	i
ʝ	j\	fricativa palatale sonora	enhielar, sayo, inyectado, desyerba	J
k	k	occlusiva velare sorda	caña, laca, quisimos	k
l	l	alveolare laterale approssimante	lino, calor, principal	t
m	m	nasale bilabiale	madre, comer, anfibio	p
n	n	nasale alveolare	nido, anillo, sin	t
ɲ	J	nasale palatale	cabaña, ñoquis	J
ŋ	N	nasale velare	angosto, increíble	k

IPA	X-SAMPA	Descrizione	Esempio	Visema
p	p	occlusiva bilabiale sorda	pozo, topo	p
r	r	vibrante alveolare	perro, enrachado	r
s	s	fricativa alveolare sorda	saco, casa, puertas	s
ʃ	S	fricativa postalveolare sorda	show, flash	S
t	t	occlusiva alveolare sorda	tamiz, átomo	t
t͡ʃ	tS	affricata postalveolare sorda	chubasco	S
w	w	approssimante labiovelare sonora	fuego, fuimos, cuota, cuadro	u
x	x	fricativa velare sorda	jamón, general, peaje, reloj	k
z	z	fricativa alveolare sonora	rasgo, mismo	s
Vocali				
a	a	vocale aperta centrale non arrotondata	tanque	a
e	e	vocale anteriore semichiusa non arrotondata	peso	e

IPA	X-SAMPA	Descrizione	Esempio	Visema
i	i	vocale anteriore chiusa non arrotondata	cinco	i
o	o	vocale posterior e semichiusa arrotondata	bosque	o
u	u	vocale posteriore chiusa arrotondata	publicar	u
Simboli aggiuntivi				
ˈ	"	enfasi principale	Alabama	
ˌ	%	enfasi secondaria	Alabama	
.	.	sillabazione	A.la.ba.ma	

Spagnolo (Stati Uniti) (es-US)

La tabella seguente elenca i fonemi dell'alfabeto fonetico internazionale (International Phonetic Alphabet, IPA), i simboli X-SAMPA (Extended Speech Assessment Methods Phonetic Alphabet) e i corrispondenti visemi per la lingua spagnola americana supportati da Amazon Polly.

Tabella fonemi/visemi

IPA	X-SAMPA	Descrizione	Esempio	Visema
Consonanti				
ɾ	4	monovibrante alveolare	pero, bravo, amor, eterno	t
b	b	occlusiva bilabiale sonora	bestia	p

IPA	X-SAMPA	Descrizione	Esempio	Visema
β	B	fricativa bilabiale sonora	bebé	B
d	d	occlusiva alveolare sonora	cuando	t
ð	D	fricativa dentale sonora	arder	T
f	f	fricativa labiodent ale sorda	fase, café	f
g	g	occlusiva velare sonora	gato, lengua, guerra	k
ɣ	G	fricativa velare sonora	trigo, Argos	k
j	j	approssimante palatale	hacia, tierra, radio, viuda	i
ʝ	j\	fricativa palatale sonora	enhielar, sayo, inyectado, desyerba	J
k	k	occlusiva velare sorda	caña, laca, quisimos	k
l	l	alveolare laterale approssimante	lino, calor, principal	t
m	m	nasale bilabiale	madre, comer, anfibio	p
n	n	nasale alveolare	nido, anillo, sin	t
ɲ	J	nasale palatale	cabaña, ñoquis	J

IPA	X-SAMPA	Descrizione	Esempio	Visema
ŋ	N	nasale velare	angosto, increíble	k
p	p	occlusiva bilabiale sorda	pozo, topo	p
r	r	vibrante alveolare	perro, enrachado	r
s	s	fricativa alveolare sorda	saco, casa, puertas	s
ʃ	S	fricativa postalveolare sorda	show, flash	S
t	t	occlusiva alveolare sorda	tamiz, átomo	t
͡tʃ	tS	affricata postalveolare sorda	chubasco	S
w	w	approssimante labiovelare sonora	fuego, fuimos, cuota, cuadro	u
x	x	fricativa velare sorda	jamón, general, peaje, reloj	k
z	z	fricativa alveolare sonora	rasgo, mismo	s
Vocali				
a	a	vocale aperta centrale non arrotondata	tanque	a
e	e	vocale anteriore semichiusa non arrotondata	peso	e

IPA	X-SAMPA	Descrizione	Esempio	Visema
i	i	vocale anteriore chiusa non arrotondata	cinco	i
o	o	vocale posterior e semichiusa arrotondata	bosque	o
u	u	vocale posteriore chiusa arrotondata	publicar	u
Simboli aggiuntivi				
ˈ	"	enfasi principale	Alabama	
ˌ	%	enfasi secondaria	Alabama	
.	.	sillabazione	A.la.ba.ma	

Svedese (sv-SE)

La tabella seguente elenca i fonemi dell'alfabeto fonetico internazionale (International Phonetic Alphabet, IPA), i simboli X-SAMPA (Extended Speech Assessment Methods Phonetic Alphabet) e i corrispondenti visemi per la lingua svedese supportati da Amazon Polly.

Tabella fonemi/visemi

IPA	X-SAMPA	Descrizione	Esempio	Visema
Consonanti				
b	b	occlusiva bilabiale sonora	bil	p
d	d	occlusiva alveolare sonora	dal	t

IPA	X-SAMPA	Descrizione	Esempio	Visema
ɖ	d`	occlusiva retroflessa sonora	bord	t
f	f	fricativa labiodentale sorda	fil	f
g	g	occlusiva velare sonora	gås	k
h	h	fricativa glottidale sorda	hal	k
j	j	approssimante palatale	jag	i
k	k	occlusiva velare sorda	kal	k
l	l	approssimante alveolare laterale	lös	t
ɭ	l`	approssimante laterale retroflessa	härlig	t
m	m	nasale bilabiale	mil	p
n	n	nasale alveolare	nålar	t
ɳ	n`	nasale retroflessa	barn	t
ŋ	N	nasale velare	ring	k
p	p	occlusiva bilabiale sorda	pil	p
r	r	vibrante alveolare	ris	r

IPA	X-SAMPA	Descrizione	Esempio	Visema
s	s	fricativa alveolare sorda	sil	s
ɕ	s\	fricativa alveolopa latale sorda	tjock	J
ʂ	s`	fricativa retroflessa sorda	fors, schlager	S
t	t	occlusiva alveolare sorda	tal	t
ʈ	t`	occlusiva retrofles sa sorda	hjort	t
v	v	fricativa labiodent ale sonora	vår	f
w	w	approssimante labiovelare sonora	aula, airways	u
ɧ	x\	fricativa velare palatale sorda	sjuk	k
Vocali				
ø	2	vocale anteriore semichiusa arrotondata	föll, förr	o
ø	2:	vocale anteriore semichiusa arrotondata allungata	föl, nöt, för	o

IPA	X-SAMPA	Descrizione	Esempio	Visema
ɵ	8	vocale centrale semichiusa arrotondata	buss, full	o
ə	@	vocale centrale media	pojken	@
ʉː	}:	vocale centrale chiusa arrotondata allungata	hus, ful	u
a	a	vocale anteriore aperta non arrotondata	hall, matt	a
æ	{	vocale anteriore quasi aperta non arrotondata	herr	a
ɑː	A:	vocale posterior e aperta non arrotondata allungata	hal, mat	a
eː	e:	vocale anteriore semichiusa non arrotondata allungata	vet, hel	e
ɛ	E	vocale anteriore semiaperta non arrotondata	vett, rätt, hetta, häll	E
ɛː	E:	vocale anteriore semiaperta non arrotondata allungata	säl, häl, här	E:

IPA	X-SAMPA	Descrizione	Esempio	Visema
iː	i:	vocale anteriore chiusa non arrotondata allungata	vit, sil	i:
ɪ	I	vocale quasi anteriore quasi chiusa non arrotondata	vitt, sill	i
oː	o:	vocale posterior e semichius a arrotondata allungata	hål, mål	o
ɔ	O	vocale posterior e semiaperta arrotondata	håll, moll	O
uː	u:	vocale posteriore chiusa arrotondata allungata	sol, bot	u
ʊ	U	vocale quasi posteriore quasi chiusa arrotondata	bott	u
y	y	vocale anteriore chiusa arrotondata	bytt	u
yː	y:	vocale anteriore chiusa arrotondata allungata	syl, syl	u
Simboli aggiuntivi				
ˈ	"	enfasi principale	Alabama	

IPA	X-SAMPA	Descrizione	Esempio	Visema
ˈ	%	enfasi secondaria	Alabama	
.	.	sillabazione	A.la.ba.ma	

Turco (tr-TR)

La tabella seguente elenca i fonemi dell'alfabeto fonetico internazionale (International Phonetic Alphabet, IPA), i simboli X-SAMPA (Extended Speech Assessment Methods Phonetic Alphabet) e i corrispondenti visemi per la lingua turca supportati da Amazon Polly.

Tabella fonemi/visemi

IPA	X-SAMPA	Descrizione	Esempio	Visema
Consonanti				
ɾ	4	monovibrante alveolare	durum	t
t̥ɾ	4_0_r	monovibrante affricata alveolare sorda	bir	t
ɾ	4_r	monovibrante alveolare affricata	raf	t
b	b	occlusiva bilabiale sonora	raf	p
c	c	Occlusiva palatale sorda	kedi	k
d	d	occlusiva alveolare sonora	dede	t
d͡ʒ	dZ	affricata postalveolare sonora	cam	S

IPA	X-SAMPA	Descrizione	Esempio	Visema
f	f	fricativa labiodentale sorda	fare	f
g	g	occlusiva velare sonora	galibi	k
h	h	fricativa glottidale sorda	hasta	k
j	j	approssimante palatale	yat	i
ɟ	J\	occlusiva palatale sonora	genç	J
k	k	occlusiva velare sorda	akıl	k
l	l	approssimante alveolare laterale	lale	t
ɫ	5	laterale alveolare approssimante velarizzata	labirent	t
m	m	nasale bilabiale	maaş	p
n	n	nasale alveolare	anı	t
p	p	occlusiva bilabiale sorda	ip	p
s	s	fricativa alveolare sorda	ses	s
ʃ	S	fricativa postalveolare sorda	aşı	S

IPA	X-SAMPA	Descrizione	Esempio	Visema
t	t	occlusiva alveolare sorda	ütü	t
͡tʃ	tS	affricata postalveolare sorda	çaba	S
v	v	fricativa labiodentale sonora	ekvator, kahveci, akvaryum, isveçli, teşviki, cetvel	f
z	z	fricativa alveolare sonora	ver	s
ʒ	Z	fricativa postalveolare sonora	azık	S
Vocali				
ø	2	vocale anteriore semichiusa arrotondata	göl	0
œ	9	vocale anteriore semiaperta arrotondata	banliyö	O
a	a	vocale anteriore aperta non arrotondata	kal	a
a:	a:	vocale anteriore aperta non arrotondata allungata	davacı	a

IPA	X-SAMPA	Descrizione	Esempio	Visema
æ	{	vocale anteriore quasi aperta non arrotondata	özlem, güvenlik, gürel, somersault	a
e	e	vocale anteriore semichiusa non arrotondata	keçi	e
ɛ	E	vocale anteriore semiaperta non arrotondata	dede	E
i	i	vocale anteriore chiusa non arrotondata	bir	i
iː	i:	vocale anteriore chiusa non arrotondata allungata	izah	i
ɪ	I	vocale quasi anteriore quasi chiusa non arrotondata	keçi	i
ɯ	M	vocale posterior e chiusa non arrotondata	kıl	i
o	o	vocale posterior e semichiusa arrotondata	kol	o

IPA	X-SAMPA	Descrizione	Esempio	Visema
o:	o:	vocale posterior e semichius a arrotondata allungata	dolar	o
u	u	vocale posteriore chiusa arrotondata	durum	u
u:	u:	vocale posteriore chiusa arrotondata allungata	ruhum	u
ʊ	U	vocale quasi posteriore quasi chiusa arrotondata	dolu	u
y	y	vocale anteriore chiusa arrotondata	güvenlik	u
ʏ	Y	vocale quasi anteriore quasi chiusa arrotondata	aşı	u
Simboli aggiuntivi				
ˈ	"	enfasi principale	Alabama	
ˌ	%	enfasi secondaria	Alabama	
.	.	sillabazione	A.la.ba.ma	

Gallese (cy-GB)

La tabella seguente elenca i fonemi dell'alfabeto fonetico internazionale (International Phonetic Alphabet, IPA), i simboli X-SAMPA (Extended Speech Assessment Methods Phonetic Alphabet) e i corrispondenti visemi per la lingua gallese supportati da Amazon Polly.

Tabella fonemi/visemi

IPA	X-SAMPA	Descrizione	Esempio	Visema
Consonanti				
b	b	occlusiva bilabiale sonora	baban	p
d	d	occlusiva alveolare sonora	deg	t
͡dʒ	dZ	affricata postalveolare sonora	garej	S
ð	D	fricativa dentale sonora	deuddeg	T
f	f	fricativa labiodentale sorda	ffacs	f
g	g	occlusiva velare sonora	gadael	k
h	h	fricativa glottidale sorda	haearn	k
j	j	approssimante palatale	astudio	i
k	k	occlusiva velare sorda	cant	k
l	l	approssimante alveolare laterale	lan	t
ɬ	K	fricativa alveolare laterale sorda	llan	t
m	m	nasale bilabiale	mae	p

IPA	X-SAMPA	Descrizione	Esempio	Visema
m̥	m_0	Nasale bilabiale sorda	ymhen	p
n	n	nasale alveolare	naw	t
n̥	n_0	Nasale alveolare sorda	anhawster	t
ŋ	N	nasale velare	argyfwng	k
ŋ̊	N_0	Nasale velare sorda	anghenion	k
p	p	occlusiva bilabiale sorda	pump	p
r	r	vibrante alveolare	rhoi	r
r̥	r_0	Vibrante alveolare sorda	garw	r
s	s	fricativa alveolare sorda	saith	s
ʃ	S	fricativa postalveolare sorda	siawns	S
t	t	occlusiva alveolare sorda	tegan	t
͡tʃ	tS	affricata postalveolare sorda	cytsain	S
θ	T	fricativa dentale sorda	aberth	T
v	v	fricativa labiodentale sonora	prawf	f

IPA	X-SAMPA	Descrizione	Esempio	Visema
w	w	approssimante labiovelare sonora	rhagweld	u
χ	X	fricativa uvulare sorda	chwech	k
z	z	fricativa alveolare sonora	aids	s
ʒ	Z	fricativa postalveolare sonora	rouge	S
Vocali				
ə	@	vocale centrale media	ychwanega	@
a	a	vocale anteriore aperta non arrotondata	acen	a
ai	ai	dittongo	dau	a
au	au	dittongo	awdur	a
ɑː	A:	vocale posteriore aperta non arrotondata allungata	mab	a
ɑːɨ	A:1	dittongo	aelod	a
eː	e:	vocale anteriore semichiusa non arrotondata allungata	peth	e

IPA	X-SAMPA	Descrizione	Esempio	Visema
ɛ	E	vocale anteriore semiaperta non arrotondata	pedwar	E
ɛi	Ei	dittongo	beic	E
iː	i:	vocale anteriore chiusa non arrotondata allungata	tri	i
ɪ	I	vocale quasi anteriore quasi chiusa non arrotondata	miliwn	i
ɨu	1u	dittongo	unigryw	i
oː	o:	vocale posterior e semichius a arrotondata allungata	oddi	o
ɔ	O	vocale posterior e semiaperta arrotondata	oddieithr	O
ɔi	Oi	dittongo	troi	O
ɔu	Ou	dittongo	rownd	O
uː	u:	vocale posteriore chiusa arrotondata allungata	cwch	u

IPA	X-SAMPA	Descrizione	Esempio	Visema
ʊ	U	vocale quasi posteriore quasi chiusa arrotondata	acwstig	u
ʊi	Ui	dittongo	wyth	u
Simboli aggiuntivi				
ˈ	"	enfasi principale	Alabama	
ˌ	%	enfasi secondaria	Alabama	
.	.	sillabazione	A.la.ba.ma	

Motori vocali Amazon Polly

Amazon Polly dispone di quattro motori vocali che convertono il testo di input in un linguaggio realistico. Questi includono: generativo, in forma lunga, neurale e standard. Per usare una voce Amazon Polly, seleziona un motore e un'operazione API di sintesi vocale. Quindi fornisci il testo di input che il motore deve sintetizzare e seleziona un formato di uscita audio. Grazie a questi input, Amazon Polly sintetizza il testo fornito in un flusso audio vocale di alta qualità.

Le seguenti sezioni includono dettagli sui motori vocali offerti da Amazon Polly.

Argomenti

- Voci generative
- Voci a forma lunga
- Voci neurali
- Voci standard
- Scelta di un motore vocale

Voci generative

Il motore generativo text-to-speech (TTS) di Amazon Polly offre le voci conversazionali più simili a quelle umane, emotivamente coinvolgenti e adattive disponibili per l'uso tramite la console Amazon Polly.

Il motore Generative è il modello Amazon Polly TTS più grande mai realizzato fino ad oggi. Implementa un trasformatore da miliardi di parametri che converte il testo non elaborato in codici vocali, seguito da un decodificatore basato sulla convoluzione che converte questi codici vocali in forme d'onda in modo incrementale e semplificato. Questo metodo mostra le capacità emergenti ampiamente diffuse dei Large Language Models (LLMs) quando vengono addestrati su volumi crescenti di dati proprietari e disponibili al pubblico che comprendono una varietà di voci, lingue e stili.

Il motore generativo crea un linguaggio sintetico che è emotivamente coinvolgente, assertivo e altamente colloquiale in un modo notevolmente simile a una voce umana. Puoi usare queste voci come un assistente clienti esperto, un trainer virtuale o un inserzionista con un linguaggio sintetico quasi umano.

> **Note**
>
> La state-of-the-art tecnologia alla base di queste voci rientra nel paradigma dell'intelligenza artificiale generativa per la modellazione del linguaggio e della voce. Un effetto collaterale della tecnologia è che qualsiasi aggiornamento ai dati di addestramento e al modello potrebbe comportare lievi variazioni nel modo in cui le voci suonano, anche nel caso in cui la loro qualità complessiva migliori con gli aggiornamenti del modello. Ciò potrebbe avere un impatto sui casi d'uso con diverse parti di contenuto sintetizzate per un lungo periodo di tempo, ad esempio una stagione di podcast.

Voci generative disponibili

Amazon Polly offre attualmente 20 voci in una variante generativa. Queste voci generative sono disponibili anche in una variante NTTS conversazionale.

	Lingua	Codice lingua	Nome/ID	Gender
1	Inglese (Australia)	en-AU	Olivia	Femmina
2	Inglese (India)	en-IN	Kajal	Femmina
3	Inglese (Regno Unito)	en-GB	Amy	Femmina
4	Inglese (Stati Uniti)	it-IT	Danielle	Femmina
5	Inglese (Stati Uniti)	it-IT	Joanna	Femmina
6	Inglese (Stati Uniti)	it-IT	Matthew	Maschio
7	Inglese (Stati Uniti)	it-IT	Ruth**	Femmina

	Lingua	Codice lingua	Nome/ID	Gender
8	Inglese (Stati Uniti)	it-IT	Stephen**	Maschio
9	Inglese (Sud Africa)	en-ZA	Ayanda	Femmina
10	Francese (Francia)	fr-FR	Léa	Femmina
11	Francese (Francia)	fr-FR	Rémi**	Maschio
12	Spagnolo (Spagna)	es-ES	Lucia	Femmina
13	Spagnolo (Spagna)	es-ES	Sergio**	Maschio
14	Spagnolo (messicano)	es-MX	Mía	Femmina
15	Spagnolo (messicano)	es-MX	Andres**	Maschio
16	Spagnolo (Stati Uniti)	es-US	Lupe	Femmina
17	Spagnolo (Stati Uniti)	es-US	Pedro	Maschio
18	Tedesco (Germania)	de-DE	Vicki	Femmina
19	Tedesco (Germania)	de-DE	Daniele	Maschio
20	Italiano (Italia)	it-IT	Bianca	Femmina

> **Note**
>
> Il costo delle voci generative è specificato nella pagina di informazioni sui prezzi di Amazon Polly.

Compatibilità tra funzionalità e aree geografiche

Le voci generative di Amazon Polly sono disponibili nelle seguenti regioni:

- Stati Uniti orientali (Virginia settentrionale): us-east-1
- Europa (Francoforte): eu-central-1
- Stati Uniti occidentali (Oregon): us-west-2
- Altre regioni non sono disponibili

Le seguenti funzionalità sono supportate per le voci generative:

- Operazioni di sintesi vocale asincrona e in tempo reale.
- Lo stile di pronuncia di Newscaster non è supportato nel motore Generative.
- Molti tag SSML (ma non tutti) sono supportati da Amazon Polly. Per ulteriori informazioni sui tag SSML supportati da NTTS, consulta Tag SSML supportati
- Come per le voci standard, puoi scegliere tra diverse frequenze di campionamento per ottimizzare la larghezza di banda e la qualità audio della tua applicazione. Le frequenze di campionamento valide per le voci standard e neurali sono 8 kHz, 16 kHz, 22 kHz o 24 kHz. Il valore predefinito per le voci standard è 22 kHz. L'impostazione predefinita per le voci generative è 24 kHz. Amazon Polly supporta i formati di MP3 streaming audio OGG (Vorbis) e PCM non elaborati.

Il supporto per la generazione di contrassegni vocali non è attualmente disponibile.

> **Note**
>
> Nell'improbabile eventualità che si verifichi un'allucinazione del modello (e con il comportamento del motore Generative che prevede la riproduzione vocale token per token), viene imposto un meccanismo di arresto di emergenza. Il meccanismo integrato impedisce al modello di riprodurre ulteriormente il parlato. Questa funzionalità di sicurezza si basa sull'analisi dei dati in cui il modello può generare allucinazioni, di solito alla fine della frase.

Potrebbero verificarsi casi in cui il modello pensa di provocare allucinazioni e poi finisce per tagliare una parola durante una fase generazionale, restituendo così metà della parola. Ciò potrebbe generare risultati inappropriati.

Voci a forma lunga

Amazon Polly ha un motore a forma lunga che produce voci simili a quelle umane, altamente espressive ed emotivamente abili. Le voci lunghe sono progettate per attirare l'attenzione degli ascoltatori quando si tratta di contenuti più lunghi, come articoli di notizie, materiali di formazione o video di marketing.

Le voci lunghe di Amazon Polly sono sviluppate con una tecnologia TTS di deep learning all'avanguardia. Il modello impara a replicare fonemi, prosodia, intonazione e altri aspetti fonetici e acustici del linguaggio umano, ottenendo un output vocale estremamente naturale.

Il motore Long-form utilizza gli incorporamenti di testo per interpretare il significato di un testo. Utilizzando gli incorporamenti di testo, il motore Long-form può generare l'enfasi, le pause e il tono corretti di una voce naturale. Il risultato è una voce che combina la gamma completa di elementi emotivi presenti nella comunicazione umana. Ciò include l'imitazione del dialogo a sorpresa o la differenziazione del dialogo dalla narrazione. Insieme, questo crea un prodotto vocale di alta qualità che sembra un essere umano vivo.

Note

La state-of-the-art tecnologia alla base di queste voci rientra nel paradigma dell'intelligenza artificiale generativa per la modellazione del linguaggio e della voce. Un effetto collaterale della tecnologia è che qualsiasi aggiornamento ai dati di addestramento e al modello potrebbe comportare lievi variazioni nel modo in cui le voci suonano, anche nel caso in cui la loro qualità complessiva migliori con gli aggiornamenti del modello. Ciò potrebbe avere un impatto sui casi d'uso con diverse parti di contenuto sintetizzate per un lungo periodo di tempo, ad esempio una stagione di podcast.

Voci lunghe disponibili

Amazon Polly attualmente offre quattro voci in formato esteso en-US e due es-ES. In entrambe le lingue sono disponibili voci femminili e maschili. Le voci lunghe inglesi Daniel, Gregory e Ruth sono disponibili anche in una variante conversazionale NTTS.

	Lingua	Codice lingua	Nome/ID	Gender
1	Inglese (Stati Uniti)	it-IT	Danielle	Femmina
			Gregory	Maschio
			Ruth**	Femmina
			Patrizio	Maschio
2	Spagnolo (Spagna)	es-ES	Alba	Femmina
			Raúl	Maschio

Compatibilità tra funzionalità e aree geografiche

Le voci lunghe di Amazon Polly sono disponibili nelle seguenti regioni:

- Stati Uniti orientali (Virginia settentrionale): us-east-1
- Altre regioni non disponibili

Il motore Amazon Polly Long-form supporta le seguenti funzionalità:

- Operazioni di sintesi vocale asincrona e in tempo reale.
- Tutti i segni vocali.
- Molti tag SSML (ma non tutti) sono supportati da Amazon Polly. Per ulteriori informazioni sui tag SSML supportati da NTTS, consulta Tag SSML supportati
- Come per le voci standard, puoi scegliere tra diverse frequenze di campionamento per ottimizzare la larghezza di banda e la qualità audio della tua applicazione. Le frequenze di campionamento valide per voci standard, di forma lunga e neurali sono: 8 kHz, 16 kHz, 22 kHz o 24 kHz. Il valore predefinito per le voci standard è 22 kHz. L'impostazione predefinita per le voci lunghe e neurali

è 24 kHz. Amazon Polly supporta i formati di MP3 streaming audio OGG (Vorbis) e PCM non elaborati.

> **Note**
>
> Il costo delle voci in formato lungo è specificato nella pagina di informazioni sui prezzi di Amazon Polly.

Voci neurali

Amazon Polly dispone di un motore Neural text-to-speech (NTTS) in grado di produrre voci di qualità ancora superiore rispetto alle sue voci standard. Le voci TTS standard utilizzano la sintesi concatenativa. Il motore standard concatena i fonemi del parlato registrato, producendo un parlato sintetizzato dal suono molto naturale. Tuttavia, le inevitabili variazioni della voce e le tecniche utilizzate per segmentare le forme d'onda limitano la qualità della voce. Il motore Amazon Polly NTTS non utilizza la sintesi concatenativa standard per produrre il parlato. È costituito da due parti:

- Una rete neurale, che converte una sequenza di fonemi (le unità linguistiche più elementari) in una sequenza di spettrogrammi. (Gli spettogrammi sono istantanee dei livelli di energia in diverse bande di frequenza.)
- Un vocoder, che converte gli spettrogrammi in un segnale audio quasi continuo.

Il primo componente del sistema TTS neurale è un modello. sequence-to-sequence Questo modello non crea i risultati solo dall'input corrispondente, ma tiene in considerazione anche il modo in cui interagisce la sequenza degli elementi dell'input. Il modello sceglie gli spettrogrammi che emette in modo che le bande di frequenza enfatizzino le caratteristiche acustiche che il cervello umano utilizza durante l'elaborazione della sintesi vocale.

L'output di questo modello passa quindi a un vocoder neurale che converte gli spettrogrammi in forme d'onda della sintesi vocale. Se addestrato sui grandi set di dati utilizzati per creare sistemi di sintesi concatenativa generici, questo approccio produrrà voci di qualità superiore e dal suono più naturale. sequence-to-sequence

Voci neurali disponibili

Le voci neurali sono disponibili in 36 lingue e varianti linguistiche. Le voci sono elencate nella tabella seguente.

	Lingua e varianti linguistiche	Codice lingua	Nome/ID	Gender
1	Arabo (Golfo)	ar-AE	Hala Zayd	Femmina Maschio
2	Olandese belga (fiammingo)	nl-BE	Lisa	Femmina
3	Catalano	ca-ES	Arlet	Femmina
4	Ceco	CS-CZ	Jitka	Femmina
5	Cinese (Cantonese)	yue-CN	Hiujin	Femmina
6	Cinese (Mandarino)	cmn-CN	Zhiyu	Femmina
7	Danese	da-DK	Sofie	Femmina
8	Olandese	nl-NL	Laura	Femmina
9	Inglese (Australia)	en-AU	Olivia	Femmina
10	Inglese (Regno Unito)	en-GB	Amy* Emma Brian Arturo	Femmina Femmina Maschio Maschio
11	Inglese (India)	en-IN	Kajal	Femmina

	Lingua e varianti linguistiche	Codice lingua	Nome/ID	Gender
12	Inglese (irlandese)	en-IN	Niamh	Femmina
13	Inglese (Nuova Zelanda)	en-NZ	Aria	Femmina
14	Inglese (singaporiano)	it-SG	Gelsomino	Femmina
15	Inglese (Sud Africa)	en-ZA	Ayanda	Femmina

	Lingua e varianti linguistiche	Codice lingua	Nome/ID	Gender
16	Inglese (Stati Uniti)	it-IT	Danielle	Femmina
			Gregory	Maschio
			Ivy	Femmina (bambina)
			Joanna*	Femmina
			Kendra	Femmina
			Kimberly	Femmina
			Salli	Femmina
			Joey	Maschio
			Justin	Maschio (bambino)
			Kevin	Maschio (bambino)
			Matthew*	Maschio
			Ruth**	Femmina
			Stephen**	Maschio
17	Finlandese	fi-FI	Suvi	Femmina
18	Francese (belga)	fr-BE	Isabelle	Femmina
19	Francese (Canada)	fr-CA	Gabrielle	Femmina
			Liam	Maschio
20	Francese	fr-FR	Léa	Femmina
			Rémi**	Maschio

	Lingua e varianti linguistiche	Codice lingua	Nome/ID	Gender
21	Tedesco	de-DE	Vicki Daniele	Femmina Maschio
22	Tedesco (austriaco)	de-AT	Hannah	Femmina
23	Tedesco (svizzero)	de-CH	Sabrina	Femmina
24	Hindi	hi-IN	Kajal	Femmina
25	Italiano	it-IT	Bianca Adriano**	Femmina Maschio
26	Giapponese	ja-JP	Takumi Kazuha Tomoko	Maschio Femmina Femmina
27	Coreano	ko-KR	Seoyeon Jihye	Femmina Femmina
28	Norvegese	nb-NO	Ida	Femmina
29	Polacco	pl-PL	Ola	Femmina
30	Portoghese (Brasile)	pt-BR	Camila Vitória/Vitoria Thiago**	Femmina Femmina Maschio
31	Portoghese (Europa)	pt-PT	Inês/Ines	Femmina

	Lingua e varianti linguistiche	Codice lingua	Nome/ID	Gender
32	Spagnolo (Spagna)	es-ES	Lucia Sergio**	Femmina Maschio
33	Spagnolo (messicano)	es-MX	Mia Andres**	Femmina Maschio
34	Spagnolo (Stati Uniti)	es-US	Lupe* Pedro	Femmina Maschio
35	Svedese	sv-SE	Elin	Femmina
36	Turco	tr-TR	Burcu	Femmina

*Le voci di Amy, Joanna, Lupe, e Matthew possono essere utilizzate con gli stili di pronuncia Newscaster. Per ulteriori informazioni, consulta Applicare la voce del giornalista.

Compatibilità tra funzionalità e regioni

Le voci neurali non sono disponibili in tutte le AWS regioni, né supportano tutte le funzionalità di Amazon Polly.

Le voci neurali sono supportate nelle seguenti regioni:

- Stati Uniti orientali (Virginia settentrionale): us-east-1
- Stati Uniti occidentali (Oregon): us-west-2
- Africa (Città del Capo): af-south-1
- Asia Pacifico (Tokyo): ap-northeast-1
- Asia Pacifico (Seoul): ap-northeast-2
- Asia Pacifico (Osaka): ap-northeast-3
- Asia Pacifico (Mumbai): ap-south-1
- Asia Pacifico (Singapore): ap-southeast-1

- Asia Pacifico (Sydney): ap-southeast-2
- Asia Pacifico (Malesia): ap-southeast-5
- Canada (Centrale): ca-central-1
- Europa (Francoforte): eu-central-1
- Europa (Irlanda): eu-west-1
- Europa (Londra): eu-west-2
- Europe (Parigi): eu-west-3
- Europa (Spagna): eu-south-2
- AWS GovCloud (Stati Uniti occidentali): -1 us-gov-west

Gli endpoint e i protocolli per queste regioni sono identici a quelli utilizzati per le voci standard. Per ulteriori informazioni, consulta Endpoint ed quote di Amazon Polly.

Le seguenti caratteristiche sono supportate per le voci neurali:

- Operazioni di sintesi vocale asincrona e in tempo reale.
- Stile di pronuncia newscaster. Per ulteriori informazioni sullo stile di pronuncia, consulta Applicare la voce del giornalista.
- Tutti i segni vocali.
- Molti (ma non tutti) i tag SSML supportati da Amazon Polly. Per ulteriori informazioni sui tag SSML supportati da NTTS, consulta Tag supportati.

Come per le voci standard, puoi scegliere tra diverse frequenze di campionamento per ottimizzare la larghezza di banda e la qualità audio della tua applicazione. Le frequenze di campionamento valide per le voci standard e neurali sono 8 kHz, 16 kHz, 22 kHz o 24 kHz. Il valore predefinito per le voci standard è 22 kHz. Il valore predefinito per le voci neurali è 24 kHz. Amazon Polly supporta i formati di MP3 streaming audio OGG (Vorbis) e PCM non elaborati.

Voci standard

Amazon Polly dispone di un motore standard che utilizza la sintesi concatenativa. Il motore standard concatena i fonemi del parlato registrato, producendo un parlato sintetizzato dal suono molto naturale.

Voci standard disponibili

Amazon Polly attualmente offre 40 voci standard femminili e 20 maschili in 29 lingue e varianti linguistiche.

	Lingua	Codice lingua	Nome/ID	Gender
1	Arabo	arb	Zeina	Femmina
2	Cinese (Mandarino)	cmn-CN	Zhiyu	Femmina
3	Danese	da-DK	Naja Mads	Femmina Maschio
4	Olandese	nl-NL	Lotte Ruben	Femmina Maschio
5	Inglese (Australia)	en-AU	Nicole Russell	Femmina Maschio
6	Inglese (Regno Unito)	en-GB	Amy Emma Brian	Femmina Femmina Maschio
7	Inglese (India)	en-IN	Aditi Raveena	Femmina Femmina
8	Inglese (Stati Uniti)	it-IT	Ivy Joanna Kendra Kimberly	Femmina Femmina Femmina Femmina

	Lingua	Codice lingua	Nome/ID	Gender
			Salli	Femmina
			Joey	Maschio
			Kevin	Maschio
9	Inglese (Galles)	en-GB-WLS	Geraint	Maschio
10	Francese	fr-FR	Céline/Celine	Femmina
			Léa	Femmina
			Mathieu	Maschio
11	Francese (Canada)	fr-CA	Chantal	Femmina
12	Tedesco	de-DE	Marlene	Femmina
			Vicki	Femmina
			Hans	Maschio
13	Hindi	hi-IN	Aditi	Femmina
14	Islandese	is-IS	Dóra/Dora	Femmina
			Karl	Maschio
15	Italiano	it-IT	Carla	Femmina
			Bianca	Femmina
			Giorgio	Maschio
16	Giapponese	ja-JP	Mizuki	Femmina
			Takumi	Maschio
17	Coreano	ko-KR	Seoyeon	Femmina

	Lingua	Codice lingua	Nome/ID	Gender
18	Norvegese	nb-NO	Liv	Femmina
19	Polacco	pl-PL	Ewa	Femmina
			Maja	Femmina
			Jacek	Maschio
			Jan	Maschio
20	Portoghese (Brasile)	pt-BR	Camila	Femmina
			Vitória/Vitoria	Femmina
			Ricardo	Maschio
21	Portoghese (Europa)	pt-PT	Inês/Ines	Femmina
			Cristiano	Maschio
22	Romeno	ro-RO	Carmen	Femmina
23	Russo	ru-RU	Tatyana	Femmina
			Maxim	Maschio
24	Spagnolo (Spagna)	es-ES	Conchita	Femmina
			Lucia	Femmina
			Enrique	Maschio
25	Spagnolo (messicano)	es-MX	Mia	Femmina

	Lingua	Codice lingua	Nome/ID	Gender
26	Spagnolo (Stati Uniti)	es-US	Lupe Penélope/ Penelope Miguel	Femmina Femmina Maschio
27	Svedese	sv-SE	Astrid	Femmina
28	Turco	tr-TR	Filiz	Maschio
29	Gallese	cy-GB	Gwyneth	Femmina

Compatibilità tra funzionalità e aree geografiche

Le voci standard di Amazon Polly sono disponibili nelle seguenti regioni Amazon Polly:

- Stati Uniti orientali (Virginia settentrionale): us-east-1
- Stati Uniti orientali (Ohio): us-east-2
- Stati Uniti occidentali (California settentrionale): us-west-1
- Stati Uniti occidentali (Oregon): us-west-2
- Africa (Città del Capo): af-south-1
- Asia Pacifico (Hong Kong): ap-east-1
- Asia Pacifico (Tokyo): ap-northeast-1
- Asia Pacifico (Seoul): ap-northeast-2
- Asia Pacifico (Osaka): ap-northeast-3
- Asia Pacifico (Mumbai): ap-south-1
- Asia Pacifico (Singapore): ap-southeast-1
- Asia Pacifico (Sydney): ap-southeast-2
- Asia Pacifico (Malesia): ap-southeast-5
- Cina (Ningxia): cn-northwest-1;
- Canada (Centrale): ca-central-1

- Europa (Francoforte): eu-central-1
- Europa (Irlanda): eu-west-1
- Europa (Londra): eu-west-2
- Europe (Parigi): eu-west-3
- Europa (Spagna): eu-south-2
- Europa (Stoccolma): eu-north-1
- Medio Oriente (Bahrein): me-south-1
- America del Sud (San Paolo): sa-east-1
- AWS GovCloud (Stati Uniti occidentali): -1 us-gov-west

Gli endpoint e i protocolli per queste regioni sono identici a quelli utilizzati per le voci neurali. Per ulteriori informazioni, consulta Endpoint ed quote di Amazon Polly.

Il motore standard Amazon Polly supporta le seguenti funzionalità (TBD):

- Operazioni di sintesi vocale asincrona e in tempo reale.
- Tutti i segni vocali.
- Molti tag SSML (ma non tutti) sono supportati da Amazon Polly. Per ulteriori informazioni sui tag SSML supportati da NTTS, consulta Tag SSML supportati.
- È possibile scegliere tra diverse frequenze di campionamento per ottimizzare la larghezza di banda e la qualità audio per l'applicazione. Le frequenze di campionamento predefinite per le voci standard sono 22 kHz. Amazon Polly supporta i formati di MP3 streaming audio OGG (Vorbis) e PCM non elaborati.

> **Note**
>
> Il costo delle voci standard è specificato nella pagina di informazioni sui prezzi di Amazon Polly.

Scelta di un motore vocale

Puoi accedere alle voci di Amazon Polly tramite la console Amazon Polly oppure. AWS CLI

Per scegliere un motore vocale sulla console

1. Apri la console Amazon Polly all'indirizzo. https://console.aws.amazon.com/polly/
2. Dalla console Amazon Polly, scegli il motore vocale desiderato.
3. Scegli la voce desiderata dal menu a discesa vocale.
4. Genera audio TTS con testo a tua scelta.

Per scegliere un motore vocale in AWS CLI, specifica `Engine` e `VoiceId` nelle `SyntheszieSpeech` operazioni `StartSpeechSynthesisTask` API. Per alcuni esempi, consulta gli esempi di codice di avvio rapido e gli esempi in Python.

Segni vocali

I contrassegni vocali sono metadata che descrivono la sintesi vocale, ad esempio il punto in cui una frase o parola inizia e finisce nel flusso audio. Quando richiedi contrassegni vocali per il tuo testo, Amazon Polly restituisce questi metadati anziché la sintesi vocale. Utilizzando i contrassegni vocali in combinazione con il flusso audio della sintesi vocale, è possibile offrire alle tue applicazioni un'esperienza visiva ottimizzata.

Ad esempio, la combinazione di metadati con il flusso audio del tuo testo consente di sincronizzare la sintesi vocale con le animazioni del volto (sincronia labiale) o di evidenziare le parole scritte appena vengono pronunciate.

I segni vocali sono disponibili quando si utilizzano motori neurali, di forma lunga o standard. text-to-speech

Argomenti

- Tipi di segni vocali
- Visemi e Amazon Polly
- Uscita del contrassegno vocale
- Richiesta di segni vocali
- Esempio di segni vocali senza SSML
- Esempio di segni vocali con SSML

Tipi di segni vocali

È possibile richiedere i contrassegni vocali utilizzando l'SpeechMarkTypesopzione relativa ai StartSpeechSynthesisTaskcomandi SynthesizeSpeecho. Vengono specificati gli elementi di metadata che desideri vengano restituiti dal testo di input. Puoi richiedere quattro tipi di metadata ma devi specificarne almeno uno per richiesta. Non viene generato alcun output audio con la richiesta.

In AWS CLI, ad esempio:

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

Amazon Polly genera contrassegni vocali utilizzando i seguenti elementi:

- sentence (frase): indica un elemento frase nel testo di input.
- word (parola): indica un elemento parola nel testo di input.
- viseme (visema): descrive i movimenti di volto e bocca corrispondenti a ciascun fonema pronunciato. Per ulteriori informazioni, consulta Visemi e Amazon Polly.
- ssml: descrive un elemento <mark> del testo di input SSML. Per ulteriori informazioni, consulta Generazione vocale da documenti SSML.

Visemi e Amazon Polly

Un visema rappresenta la posizione del volto e della bocca quando si pronuncia una parola. È l'equivalente visivo di un fonema, vale a dire l'unità acustica di base da cui è formata una parola. I visemi sono gli elementi costitutivi visivi di base della lingua parlata.

Ogni lingua ha un insieme di visemi che corrispondono ai loro fonemi specifici. In una lingua, a ciascun fonema corrisponde un visema che rappresenta la forma delineata dalla bocca nel formare il suono. Tuttavia, non tutti i visemi possono essere mappati a un determinato fonema perché numerosi fonemi appaiono identici quando pronunciati, sebbene il suono sia diverso. Ad esempio, in inglese, i termini "pet" e "bet" sono acusticamente differenti. Tuttavia, quando osservati visivamente (senza suono), sembrano identici.

Nel grafico seguente viene visualizzato un elenco parziale dei fonemi dell'alfabeto fonetico internazionale (IPA) e dei simboli dell'alfabeto X-SAMPA (Extended Speech Assessment Methods Phonetic Alphabet), nonché i visemi corrispondenti per le voci inglesi degli Stati Uniti.

Per la tabella completa e le tabelle per tutte le lingue disponibili, consultare Lingue in Amazon Polly.

IPA	X-SAMPA	Descrizione	Esempio	Visema
Consonanti				
b	b	Occlusiva bilabiale sonora	bed	p
d	d	Occlusiva alveolare sonora	dig	t
͡dʒ	dZ	Affricata postalveolare sonora	jump	S

IPA	X-SAMPA	Descrizione	Esempio	Visema
ð	D	Fricativa dentale sonora	poi	T
f	f	Fricativa labiodentale sorda	five	f
g	g	Occlusiva velare sonora	game	k
h	h	Fricativa faringale sorda	house	k
...	...	...	...	...

Uscita del contrassegno vocale

Amazon Polly restituisce gli oggetti di contrassegni vocali in un flusso JSON delimitato da righe. Un oggetto di contrassegno vocale contiene i seguenti campi:

- time (tempo): il timestamp in millisecondi dall'inizio del flusso audio corrispondente
- type (tipo): il tipo di contrassegno vocale (frase, parola, visema o ssml).
- start (avvio): lo spostamento in byte (non caratteri) dell'inizio dell'oggetto nel testo di input (non include i contrassegni visema)
- end (fine): lo spostamento in byte (non caratteri) della fine dell'oggetto nel testo di input (non include i contrassegni visema)
- value (valore): questo varia a seconda del tipo di contrassegno vocale
  - SSML: tag SSML <mark>
  - viseme (visema): nome del visema
  - word (parola) o sentence (frase): una sottostringa del testo di input delimitata dai campi di inizio e fine

Ad esempio, Amazon Polly genera il seguente oggetto di contrassegno vocale `word` dal testo "Mary had a little lamb":

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

La parola descritta ("had") comincia 373 millisecondi dopo l'inizio del flusso audio, nonché inizia al byte 5 e finisce al byte 8 del testo di input.

> **Note**
>
> Questi metadata sono relativi all'ID vocale `Joanna`. Se utilizzi un'altra voce con lo stesso testo di input, i metadata possono variare.

Richiesta di segni vocali

Puoi utilizzare la console o il `synthesize-speech` comando per richiedere i contrassegni vocali da Amazon Polly. Puoi quindi visualizzare i metadata o salvarli in un file.

Console

Per generare contrassegni vocali sulla console

1. Accedi a AWS Management Console e apri la console Amazon Polly all'indirizzo. https://console.aws.amazon.com/polly/
2. Scegli la scheda Text-to-Speech (Sintesi vocale).
3. Attiva SSML per usare SSML.
4. Digita o incolla il testo nella casella di input.
5. Per Lingua, scegli la lingua del testo.
6. Per Voce, scegli la voce che desideri usare.
7. Per modificare la pronuncia del testo, espandi Additional settings (Impostazioni aggiuntive), attiva Customize pronunciation (Personalizza la pronuncia), e per Apply lexicon (Applica il lessico), scegli il lessico desiderato.
8. Per verificare il parlato, scegli Ascolta.
9. Attiva Speech file format settings (Impostazioni formato di file vocale).

   > **Note**
   >
   > I formati di download MP3, OGG o PCM non genereranno contrassegni vocali.

10. In File Format (Formato file), scegli Speech marks (Contrassegni vocali).
11. In Speech mark types (Tipi di contrassegni vocali), scegli i tipi di contrassegni vocali da generare. L'opzione per la scelta dei metadata SSML è disponibile solo quando SSML è attivo. Per ulteriori informazioni sull'uso di SSML con Amazon Polly; consulta Generazione vocale da documenti SSML.
12. Scegli Download (Scarica).

AWS CLI

Oltre al testo di input, per restituire questi metadati sono necessari i seguenti elementi:

- `output-format`

  Amazon Polly supporta solo il formato JSON quando restituisce i contrassegni vocali.

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

  Se utilizzi un formato di output non supportato, Amazon Polly genera un'eccezione.

- `voice-id`

  Per garantire che i metadata corrispondano al flusso audio associato, specifica la stessa voce utilizzata per generare il flusso audio della sintesi vocale. Le voci disponibili non hanno velocità identiche. Se utilizzi una voce diversa da quella utilizzata per generare la sintesi vocale, i metadata non corrisponderanno al flusso audio.

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

- `speech-mark-types`

  Specifica il tipo o i tipi di contrassegni vocali desiderati. Puoi richiedere alcuni o tutti i tipi di contrassegni vocali, ma devi specificare almeno un tipo.

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

- `text-type`

  Il testo semplice è il testo di input predefinito per Amazon Polly, perciò devi utilizzare `text-type ssml` se desideri che vengano restituiti contrassegni vocali SSML.

- `outfile`

  Specifica il file di output in cui sono scritti i metadata.

  ```
  MaryLamb.txt
  ```

L' AWS CLI esempio seguente è formattato per Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione Unix barra rovesciata (\) al termine di ogni riga con un accento circonflesso (^) e usa virgolette (") attorno al testo di input con virgolette singole (') per i tag interni.

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

Esempio di segni vocali senza SSML

L'esempio seguente mostra l'aspetto dei metadata richiesti sullo schermo per la frase semplice: "Mary had a little lamb". Per semplicità, non sono inclusi contrassegni vocali SSML in questo esempio.

L' AWS CLI esempio seguente è formattato per Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione Unix barra rovesciata (\) al termine di ogni riga con un accento circonflesso (^) e usa virgolette (") attorno al testo di input con virgolette singole (') per i tag interni.

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

Quando si effettua questa richiesta, Amazon Polly restituisce i seguenti elementi nel 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"}
```

In questo output, ogni parte del testo è suddivisa a livello di contrassegni vocali:

- La frase "Mary had a little lamb".
- Ogni parola del testo: "Mary", "had", "a", "little" e "lamb".
- Il visema di ogni suono nel flusso audio corrispondente: "p", "E", "r", "i" e così via. Per ulteriori informazioni sui visemi, consulta Visemi e Amazon Polly.

Esempio di segni vocali con SSML

Il processo di generazione dei contrassegni vocali da un testo ottimizzato per SSML è analogo al processo utilizzato quando SSML non è presente. Utilizza il comando `synthesize-speech` e specifica il testo ottimizzato per SSML e il tipo di contrassegni vocali che desideri, come mostrato nell'esempio seguente. Per facilitare la lettura dell'esempio, non includiamo i segni vocali di Viseme, ma potrebbero essere inclusi anche questi.

L' AWS CLI esempio seguente è formattato per Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione Unix barra rovesciata (\) al termine di ogni riga con un accento circonflesso (^) e usa virgolette (") attorno al testo di input con virgolette singole (') per i tag interni.

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

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

Quando si effettua questa richiesta, Amazon Polly restituisce i seguenti elementi nel 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"}
```

Generazione vocale da documenti SSML

È possibile utilizzare Amazon Polly per generare input vocale da testo semplice o documenti contrassegnati con Speech Synthesis Markup Language (SSML). L'utilizzo di testo ottimizzato per SSML fornisce ulteriore controllo sul modo in cui Amazon Polly genera la sintesi vocale dal testo fornito.

Con i tag SSML, è possibile personalizzare e controllare diversi aspetti del discorso restituito, ad esempio pronuncia, volume e velocità. In AWS Management Console, il testo migliorato con SSML che si desidera convertire in audio viene immesso nella scheda SSML della pagina. Text-to-Speech Sebbene il testo inserito come testo normale si basi su impostazioni predefinite per la lingua e la voce scelte, il testo ottimizzato per SSML comunica a Amazon Polly non solo ciò che si desidera dire, ma anche il modo in cui dirlo. Ad eccezione dell'aggiunta dei tag SSML, Amazon Polly sintetizza il testo ottimizzato per SSML nello stesso modo in cui sintetizza il testo normale. Per ulteriori informazioni, consulta Esempio di sintesi vocale con Amazon Polly.

Quando si utilizza SSML, l'intero testo viene racchiuso in un tag `<speak>` per comunicare a Amazon Polly che si sta utilizzando SSML. Ad esempio:

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

Si possono quindi utilizzare tag SSML specifici sul testo all'interno dei tag `<speak>` per personalizzare il modo in cui si desidera che il testo suoni. È possibile aggiungere una pausa, modificare il ritmo del discorso, ridurre o aumentare il volume della voce o aggiungere molte altre personalizzazioni in modo che il testo corrisponda alle aspettative. Per un elenco completo dei tag SSML utilizzabili consultare Tag SSML supportati.

Ad esempio, è possibile includere una lunga pausa all'interno del testo o cambiando la velocità o l'intonazione della sintesi vocale. Altre opzioni includono:

- enfatizzare parole o frasi specifiche
- utilizzare la pronuncia fonetica
- includere i suoni respiratori
- sussurrare
- utilizzare lo stile di pronuncia newscaster.

Per dettagli completi sui tag SSML supportati da Amazon Polly e su come utilizzarli, consulta Tag SSML supportati

Quando si utilizza SSML, ci sono diversi caratteri riservati che richiedono un trattamento speciale. Questo perché SSML utilizza questi caratteri come parte del suo codice. Per utilizzarli, usa un'entità specifica come carattere di escape. Per ulteriori informazioni, consulta Caratteri riservati in SSML

Amazon Polly fornisce questo tipo di controllo con un sottoinsieme di tag di markup SSML definiti da Speech Synthesis Markup Language (SSML) Version 1.1, W3C Recommendation (Speech Synthesis Markup Language (SSML) versione 1.1, specifiche W3C).

È possibile utilizzare SSML nella console di Amazon Polly o tramite AWS CLI. I seguenti argomenti illustrano come è possibile utilizzare SSML per generare discorso e controllare l'output in modo che risponda precisamente alle tue esigenze.

Argomenti

- Caratteri riservati in SSML
- Utilizzo di SSML sulla console
- Utilizzo di SSML con il comando Synthesize-Speech
- Sintetizzazione di un documento ottimizzato con SSML
- Tag SSML supportati

Caratteri riservati in SSML

Esistono cinque caratteri predefiniti che normalmente non possono essere utilizzati all'interno di un'istruzione SSML. Queste entità sono riservate dalla specifica della lingua. Questi personaggi sono i seguenti:

Carattere	Nome di escape
&	&
"	virgolette (virgolette doppie)

Carattere	Nome	Codice di escape
&	e commerciale	&
'	apostrofo o virgolette singole	'
<	segno minore di	<
>	segno maggiore di	>

Poiché SSML utilizza questi caratteri come parte del suo codice, per usare questi simboli in SSML, devi inserire un carattere di escape. Puoi utilizzare il codice di escape anziché il carattere effettivo in modo che venga visualizzato correttamente durante la creazione di un documento SSML valido. Ad esempio, la frase seguente

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

verrà sottoposta al rendering in SSML come

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

In questo caso, i caratteri speciali per l'apostrofo e la e commerciale sono preceduti da un carattere di escape, in modo che il documento SSML rimanga valido.

Per i simboli &, < e >, i codici di escape sono sempre necessari quando utilizzi SSML. Inoltre, anche quando usi l'apostrofo o la virgoletta singola (') come apostrofo, è necessario utilizzare il codice di escape.

Tuttavia, quando si utilizzano le virgolette doppie (") o le apostrophe/single virgolette (') come virgolette, l'uso o meno del codice di escape dipende dal contesto.

Virgolette doppie

- Devono essere precedute da un carattere di escape quando si trovano in un valore di attributo delimitato da virgolette doppie. Ad esempio, nel codice seguente AWS CLI

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

- Non è necessario specificare il carattere di escape nel contesto testuale, come nell'esempio seguente

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

- Non è necessario specificare il carattere di escape quando sono in un valore di attributo delimitato da virgolette singole. Ad esempio, nel seguente codice AWS CLI

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

Virgolette singole

- Devono essere precedute da un carattere di escape quando vengono utilizzate come apostrofo, come nell'esempio seguente

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

- Non è necessario specificare il carattere di escape nel contesto testuale, come nell'esempio seguente

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

- Non è necessario inserire il carattere di escape quando si trovano in un attributo di codice delimitato da virgolette doppie. Ad esempio, nel seguente codice AWS CLI

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

Utilizzo di SSML sulla console

Nell'esempio seguente, si utilizza un tag SSML per indicare a Amazon Polly di sostituire "World Wide Web Consortium" con "W3C" in un paragrafo breve. È anche possibile utilizzare i tag per introdurre una pausa e sussurrare una parola. Confrontare i risultati di questo esercizio con quelli di Applicazione dei lessici (sintesi vocale).

Per ulteriori informazioni su SSML e sui relativi esempi, consultare Tag SSML supportati.

Per eseguire la sintesi vocale da testo ottimizzato per SSML (console)

1. Accedi a AWS Management Console e apri la console Amazon Polly all'indirizzo. https://console.aws.amazon.com/polly/
2. Se non è già visualizzata, selezionare la scheda Text-to-Speech (Sintesi vocale).
3. Attiva SSML.
4. Digita o incolla questo testo nella casella di testo:

   ```
   <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>
   ```

   I tag SSML istruiscono Amazon Polly su come eseguire il rendering del testo:

   - `<break time="1s"/>` istruisce Amazon Polly di inserire una pausa di un secondo tra le prime due frasi.
   - `<sub alias="World Wide Web Consortium">W3C</sub>` istruisce Amazon Polly; di sostituire World Wide Web Consortium con l'acronimo W3C.
   - `<amazon:effect name="whispered">Nice job</amazon:effect>` indica a Amazon Polly di sussurrare la seconda istanza di "Nice job".

> **Note**
>
> Quando usi il AWS CLI, racchiudi il testo di input tra virgolette per differenziarlo dal codice circostante. La console di Amazon Polly non mostra codice, quindi non si racchiude il testo di input tra virgolette quando viene utilizzata.

5. Per Language (Lingua), scegli English, US (Inglese, Stati Uniti), quindi scegli una voce.
6. Per ascoltare la sintesi vocale, scegli Listen (Ascolta).
7. Per salvare il file locale, scegli Download. Se si desidera salvarlo in un formato differente, espandi Additional settings (Impostazioni aggiuntive) attiva Speech file format settings (Impostazioni del formato di file vocale) e scegli il formato desiderato, quindi scegli Download.

Utilizzo di SSML con il comando Synthesize-Speech

Questo esempio illustra come utilizzare il comando `synthesize-speech` con una stringa SSML. Quando si utilizza il comando `synthesize-speech`, in genere si fornisce quanto segue:

- Testo di input (obbligatorio)
- Tag di apertura e chiusura (obbligatori)
- Formato di output
- Voce

In questo esempio, specificare una semplice stringa di testo tra virgolette con i necessari tag di apertura e chiusura `<speak></speak>`.

> **Important**
>
> Anche se non utilizzi virgolette nel testo di input nella console di Amazon Polly, utilizzale con AWS CLI . È anche importante distinguere le virgolette nel testo di input e le virgolette necessarie per tag singoli.
> Ad esempio, è possibile utilizzare le virgolette standard (") per racchiudere l'input di testo e le virgolette singole (') per i tag interni o viceversa. Entrambe le opzioni funzionano per Unix, Linux e macOS. Tuttavia, con Windows racchiudere il testo di input tra virgolette standard e utilizzare virgolette singole per i tag.

Per tutti i sistemi operativi, è possibile utilizzare le virgolette standard (") per racchiudere l'input di testo e le virgolette singole (') per i tag interni o viceversa. Ad esempio:

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

Per Unix, Linux e macOS, è possibile anche fare l'inverso, ovvero utilizzare le virgolette singole (') per racchiudere l'input di testo e le virgolette standard (") per i tag interni:

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

L' AWS CLI esempio seguente è formattato per Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione Unix barra rovesciata (\) al termine di ogni riga con un accento circonflesso (^) e usa virgolette (") attorno al testo di input con virgolette singole (') per i tag interni.

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

Per ascoltare la sintesi vocale, riprodurre il file risultante `speech.mp3` utilizzando qualsiasi lettore audio.

Sintetizzazione di un documento ottimizzato con SSML

Per più testo di input, è consigliabile salvare i tuoi contenuti SSML in un file e semplicemente specificare il nome di file nel comando `synthesize-speech`. Ad esempio è possibile salvare quanto segue in un file denominato `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>
```

L'attributo `xml:lang` specifica `en-US` (Inglese - Stati Uniti) come lingua del testo di input. Per informazioni su come la lingua del testo di input e la lingua della voce selezionata influiscono sul funzionamento di `SynthesizeSpeech`, consultare Specificare un'altra lingua per parole specifiche.

Per eseguire un file ottimizzato per SSML

1. Salvare SSML in un file, (ad esempio `example.xml`).
2. Eseguire il comando `synthesize-speech` dal percorso dove il file XML è memorizzato e specificare il file SSML come input sostituendo il testo di input con `file:\\example.xml`. Poiché questo comando punta a un file anziché contenere il testo di input effettivo, non utilizzare le virgolette.

   > **Note**
   >
   > L' AWS CLI esempio seguente è formattato per Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione UNIX barra rovesciata (\) al termine di ogni riga con un accento circonflesso (^).

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

3. Per ascoltare la sintesi vocale, riprodurre il file risultante `speech.mp3` utilizzando qualsiasi lettore audio.

Tag SSML supportati

Tutti i tag tranne i `<amazon:domain name="news">` sono supportati per le voci standard. La disponibilità dei tag per altre voci è riportata nella tabella seguente.

Amazon Polly supporta i seguenti tag SSML:

Azione	Tag SSML	Disponibilità della voce neurale	Disponibilità vocale a lungo termine	Disponibilità vocale generativa
Aggiungere una pausa	<break>	Disponibilità completa	Disponibilità completa	Disponibilità completa
Enfatizzare le parole	<emphasis>	Non disponibile	Non disponibile	Non disponibile
Specificare un'altra lingua per parole specifiche	<lang>	Disponibilità completa	Disponibilità completa	Disponibilità completa
Inserimento di un tag personalizzato nel testo	<mark>	Disponibilità completa	Disponibilità completa	Disponibilità parziale
Aggiungere una pausa tra i paragrafi	<p>	Disponibilità completa	Disponibilità completa	Disponibilità completa
Uso della pronuncia fonetica	<phoneme>	Disponibilità completa	Disponibilità completa	Disponibilità parziale
Controllo del volume, della velocità di conversazione e dell'intonazione	<prosody>	Disponibilità parziale	Disponibilità parziale	Disponibilità parziale
Impostazione di una durata massima per il parlato sintetizzato	<prosody amazon:max-duration>	Non disponibile	Non disponibile	Non disponibile

Azione	Tag SSML	Disponibilità della voce neurale	Disponibilità vocale a lungo termine	Disponibilità vocale generativa
Aggiungere una pausa tra le frasi	<s>	Disponibilità completa	Disponibilità completa	Disponibilità completa
Controllo del modo in cui vengono pronunciati tipi speciali di parole	<say-as>	Disponibilità parziale	Disponibilità completa	Disponibilità completa
Identificazione del testo migliorato con SSML	<speak>	Disponibilità completa	Disponibilità completa	Disponibilità completa
Pronunciare acronimi e abbreviazioni	<sub>	Disponibilità completa	Disponibilità completa	Disponibilità completa
Migliorare la pronuncia specificando parti del discorso	<w>	Disponibilità completa	Disponibilità completa	Disponibilità completa
Aggiungere il suono del respiro	<amazon:auto-breaths>	Non disponibile	Non disponibile	Non disponibile
Stile di pronuncia newscaster	<amazon:domain name="news">	Seleziona solo voci neurali	Non disponibile	Non disponibile
Aggiungere la compressione della gamma dinamica	<amazon:effect name="drc">	Disponibilità completa	Disponibilità completa	Non disponibile
Parlando a bassa voce	<amazon:effect phonation="soft">	Non disponibile	Non disponibile	Non disponibile

Azione	Tag SSML	Disponibilità della voce neurale	Disponibilità vocale a lungo termine	Disponibilità vocale generativa
Controllo del timbro	<amazon:effect > vocal-tract-length	Non disponibile	Non disponibile	Non disponibile
Sussurrare	<amazon:effect name="whispered">	Non disponibile	Non disponibile	Non disponibile

> **Note**
>
> Se utilizzi tag SSML non supportati in formato standard, neurale o lungo, riceverai un errore.

Identificazione del testo migliorato con SSML

<speak>

Questo tag è supportato dai formati TTS generativi, estesi, neurali e standard.

Il tag `<speak>` è l'elemento principale di tutto il testo SSML di Amazon Polly. Tutto il testo ottimizzato per SSML deve essere racchiuso in una coppia di tag <speak>.

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

Aggiungere una pausa

<break>

Questo tag è supportato dai formati TTS generativi, estesi, neurali e standard.

Per aggiungere una pausa al testo, utilizza il tag <break>. È possibile impostare una pausa in base all'intensità (equivalente alla pausa dopo una virgola, una frase o un paragrafo), oppure in base a un determinato intervallo di tempo in secondi o millisecondi. Se non si specifica un attributo per stabilire la durata della pausa, Amazon Polly utilizza l'impostazione predefinita `<break strength="medium"/>`, che aggiunge una pausa della lunghezza di una pausa dopo la virgola.

`strength` valori attributo:

- `none`: nessuna pausa. Utilizza `none` per rimuovere una pausa che si verifica normalmente, ad esempio dopo un punto.
- `x-weak`: ha la stessa forza di `none`, senza pausa.
- `weak`: imposta una pausa della stessa durata della pausa dopo la virgola.
- `medium`: ha la stessa forza di `weak`.
- `strong`: imposta una pausa della stessa durata della pausa dopo una frase.
- `x-strong`: imposta una pausa della stessa durata della pausa dopo un paragrafo.

`time` valori attributo:

- *`[number]`*`s`: la durata della pausa, in secondi. La durata massima è `10s`.
- *`[number]`*`ms`: la durata della pausa, in millisecondi. La durata massima è `10000ms`.

Ad esempio:

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

Se con il tag `break` non viene utilizzato nessun attributo, il risultato varia a seconda del testo:

- Se non c'è altra punteggiatura accanto al tag `break`, verrà creata una `<break strength="medium"/>` (comma-length pause).
- Se il tag è accanto a una virgola, il tag verrà aggiornato a una `<break strength="strong"/>` (sentence-length pause).
- Se il tag è accanto a un punto, il tag verrà aggiornato a una `<break strength="x-strong"/>` (paragraph-length pause).

Enfatizzare le parole

<emphasis>

Questo tag è supportato solo dal formato TTS standard.

Per enfatizzare le parole, utilizza il tag <emphasis>. L'enfatizzazione delle parole modifica la velocità e il volume della sintesi vocale. Una maggiore enfasi significa che Amazon Polly pronuncia il testo a voce più alta e più lentamente. Con una minore enfasi, viene pronunciato più velocemente e con tono più pacato. Per specificare il grado di enfasi, utilizzare l'attributo `level`.

`level` valori attributo:

- `Strong`: aumenta il volume e rallenta la velocità di pronuncia in modo che la sintesi vocale sia più lenta e con un tono più alto.
- `Moderate`: aumenta il volume e rallenta la velocità di pronuncia, ma meno rispetto a `strong`. `Moderate` è il valore predefinito.
- `Reduced`: diminuisce il volume e accelera la velocità di pronuncia. La sintesi vocale viene pronunciata più velocemente e con tono più morbido.

> **Note**
>
> Il volume e la velocità di pronuncia normali per una voce rientrano tra i livelli `moderate` e `reduced`.

Ad esempio:

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

Specificare un'altra lingua per parole specifiche

<lang>

Questo tag è supportato dai formati TTS generativi, estesi, neurali e standard.

Specifica un'altra lingua per una determinata parola o frase con il tag <lang>. Le parole e frasi di lingue straniere generalmente sono pronunciate in modo più chiaro quando vengono incluse all'interno di una coppia di tag `<lang>`. Per specificare la lingua, utilizzare l'attributo `xml:lang`. Per un elenco completo delle lingue disponibili, consultare Lingue in Amazon Polly.

Se non si applica il tag `<lang>`, tutte le parole del testo di input vengono pronunciate nella lingua della voce specificata in `voice-id`. Se si applica il tag `<lang>`, le parole sono pronunciate in tale lingua.

Ad esempio, se il `voice-id` è Joanna (che parla inglese, Stati Uniti), Amazon Polly pronuncia quanto segue con la voce di Joanna senza un accento francese:

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

Se si utilizza la voce Joanna con il tag `<lang>`, Amazon Polly pronuncia la frase nella voce di Joanna con un francese con un accento americano:

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

Poiché Joanna non è madrelingua francese, la pronuncia si basa sulla sua lingua nativa, ovvero l'inglese degli Stati Uniti. Ad esempio, sebbene una perfetta pronuncia francese presenti una /R/ vibrata alveolare nella parola français, la voce di Joanna in inglese americano pronuncia questo fonema con il suono corrispondente /r/.

Se utilizzi il `voice-id` di Giorgio, che parla italiano, con il seguente testo, Amazon Polly pronuncia la frase con la voce di Giorgio e la pronuncia in italiano:

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

Se si utilizza la stessa voce con il seguente tag `<lang>`, Amazon Polly pronuncia Bruce Springsteen in inglese con l'accento italiano:

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

Questo tag può essere utilizzato anche come sostituto dell'opzione opzionale DefaultLangCodeper la sintesi vocale. Tuttavia, questa operazione richiede la formattazione del testo con SSML.

Inserimento di un tag personalizzato nel testo

<mark>

Questo tag è supportato dai formati TTS estesi, neurali e standard. Questo tag non fa nulla per le voci generative perché i segni vocali non sono disponibili per le voci generative.

Per inserire un tag personalizzato nel testo, utilizza il tag <mark>. Amazon Polly non esegue alcuna operazione sul tag, ma restituisce la posizione del tag nei metadati di SSML. Questo tag può essere qualsiasi operazione per la quale si desidera effettuare la chiamata, purché mantenga il formato seguente:

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

Ad esempio, se il nome del tag è "animal" e il testo di input è:

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

È possibile che Amazon Polly restituisca i seguenti metadati da SSML:

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

Aggiungere una pausa tra i paragrafi

<p>

Questo tag è supportato dai formati TTS generativi, estesi, neurali e standard.

Per aggiungere una pausa tra paragrafi nel tuo testo, utilizza il tag <p>. L'utilizzo di questo tag fornisce una pausa più lunga rispetto alle pause dei madrelingua dopo le virgole o la fine di una frase. Utilizza il tag <p> per racchiudere il paragrafo:

```
<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>
```

Ciò equivale a specificare una pausa utilizzando <break strength="x-strong"/>.

Uso della pronuncia fonetica

<phoneme>

Questo tag è supportato dai formati TTS in formato esteso, neurali e standard.

Per fare in mode che Amazon Polly utilizzi la pronuncia fonetica per testo specifico, utilizzare il tag <phoneme>.

Con il tag `<phoneme>` sono necessari due attributi. Indicano l'alfabeto fonetico utilizzato da Amazon Polly e i simboli fonetici della pronuncia corretta:

- `alphabet`
  - `ipa`: indica che verrà utilizzato l'alfabeto fonetico internazionale (IPA).
  - `x-sampa`: indica che verrà utilizzato l'alfabeto X-SAMPA (Extended Speech Assessment Methods Fonetico Alphabet).
- `ph`
  - Specifica i simboli fonetici per la pronuncia. Per ulteriori informazioni, consulta Lingue in Amazon Polly

Con il tag `<phoneme>`, Amazon Polly utilizza la pronuncia specificata dall'attributo `ph` invece della pronuncia standard associata come impostazione predefinita al linguaggio utilizzato dalla voce selezionata.

Ad esempio, la parola "pecan" può essere pronunciata in due modi. Nell'esempio che segue, alla parola "pecan" viene assegnata una pronuncia personalizzata diversa in ciascuna riga. Amazon Polly pronuncia pecan come specificato negli attributi `ph`, anziché utilizzare la pronuncia predefinita:

Alfabeto fonetico internazionale (IPA)

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

Extended Speech Assessment Methods Phonetic Alphabet (X-SAMPA)

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

Il cinese mandarino utilizza Pinyin per la pronuncia fonetica.

Pinyin

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

Il giapponese usa Yomigana e Pronuncia 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>
```

Pronuncia Kana

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

Controllo del volume, della velocità di conversazione e dell'intonazione

<prosody>

Gli attributi dei tag Prosody sono completamente supportati dalle voci TTS standard. Le voci generative, neurali e in forma lunga supportano gli `rate` attributi `volume` and, ma non supportano l'attributo. `pitch` Per le voci generative, il tag prosody può essere usato solo intorno a frasi complete.

Per controllare il volume, la velocità o l'intonazione della voce selezionata, utilizza il tag `prosody`.

Il volume, la velocità della sintesi vocale e l'intonazione dipendono dalla voce specifica selezionata. Oltre alle differenze tra voci per diversi linguaggi, ci sono differenze tra le singole voci che parlano la stessa lingua. Per questo motivo, mentre gli attributi sono simili in tutti i linguaggi, ci sono chiare varianti da lingua a lingua e nessun valore assoluto è disponibile.

Il tag `prosody` ha tre attributi, ognuno dei quali dispone di diversi valori disponibili per impostare l'attributo. Ogni attributo utilizza la stessa sintassi:

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

- `volume`
  - `default`: reimposta il volume sul livello di default per la voce corrente.
  - `silent`, `x-soft`, `soft`, `medium`, `loud`, `x-loud`: consente di specificare il volume su un valore di default per la voce corrente.
  - `+ndB`, `-ndB`: modifica il volume in relazione al livello corrente. Il valore `+0dB` indica che il volume rimane invariato, mentre `+6dB` indica circa il doppio dell'attuale volume e `-6dB` indica circa la metà dell'attuale volume.

  Ad esempio, è possibile impostare il volume per un passaggio come segue:

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

  In alternativa, puoi impostarlo in questo modo:

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

- `rate`
  - `x-slow`, `slow`, `medium`, `fast`,`x-fast`. Impostare l'intonazione su un valore predefinito per la voce selezionata.
  - `n%`: Una modifica della percentuale non negativa nella velocità di pronuncia. Ad esempio, un valore di 100% significa nessun cambiamento alla velocità di pronuncia, un valore pari a 200%

significa una velocità di pronuncia il doppio della velocità predefinita e un valore del 50% significa una velocità di pronuncia la metà della velocità predefinita. Questo valore ha un intervallo di 20-200%.

Ad esempio, è possibile impostare la velocità della sintesi vocale per un passaggio come segue:

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

In alternativa, puoi impostarlo in questo modo:

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

- `pitch`
  - `default`: reimposta l'intonazione sul livello di default per la voce corrente.
  - `x-low`, `low`, `medium`, `high`, `x-high`: imposta l'intonazione su un valore predefinito per la voce corrente.
  - `+n%` o `-n%`: regola l'intonazione in base a una percentuale relativa. Ad esempio, un valore di `+0%` significa che non c'è alcun cambiamento nell'intonazione di base, `+5%` offre un'intonazione di base leggermente più elevata, mentre `-5%` avrà come risultato un'intonazione di base leggermente inferiore.

Ad esempio, è possibile impostare l'intonazione per un passaggio come segue:

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

In alternativa, puoi impostarlo in questo modo:

```
<speak>
```

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

Il tag **<**prosody **>** deve contenere almeno un attributo, ma può includerne di più all'interno dello stesso tag.

```
<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>
```

Può anche essere combinato con tag nidificati, come segue:

```
<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>
```

Impostazione di una durata massima per il parlato sintetizzato

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

Questo tag è attualmente supportato solo dal formato TTS standard.

Per controllare la durata di un discorso quando viene sintetizzato, utilizza il tag `<prosody>` con l'attributo `amazon:max-duration`.

La durata della sintesi vocale varia leggermente a seconda della voce selezionata. Potrebbe quindi essere difficile associare la sintesi vocale a grafica o altre attività che richiedono tempi precisi. Questo problema peggiora con le applicazioni di traduzione, perché il tempo necessario per pronunciare frasi specifiche può variare ampiamente con lingue diverse.

Il tag `<prosody amazon:max-duration>` associa la sintesi vocale alla quantità di tempo che desidera richiedere (la durata).

Questo tag utilizza la sintassi seguente:

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

Con il tag `<prosody amazon:max-duration>`, puoi specificare la durata in secondi o millisecondi:

- *n*`s`: durata massima in secondi
- *n*`ms`: durata massima in millisecondi

Ad esempio, il seguente testo parlato ha una durata massima di 2 secondi:

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

Il testo posizionato all'interno del tag non supera la durata specificata. Se la voce o la lingua scelta richiederebbe normalmente più tempo di tale durata, Amazon Polly accelera la sintesi vocale in modo che rientri nella durata specificata.

Se la durata specificata è superiore a quella richiesta per leggere il testo a una velocità normale, Amazon Polly legge la sintesi vocale normalmente. Non rallenta la sintesi vocale, né aggiunge silenzio, perciò l'audio risultante è più breve di quanto richiesto.

> **Note**
>
> Amazon Polly aumenta la velocità non più di 5 volte rispetto alla velocità normale. Se il testo viene letto più velocemente, non è in genere comprensibile. Se una sintesi vocale non rientra nella durata specificata anche quando accelerata al massimo, l'audio sarà accelerato ma durerà più della durata specificata.

Puoi includere una sola frase o più frasi all'interno di un tag `<prosody amazon:max-duration>` e puoi utilizzare più tag `<prosody amazon:max-duration>` all'interno del testo.

Ad esempio:

```
<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>
```

L'utilizzo del tag `<prosody amazon:max-duration>` è in grado di aumentare la latenza quando Amazon Polly restituisce una sintesi vocale. Il grado di latenza dipende dal passaggio e dalla relativa lunghezza. È consigliabile utilizzare testo costituito da passaggi di testo relativamente brevi.

Limitazioni

Non vi sono limitazioni nella modalità di utilizzo del tag `<prosody amazon:max-duration>` né su come funziona con altri tag SSML:

- Il testo all'interno di un tag `<prosody amazon:max-duration>` non può superare il limite di 1.500 caratteri.
- Non è possibile nidificare i tag `<prosody amazon:max-duration>`. Se hai inserito un tag `<prosody amazon:max-duration>` all'interno di un altro, Amazon Polly ignora il tag interno.

  Ad esempio, nell'istruzione seguente, il tag `<prosody amazon:max-duration="5s">` viene ignorato:

```
<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>
```

- Non è possibile utilizzare i tag `<prosody>` con l'attributo `rate` all'interno di un tag `<prosody amazon:max-duration>`. Questo perché entrambi hanno impatto sulla velocità a cui il testo viene pronunciato.

  Nell'esempio seguente, Amazon Polly ignora il 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>
```

Pause e **`max-duration`**

Quando utilizzi il tag `max-duration`, puoi comunque inserire pause all'interno del testo. Tuttavia, Amazon Polly include la lunghezza della pausa quando si calcola la durata massima per sintesi vocale. Inoltre, Amazon Polly conserva le brevi pause che si verificano laddove virgole e punti vengono posizionati all'interno di un passaggio e le include nella durata massima.

Ad esempio, nel blocco seguente, l'interruzione di 600 millisecondi e le interruzioni causate dalle virgole e i punti si verificano all'interno della sintesi vocale da 8 secondi:

```
<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>
```

Aggiungere una pausa tra le frasi

<s>

Questo tag è supportato dai formati TTS generativi, estesi, neurali e standard.

Per aggiungere una pausa tra righe o frasi nel tuo testo, utilizzare il tag `<s>`. L'utilizzo del tag ha lo stesso effetto di:

- Terminare una frase con un punto (.)
- Specificare una pausa con `<break strength="strong"/>`

A differenza del tag `<break>`, il tag **<s >** racchiude la frase. Questa funzione è utile per la sintesi vocale che è organizzata in righe, invece di frasi, ad esempio le poesie.

Nell'esempio seguente, il tag `<s>` crea una breve pausa dopo la prima e la seconda frase. La frase finale non presenta tag `<s>`, ma è anche seguita da una breve pausa perché finisce con un punto.

```
<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>
```

Controllo del modo in cui vengono pronunciati tipi speciali di parole

<say-as>

Il `<say-as>` tag è supportato da motori TTS generativi, di lunga durata, neurali e standard. Tieni presente, tuttavia, che se Amazon Polly utilizza una voce neurale e incontra il `<say-as>` tag con l'`characters`opzione in fase di esecuzione, la frase interessata verrà sintetizzata utilizzando la relativa voce standard. Tuttavia, la frase interessata verrà comunque fatturata come se utilizzasse una voce neurale.

Utilizza il tag `<say-as>` con l'attributo `interpret-as` per indicare a Amazon Polly come pronunciare determinati caratteri, parole e numeri. In tal modo è possibile fornire ulteriore contesto

per eliminare qualsiasi ambiguità in merito a come deve essere effettuato il rendering del testo da parte di Amazon Polly.

Il `<say-as>` tag utilizza un attributo`interpret-as`, che utilizza una serie di possibili valori disponibili. Ogni attributo utilizza la stessa sintassi:

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

I valori seguenti sono disponibili con `interpret-as`:

- `characters`oppure`spell-out`: compila ogni lettera del testo, ad a-b-c esempio.

> **Note**
>
> Questa opzione non è attualmente supportata per le voci neurali. Se utilizzi una voce neurale e questo codice SSML viene rilevato da Amazon Polly in fase di esecuzione, la frase interessata verrà sintetizzata utilizzando la relativa voce standard. Tieni presente, tuttavia, che questa frase verrà comunque fatturata come se utilizzasse una voce neurale.

- `cardinal` o `number`: interpreta il testo numerico come un numero cardinale, come in 1.234.
- `ordinal`: interpreta il testo numerico come numero ordinale, come 1.234°.
- `digits`: pronuncia ogni cifra individualmente, come in 1-2-3-4.
- `fraction`: interpreta il testo numerico come frazione. Questo è valido sia per frazioni comuni, ad esempio 3/20, sia per frazioni miste, ad esempio 2 ½. Guardare qui di seguito per ulteriori informazioni.
- `unit`: interpreta un testo numerico come una misura. Il valore deve essere un numero o una frazione seguita da un'unità senza spazi tra i due elementi come in `1/2inch` oppure da un'unità come in `1meter`.
- `date`: interpreta il testo come data. Il formato della data deve essere specificato con l'attributo formato. Guardare qui di seguito per ulteriori informazioni.
- `time`: interpreta il testo numerico come durata espressa in minuti e secondi, come in `1'21"`.
- `address`: interpreta il testo come parte di un indirizzo.
- `expletive`: sostituisce con un segnale acustico il contenuto incluso nel tag.
- `telephone`: interpreta il testo numerico come un numero di telefono di 7 cifre o 10 cifre, ad esempio `2025551212`. È anche possibile utilizzare questo valore per gestire le estensioni telefoniche, ad esempio `2025551212x345`. Guardare qui di seguito per ulteriori informazioni.

> **Note**
>
> Attualmente l'opzione `telephone` non è disponibile per tutte le lingue. Tuttavia, è disponibile per le voci che parlano varianti della lingua inglese (en-AU, en-GB, en-IN, en-US e en-GB-WLS), varianti dello spagnolo (es-ES, es-MX e es-US), varianti della lingua francese (fr-FR e fr-CA) e varianti del portoghese (pt-BR e pt-PT), oltre al tedesco (de-DE), italiano (it-IT), giapponese (ja-JP) e russo (ru-IT). Va inoltre notato che in alcuni casi, lingue come l'arabo (arb) gestiscono automaticamente il numero impostato come numero di telefono e quindi non implementano effettivamente il tag `telephone` SSML.

Frazioni

Amazon Polly interpreta i valori nel tag `say-as` che presentano l'attributo `interpret-as="fraction"` come comuni frazioni. Di seguito è riportata la sintassi per le frazioni:

- Frazione

  Sintassi:*`cardinal number`*/*`cardinal number`*, ad esempio 2/9.

  Ad esempio: `<say-as interpret-as="fraction">2/9</say-as>` si pronuncia "due noni."
- Numero misto non negativo

  Sintassi: *`cardinal number`* +*`cardinal number`*/*`cardinal number`*, ad esempio 3+1/2.

  Ad esempio: `<say-as interpret-as="fraction">3+1/2</say-as>` si pronuncia "tre e mezzo".

> **Note**
>
> Deve esserci un + tra "3" e "1/2". Amazon Polly non supporta un numero misto senza +, ad esempio "3 1/2".

Date:

Quando `interpret-as` è impostato su `date`, è necessario indicare il formato della data.

Questo utilizza la sintassi seguente:

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

Ad esempio:

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

I formati seguenti possono essere utilizzati con l'attributo `date`.

- `mdy`: Month-day-year.
- `dmy`: Day-month-year.
- `ymd`: Year-month-day.
- `md`: mese-giorno.
- `dm`: giorno-mese.
- `ym`: anno-mese.
- `my`: mese-anno.
- `d`: giorno.
- `m`: mese.
- `y`: anno.
- `yyyymmdd` Year-month-day:. Se utilizzi questo formato, puoi fare in modo che Amazon Polly salti parti della data utilizzando punti interrogativi.

  Ad esempio, Amazon Polly riproduce quanto segue come "22 settembre":

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

  `Format` non è necessario.

Telefono

Amazon Polly tenta di interpretare il testo fornito correttamente in base alla formattazione del testo anche senza il tag `<say-as>`. Ad esempio, se il testo include "202-555-1212", Amazon Polly lo interpreta come un numero di telefono a 10 cifre e pronuncia ogni singola cifra individualmente, con una breve pausa per ogni trattino. In questo caso, non utilizzare `<say-as interpret-`

`as="telephone">`. Tuttavia, se si desidera che il testo "2025551212" venga pronunciato da Amazon Polly come un numero di telefono, utilizzare `<say-as interpret-as="telephone">`.

La logica per l'interpretazione di ciascun elemento è specifica per la lingua. Ad esempio, i numeri di telefono sono pronunciati in modo diverso nell'inglese americano e nell'inglese britannico (nell'inglese britannico, vengono raggruppate sequenze della stessa cifra, ad esempio "doppio cinque" o "triplo quattro"). Per mostrare la differenza, è possibile testare il seguente esempio scegliendo la voce in inglese americano e in inglese britannico:

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

Pronunciare acronimi e abbreviazioni

<sub>

Questo tag è supportato dai formati TTS generativi, estesi, neurali e standard.

Utilizza il tag `<sub>` con l'attributo `alias` per sostituire una parola diversa (o pronuncia) per il testo selezionato, ad esempio un acronimo o un'abbreviazione.

Questo utilizza la sintassi:

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

In questo esempio, il simbolo chimico dell'elemento viene sostituito con il nome "Mercury" per rendere più chiari i contenuti audio.

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

Migliorare la pronuncia specificando parti del discorso

<w>

Questo tag è supportato dai formati TTS generativi, estesi, neurali e standard.

Puoi utilizzare il tag **<w>** per personalizzare la pronuncia delle parole specificando la parte del discorso della parola o un altro significato. Questa operazione viene eseguita utilizzando l'attributo `role`.

Questo tag utilizza la sintassi seguente:

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

È possibile utilizzare i seguenti valori per l'attributo `role`:

Per specificare la parte del discorso:

- `amazon:VB`: interpreta la parola come un verbo (presente semplice).
- `amazon:VBD`: interpreta la parola come un verbo passato.
- `amazon:DT`: interpreta la parola come un determinante.
- `amazon:IN`: interpreta la parola come una preposizione.
- `amazon:JJ`: interpreta la parola come un aggettivo.
- `amazon:NN`: interpreta la parola come un sostantivo.

Ad esempio, a seconda della parte del discorso, la pronuncia inglese americana della parola "read" varia in base al tag:

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

Per specificare un significato specifico:

- `amazon:DEFAULT`: utilizza il significato predefinito della parola.
- `amazon:SENSE_1`: utilizza il significato non di default della parola, laddove presente. Ad esempio, il sostantivo "bass" è pronunciato in modi differenti a seconda del suo significato. Il significato predefinito è la parte inferiore della scala musicale. L'altro significato è una specie di pesce d'acqua dolce, detto anche "bass" ma pronunciato in modo diverso. Utilizzando `<w role="amazon:SENSE_1">bass</w>` si rende la pronuncia non di default (pesce d'acqua dolce) per il testo dell'audio.

Questa differenza di pronuncia e di significato può essere percepita sintetizzando quanto segue:

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

> **Note**
>
> Alcune lingue possono avere una selezione diversa delle parti del discorso supportate.

Aggiungere il suono del respiro

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

Questo tag è supportato solo dal formato TTS standard.

Il linguaggio naturale include sia parole correttamente pronunciate sia suoni respiratori. Aggiungendo suoni respiratori alla sintesi vocale, è possibile rendere il suono più naturale. I tag `<amazon:breath>` e `<amazon:auto-breaths>` forniscono le respirazioni. Sono disponibili le seguenti opzioni:

- Modalità manuale: puoi impostare la posizione, la durata e il volume di una respirazione all'interno del testo
- Modalità automatica: Amazon Polly inserisce automaticamente i suoni della respirazione nell'output del discorso
- Modalità mista: sia tu sia Amazon Polly aggiungete i suoni della respirazione

Modalità manuale

Nella modalità manuale, inserisci il tag `<amazon:breath/>` nel testo di input in cui desideri posizionare un respiro. Puoi personalizzare la durata e il volume delle respirazioni con, rispettivamente, gli attributi `duration` e `volume`:

- `duration`: controlla la durata del respiro. I valori validi sono: `default`, `x-short`, `short`, `medium`, `long`, `x-long`. Il valore predefinito è `medium`.
- `volume`: controlla il volume del respiro. I valori validi sono: `default`, `x-soft`, `soft`, `medium`, `loud`, `x-loud`. Il valore predefinito è `medium`.

> **Note**
>
> La lunghezza esatta e il volume di ogni valore dell'attributo dipendono dalla voce di Amazon Polly specifica utilizzata.

Per impostare il suono di una respirazione utilizzando le impostazioni predefinite, usa `<amazon:breath/>` senza attributi.

Ad esempio, per utilizzare gli attributi per impostare la durata e il volume di un respiro su un valore medio, devi configurare gli attributi come segue:

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

Per usare le impostazioni predefinite, è sufficiente utilizzare il tag:

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

Puoi aggiungere singoli suoni respiratori all'interno di un passaggio, come segue:

```
<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>
```

Modalità automatica

Nella modalità automatica, utilizzi il tag `<amazon:auto-breaths>` per dire a Amazon Polly di creare automaticamente suoni respiratori a intervalli stabiliti. Puoi impostare la frequenza degli intervalli, il volume e la durata. Posiziona il tag `</amazon:auto-breaths>` all'inizio del testo a cui vuoi applicare la respirazione automatica e chiudi il tag al termine.

> **Note**
>
> A differenza del tag `<amazon:breath/>` della modalità manuale, il tag `<amazon:auto-breaths>` richiede un tag di chiusura (`</amazon:auto-breaths>`).

È possibile utilizzare i seguenti attributi opzionali con il tag `<amazon:auto-breaths>`:

- `volume`: controlla il volume del respiro. I valori validi sono: `default`, `x-soft`, `soft`, `medium`, `loud`, `x-loud`. Il valore predefinito è `medium`.
- `frequency`: controlla la frequenza con cui il respiro è presente nel testo. I valori validi sono: `default`, `x-low`, `low`, `medium`, `high`, `x-high`. Il valore predefinito è `medium`.
- `duration`: controlla la durata del respiro. I valori validi sono: `default`, `x-short`, `short`, `medium`, `long`, `x-long`. Il valore predefinito è `medium`.

Per impostazione predefinita, la frequenza dei suoni di respirazione dipende dal testo di input. Tuttavia, i suoni della respirazione spesso si verificano dopo virgole e punti.

Gli esempi seguenti mostrano come utilizzare il tag `<amazon:auto-breaths>`. Per scegliere le opzioni da utilizzare per i tuoi contenuti, copia gli esempi applicabili sulla console di Amazon Polly e ascolta le differenze.

- Utilizzo della modalità automatica senza parametri opzionali.

```
<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>
```

- Utilizzo della modalità automatica con il controllo del volume. I parametri non specificati (`duration` e `frequency`) sono impostati sui valori predefiniti (`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>
```

- Utilizzo della modalità automatica con il controllo della frequenza. I parametri non specificati (`duration` e `volume`) sono impostati sui valori predefiniti (`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>
```

- Utilizzo della modalità automatica con più parametri. Per il parametro `Duration` non specificato, Amazon Polly usa il valore predefinito (`medium`).

```
<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>
```

Stile di pronuncia newscaster

<amazon:domain name="news">

Lo stile newscaster è disponibile solo per le voci di Matthew o Joanna, disponibili solo in inglese Stati Uniti (en-US), Lupe, in inglese spagnolo (es-US) e Amy, in inglese Regno Unito (en-GB). È supportato solo quando si utilizza il formato `Neural`.

Per utilizzare lo stile newscaster, utilizza i tag SSML e la sintassi seguente:

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

Ad esempio, puoi utilizzare lo stile newscaster con la voce di Amy come segue:

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

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

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

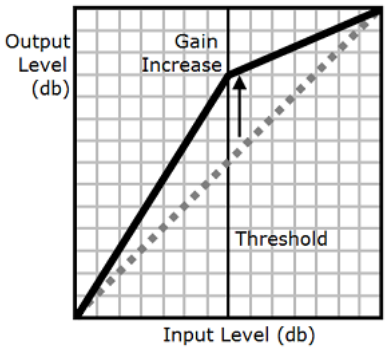
Aggiungere la compressione della gamma dinamica

<amazon:effect name="drc">

Questo tag è supportato dai formati TTS a lungo termine, neurali e standard.

A seconda del testo, della lingua e della voce utilizzati in un file audio, la gamma sonora varia da morbida a forte. I suoni ambientali, ad esempio il suono di un veicolo in movimento, possono spesso mascherare i suoni più tenui. Ciò impedisce di percepire chiaramente la traccia audio. Per aumentare il volume di alcuni suoni nel tuo file audio, utilizza il tag (`drc`) per la compressione dinamica intervalli.

Il tag `drc` imposta una soglia di "sonorità" media per l'audio e aumenta il volume (il guadagno) dei suoni attorno a tale soglia. Applica l'aumento massimo di guadagno nel punto più vicino alla soglia, mentre l'aumento di guadagno diminuisce quanto più lontano dalla soglia.



In questo modo, i suoni intermedi vengono uditi più facilmente in un ambiente rumoroso e ciò rende più chiaro l'intero file audio.

Il tag `drc` è un parametro Booleano (o è presente o non lo è). Utilizza la sintassi: `<amazon:effect name="drc">` e viene chiuso con `</amazon:effect>`.

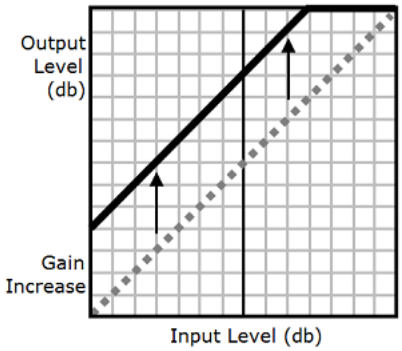
È possibile utilizzare il tag `drc` con qualsiasi voce o lingua supportate da Amazon Polly. È possibile applicarlo a un'intera sezione della registrazione o solo ad alcune parole. Ad esempio:

```
<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>
```

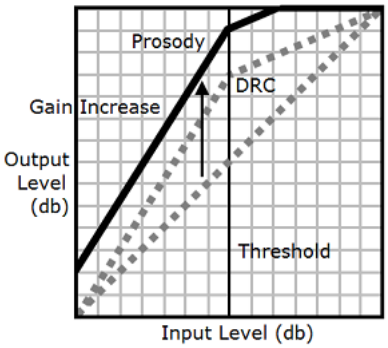
> **Note**
>
> Quando utilizzi"`drc`" nella sintassi , ricorda che fa distinzione tra maiuscole e minuscole.`amazon:effect`

Utilizzo di **`drc`** con il tag **`prosody volume`**

Come illustrato nel grafico che segue, il tag `prosody volume` aumenta in modo uniforme il volume di un intero file audio dal livello originale (linea punteggiata) a un livello adeguato (linea continua). Per aumentare ulteriormente il volume di determinate parti del file, utilizza il tag `drc` con il tag `prosody volume`. La combinazione dei tag non modifica le impostazioni del tag `prosody volume`.



Quando utilizzi i tag `drc` e `prosody volume` insieme, Amazon Polly applica prima il tag `drc`, aumentando i suoni intermedi (quelli vicino alla soglia). Quindi, applica il tag `prosody volume` e aumenta ulteriormente il volume dell'intera traccia audio in modo uniforme.



Per usare i tag in combinazione, occorre nidificarli uno all'interno dell'altro. Ad esempio:

```
<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>
```

In questo testo, il tag `prosody volume` aumenta il volume dell'intero passaggio a "loud" (alto). Il tag `drc` ottimizza il volume dei valori intermedi nella seconda frase.

> **Note**
>
> Quando utilizzi i tag `drc` e `prosody volume` insieme, segui le procedure XML standard per la nidificazione dei tag.

Parlando a bassa voce

<amazon:effect phonation="soft">

Questo tag è attualmente supportato solo dal formato TTS standard.

Per specificare che il testo di input deve essere pronunciato con una softer-than-normal voce, usa il <amazon:effect phonation="soft">tag.

Questo utilizza la sintassi:

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

Ad esempio, è possibile utilizzare questo tag con la voce di Matthew come segue:

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

Controllo del timbro

<amazon:effect > vocal-tract-length

Questo tag è attualmente supportato solo dal formato TTS standard.

Il timbro è la qualità tonale di una voce che consente di stabilire la differenza tra voci, anche quando hanno la stessa intonazione e sonorità. Una delle più importanti caratteristiche fisiologiche che contribuisce al timbro della sintesi vocale è la lunghezza del tratto vocale. Il tratto vocale è una cavità di aria che si estende dalla parte superiore delle corde vocali fino al bordo delle labbra.

Per controllare il timbro dell'output vocale in Amazon Polly, utilizza il tag `vocal-tract-length`. Questo tag ha l'effetto di modificare la lunghezza del tratto vocale dell'oratore, che suona come una modifica delle dimensioni dello stesso. Aumentando `vocal-tract-length`, l'oratore dà l'impressione di essere fisicamente più grande. Diminuendolo, l'oratore dà l'impressione di essere più piccolo. Puoi utilizzare questo tag con qualsiasi voce del portafoglio Amazon Polly Text-to-Speech.

Per modificare il timbro, utilizza i seguenti valori:

- `+n%` o `-n%`: regola la lunghezza del tratto vocale in base a una modifica percentuale relativa nella voce corrente. Ad esempio, +4% o -2%. I valori validi sono compresi tra +100% e -50%. I valori al di fuori di questo intervallo vengono tagliati. Ad esempio, +111% suona come+100% e -60% suona come -50%.
- `n%`: consente di modificare la lunghezza del tratto vocale impostando una percentuale assoluta per la lunghezza del tratto della voce corrente. Ad esempio, 110% o 75%. Un valore assoluto di 110% equivale a un valore relativo di+10%. Un valore assoluto di 100% è lo stesso valore di default per la voce corrente.

L'esempio seguente mostra come modificare la durata del tratto vocale per modificare il timbro:

```
<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>
```

Combinazione di tag multipli

È possibile combinare il tag `vocal-tract-length` con qualsiasi altro tag SSML supportato da Amazon Polly. Poiché il timbro (la lunghezza del tratto vocale) e l'intonazione sono strettamente connessi, è possibile ottenere risultati ottimali utilizzando sia il tag `vocal-tract-length` che il tag `<prosody pitch>`. Per ottenere una voce più realistica, consigliamo di utilizzare diverse percentuali per i valori di modifica dei due tag. Per ottenere i risultati desiderati, puoi fare più prove con le diverse combinazioni.

L'esempio seguente mostra come combinare i tag.

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

Sussurrare

<amazon:effect name="whispered">

Questo tag è attualmente supportato solo dal formato TTS standard.

Questo tag indica che il testo di input deve essere pronunciato con una voce sussurrata rispetto a quella utilizzata normalmente. Può essere usato con qualsiasi voce del portafoglio Amazon Polly Text-to-Speech.

Questo utilizza la sintassi seguente:

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

Ad esempio:

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

In questo caso, la sintesi vocale pronunciata dal personaggio è sussurrata, mentre la frase "she said" viene pronunciata con la voce normalmente utilizzata per la sintesi vocale di Amazon Polly selezionata.

Puoi aumentare l'effetto "sussurrato" rallentando la velocità dell'elemento prosody fino al 10%, in base all'effetto desiderato.

Ad esempio:

```
<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>
```

Durante la generazione di contrassegni vocali per una voce sussurrata, il flusso audio deve includere anche la voce sussurrata per assicurare la corrispondenza con i contrassegni vocali.

Gestione dei lessici

I lessici di pronuncia consentono di personalizzare la pronuncia delle parole. Amazon Polly fornisce operazioni API che puoi utilizzare per archiviare i lessici in una regione. AWS Questi lessici saranno pertanto specifici di tale determinata regione. Puoi utilizzare uno o più lessici di quella regione quando esegui la sintesi del testo utilizzando l'operazione `SynthesizeSpeech`. In questo modo, il lessico specificato viene applicato al testo di input prima dell'inizio della sintesi. Per ulteriori informazioni, consulta SynthesizeSpeech.

> **Note**
>
> Questi lessici devono essere conformi alle specifiche W3C PLS (Pronunciation Lexicon Specification). Per ulteriori informazioni, consulta Pronunciation Lexicon Specification (PLS) Version 1.0 sul sito Web di W3C.

Di seguito sono elencati alcuni esempi di metodi di utilizzo dei lessici con motori di sintesi vocale:

- Talvolta le parole comuni vengono stilizzate utilizzando i numeri al posto delle lettere, come nel caso di "g3t sm4rt" (get smart). Le persone sono in grado di leggere queste parole correttamente, Tuttavia, un motore Text-to-Speech (TTS) legge il testo letteralmente, pronunciando il nome esattamente come viene scritto. In questi casi puoi utilizzare i lessici per personalizzare la sintesi vocale con Amazon Polly. In questo esempio, puoi specificare un alias (get smart) per il termine "g3t sm4rt" nel lessico.
- Il testo potrebbe includere un acronimo, ad esempio W3C. Puoi utilizzare un lessico per definire un alias per la parola W3C in modo che venga letta nella sua forma estesa e completa (World Wide Web Consortium).

I lessici offrono un ulteriore livello di controllo sul modo in cui Amazon Polly pronuncia le parole non comuni per la lingua selezionata. Ad esempio, puoi specificare la pronuncia utilizzando un alfabeto fonetico. Per ulteriori informazioni, consulta Pronunciation Lexicon Specification (PLS) Version 1.0 sul sito Web di W3C.

Argomenti

- Utilizzo di più lessici
- Caricamento di un lessico

- Applicazione dei lessici (sintesi vocale)
- Filtrare l'elenco dei lessici sulla console
- Scaricamento dei lessici sulla console
- Eliminazione di un lessico

Utilizzo di più lessici

Puoi applicare fino a cinque lessici al testo. Se lo stesso grafema appare in più lessici applicabili al testo, l'ordine in cui vengono applicati può determinare una differenza nella sintesi vocale risultante. Considera, ad esempio, il testo "Ciao, mi chiamo Bob". e due lessemi in diversi lessici che utilizzano entrambi il grafema Bob.

LexA

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

LexB

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

Se i lessici sono elencati nell'ordine LexA seguito da LexB, la sintesi vocale sarà "Ciao, mi chiamo Robert". Se sono elencati nell'ordine LexB seguito da LexA, la sintesi vocale sarà "Ciao, mi chiamo Bobby".

Example - Applicazione di LexA prima di LexB

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

Output vocale: "Ciao, mi chiamo Robert".

Example - Applicazione di LexB prima di LexA

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

Output vocale: "Ciao, mi chiamo Bobby".

Per ulteriori informazioni sull'applicazione di lessici con la console Amazon Polly, consulta Applicazione dei lessici (sintesi vocale).

Caricamento di un lessico

I lessici utilizzati devono essere conformi alla raccomandazione del W3C della Pronunciation Lexicon Specification (PLS). Per ulteriori informazioni, consulta Pronunciation Lexicon Specification (PLS) Version 1.0 sul sito Web di W3C.

Console - Lexicons tab

Per usare un lessico di pronuncia, è necessario prima caricarlo. Nella console è possibile caricare un lessico sia dalla scheda Text-to-Speech (Sintesi vocale) che dalla scheda Lexicons (Lessici).

Le procedure riportate di seguito mostrano come aggiungere lessici che è possibile utilizzare per personalizzare il modo in cui vengono pronunciate parole e frasi non comuni per la lingua scelta.

Per aggiungere un lessico dalla scheda Lexicons (Lessici)

1. Accedi a AWS Management Console e apri la console Amazon Polly all'indirizzo. https://console.aws.amazon.com/polly/
2. Scegli la scheda Lexicons (Lessici).
3. Scegli Upload lexicon (Carica lessico).
4. Fornisci un nome per il lessico e quindi utilizza Choose a lexicon (Scegli un lessico) per individuare il lessico da caricare. È possibile caricare solo file PLS con estensione.pls o .xml.

5. Scegli Upload lexicon (Carica lessico). Se esiste già un lessico con lo stesso nome (indipendentemente dall'estensione .pls o .xml), il caricamento del lessico sovrascrive il lessico esistente.

Console - TTS tab

Per aggiungere un lessico dalla scheda text-to-Speech

1. Accedi a AWS Management Console e apri la console Amazon Polly all'indirizzo. https://console.aws.amazon.com/polly/
2. Scegli la scheda Text-to-Speech (Sintesi vocale).
3. Espandi Additional settings (Impostazioni aggiuntive), attiva Customize pronunciation (Personalizza la pronuncia), scegli Upload lexicon (Carica lessico).
4. Fornisci un nome per il lessico e quindi utilizza Choose a lexicon (Scegli un lessico) per individuare il lessico da caricare. Puoi usare solo i file PLS che utilizzano estensioni .pls e .xml.
5. Scegli Upload lexicon (Carica lessico). Se esiste già un lessico con lo stesso nome (indipendentemente dall'estensione .pls o .xml), il caricamento del lessico sovrascrive il lessico esistente.

AWS CLI - one lexeme

Con Amazon Polly, puoi utilizzare PutLexicon per memorizzare i lessici di pronuncia in una AWS regione specifica per il tuo account. Quindi, puoi specificare uno o più di questi lessici archiviati nella richiesta SynthesizeSpeech da applicare prima che il servizio avvii la sintesi del testo. Per ulteriori informazioni, consulta Gestione dei lessici.

Considera il seguente lessico conforme allo standard PLS W3C.

```
<?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>
```

Tieni presente quanto segue:

- I due attributi specificati nell'elemento `<lexicon>`:
  - L'attributo `xml:lang` specifica il codice della lingua, `en-US`, a cui si applica il lessico. Amazon Polly può utilizzare questo lessico di esempio se la voce specificata nella chiamata `SynthesizeSpeech` ha lo stesso codice della lingua (it-IT).

    > **Note**
    >
    > Puoi utilizzare l'operazione `DescribeVoices` per trovare il codice della lingua associato a una voce.

  - L'attributo `alphabet` specifica `IPA`, a indicare che per le pronunce viene utilizzato l'alfabeto fonetico internazionale (IPA, International Phonetic Alphabet). L'IPA è uno degli alfabeti utilizzati per scrivere le pronunce. Amazon Polly supporta anche l'alfabeto X-SAMPA (Extended Speech Assessment Methods Phonetic Alphabet).

- L'elemento `<lexeme>` descrive la mappatura tra `<grapheme>` (ovvero una rappresentazione testuale della parola) e `<alias>`.

Per testare questo lessico, procedi come indicato di seguito.

1. Salva il lessico con il nome `example.pls`.
2. Esegui il `put-lexicon` AWS CLI comando per memorizzare il lessico (con il nome`w3c`), nella regione us-east-2.

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

3. Esegui il comando `synthesize-speech` per sintetizzare il testo di esempio in un flusso audio (`speech.mp3`) e specifica il parametro opzionale `lexicon-name`.

```
aws polly synthesize-speech \
--text 'W3C is a Consortium' \
--voice-id Joanna \
--output-format mp3 \
--lexicon-names="w3c" \
speech.mp3
```

4. Riproduci il file `speech.mp3` risultante e nota che la parola W3C nel testo è sostituita da World Wide Web Consortium.

Il lessico di esempio precedente utilizza un alias. L'alfabeto IPA menzionato nel lessico non viene utilizzato. Il lessico seguente specifica una pronuncia fonetica utilizzando l'elemento `<phoneme>` con l'alfabeto 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>
```

Segui la stessa procedura per testare questo lessico. Assicurati di specificare un testo di input che contenga la parola «pecan» (ad esempio, «Pecan pie is delicious»).

Consulta le seguenti risorse per esempi di codice aggiuntivi per il funzionamento dell' PutLexicon API:

- Esempio di Java: PutLexicon
- Esempio di Python (Boto3): PutLexicon

AWS CLI - multiple lexemes

Con Amazon Polly, puoi utilizzare PutLexicon per memorizzare i lessici di pronuncia in una AWS regione specifica per il tuo account. Quindi, puoi specificare uno o più di questi lessici archiviati nella richiesta SynthesizeSpeech da applicare prima che il servizio avvii la sintesi del testo. Per ulteriori informazioni, consulta Gestione dei lessici.

In questo esempio il lessema specificato nel lessico si applica esclusivamente al testo di input per la sintesi. Considera il lessico seguente:

```
<?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>
```

Il lessico specifica tre lessemi, due dei quali definiscono un alias per il grafema W3C come indicato di seguito:

- Il primo elemento `<lexeme>` definisce un alias (World Wide Web Consortium).
- Il secondo elemento `<lexeme>` definisce un alias alternativo (WWW Consortium).

Amazon Polly utilizza la prima sostituzione per ogni grafema in un lessico.

Il terzo elemento `<lexeme>` definisce un sostituto (Community) per la parola Consortium.

In primo luogo, è opportuno testare questo lessico. Supponi di voler sintetizzare il seguente testo di esempio per un file audio (`speech.mp3`) e di specificare il lessico in una chiamata a `SynthesizeSpeech`.

```
The W3C is a Consortium
```

In primo luogo `SynthesizeSpeech` applica il lessico come segue:

- Come per il primo lessema, la parola W3C viene sostituita da World Wide Web Consortium. Il testo rivisto viene visualizzato come segue:

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

- L'alias definito nel terzo lessema si applica solo alla parola Consortium che era parte del testo originale, generando il testo seguente:

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

Puoi verificarlo utilizzando AWS CLI quanto segue:

1. Salva il lessico con il nome `example.pls`.
2. Esegui il comando `put-lexicon` per archiviare il lessico con il nome w3c nella regione us-east-2.

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

3. Esegui il comando `list-lexicons` per verificare che il lessico w3c sia incluso nell'elenco dei lessici restituito.

   ```
   aws polly list-lexicons
   ```

4. Esegui il comando `synthesize-speech` per sintetizzare il testo di esempio in un file audio (`speech.mp3`) e specifica il parametro opzionale `lexicon-name`.

   ```
   aws polly synthesize-speech \
   --text 'W3C is a Consortium' \
   --voice-id Joanna \
   ```

```
--output-format mp3 \
--lexicon-names="w3c" \
speech.mp3
```

5. Riproduci il file `speech.mp3` risultante per verificare che la sintesi vocale rifletta le modifiche al testo.

Consulta le seguenti risorse per esempi di codice aggiuntivi per il funzionamento dell' PutLexicon API:

- Esempio di Java: PutLexicon
- Esempio di Python (Boto3): PutLexicon

Applicazione dei lessici (sintesi vocale)

I lessici utilizzati devono essere conformi alla raccomandazione del W3C della Pronunciation Lexicon Specification (PLS). Per ulteriori informazioni, consulta Pronunciation Lexicon Specification (PLS) Version 1.0 sul sito Web di W3C.

Console

La procedura seguente illustra come applicare un lessico al testo di input utilizzando il lessico `W3c.pls` per sostituire "W3C" con "World Wide Web Consortium". Se applichi più lessici al testo, questi vengono applicati in ordine decrescente, con la prima corrispondenza che ha la precedenza sulle successive. Un lessico viene applicato al testo solo se la lingua specificata nel lessico corrisponde alla lingua scelta.

È possibile applicare un lessico al testo semplice o di input SSML.

Example - Applicazione del lessico W3C.pls

Per creare il lessico necessario per questo esercizio, consulta Caricamento di un lessico. Utilizza un editor di testo semplice per creare il lessico W3C.pls visualizzato all'inizio dell'argomento. Ricorda dove salvi il file.

Per applicare il lessico W3C.pls al testo di input

In questo esempio si introduce un lessico per sostituire "W3C" con "World Wide Web Consortium". Confronta i risultati di questo esercizio con quelli di Utilizzo di SSML sulla console sia per l'inglese americano che per un'altra lingua.

1. Accedi a AWS Management Console e apri la console Amazon Polly all'indirizzo. https://console.aws.amazon.com/polly/
2. Esegui una di queste operazioni:
   - Disattiva SSML, quindi digita o incolla questo testo nell'apposita casella di input.

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

   - Attiva SSML, quindi digita o incolla questo testo nell'apposita casella di input.

     ```
     <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. Nell'elenco Language (Lingua) scegli English US (Inglese - Stati Uniti), quindi scegli la voce che desideri utilizzare per questo testo.
4. Espandi Additional settings (Impostazioni aggiuntive) e attiva Customize pronunciation (Personalizza la pronuncia).
5. Nell'elenco dei lessici scegli `W3C (English, US)`.

   Se il lessico `W3C (English, US)` non è elencato, scegli Upload lexicon (Carica lessico) e caricalo, quindi sceglilo nell'elenco. Per creare questo lessico, consulta Caricamento di un lessico.
6. Per ascoltare immediatamente la sintesi vocale, scegli Listen (Ascolta).
7. Per salvare la sintesi vocale in un file
   a. Scegli Download (Scarica).
   b. Per impostare un formato di file diverso, attiva Speech file format settings (Impostazioni formato di file vocale), scegli il formato di file desiderato, quindi scegli Download.

Ripeti le fasi precedenti, ma scegli una lingua diversa e osserva la differenza nell'output.

AWS CLI

In una chiamata a `SynthesizeSpeech` puoi specificare più lessici. In questo caso, il primo lessico specificato (da sinistra a destra) sostituisce gli eventuali lessici precedenti.

Considera i due lessici seguenti e tieni presente che ogni lessico descrive alias diversi per lo stesso grafema W3C.

- Lessico 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>
```

- Lessico 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>
```

Supponi di archiviare questi lessici con i nomi `w3c` e `w3cAlternate`, rispettivamente. Se specifichi i lessici in ordine (`w3c` seguito da `w3cAlternate`) in una chiamata `SynthesizeSpeech`, l'alias per W3C definito nel primo lessico ha la precedenza sul secondo. Per testare i lessici, procedi come indicato di seguito.

1. Salva i lessici in locale in file denominati `w3c.pls` e `w3cAlternate.pls`.

2. Carica questi lessici usando il `put-lexicon` AWS CLI comando.

   - Carica il lessico `w3c.pls` e memorizzalo con il nome `w3c`.

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

   - Carica il lessico `w3cAlternate.pls` nel servizio con il nome `w3cAlternate`.

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

3. Esegui il comando `synthesize-speech` per sintetizzare il testo di esempio in un flusso audio (`speech.mp3`) e specifica entrambi i lessici utilizzando il parametro `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. Testa il file `speech.mp3` risultante, che dovrebbe essere simile a quanto riportato di seguito:

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

Filtrare l'elenco dei lessici sulla console

La procedura riportata di seguito illustra come filtrare l'elenco dei lessici in modo che vengano visualizzati solo i lessici di una determinata lingua.

Console

Per filtrare i lessici elencati per lingua

1. Accedi a AWS Management Console e apri la console Amazon Polly all'indirizzo. https://console.aws.amazon.com/polly/
2. Scegli la scheda Lexicons (Lessici).

3. Scegli Any language (Qualsiasi lingua).
4. Nell'elenco delle lingue scegli quella per la quale desideri applicare il filtro.

   Nell'elenco vengono visualizzati solo i lessici per la lingua scelta.

AWS CLI

Amazon Polly fornisce il funzionamento dell'ListLexiconsAPI che puoi utilizzare per ottenere l'elenco dei lessici di pronuncia nel tuo account in una regione specifica. AWS La AWS CLI chiamata seguente elenca i lessici del tuo account nella regione us-east-2.

```
aws polly list-lexicons
```

Di seguito è riportata una risposta di esempio, che mostra due lessici denominati `w3c` e `tomato`. Per ogni lessico, la risposta restituisce i metadata come il codice della lingua a cui si applica il lessico, il numero di lessemi definiti nel lessico, le dimensioni in byte e così via. Il codice della lingua descrive una lingua e le impostazioni locali a cui si applicano i lessemi definiti nel lessico.

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

Le seguenti risorse contengono informazioni aggiuntive sull'operazione: ListLexicons

- Esempio di Java: ListLexicons
- Esempio di Python (Boto3): ListLexicon

Scaricamento dei lessici sulla console

Nella procedura seguente viene descritto come scaricare uno o più lessici. Puoi aggiungere, rimuovere o modificare le voci del lessico nel file e poi caricarlo di nuovo per mantenere il tuo lessico. up-to-date

Console

Per scaricare uno o più lessici

1. Accedi a AWS Management Console e apri la console Amazon Polly all'indirizzo. https://console.aws.amazon.com/polly/
2. Scegli la scheda Lexicons (Lessici).
3. Scegli uno o più lessici da scaricare.
   a. Per scaricare un singolo lessico, scegli il nome dall'elenco.
   b. Per scaricare più lessici come un singolo file di archivio compresso, seleziona la casella di controllo accanto a ogni voce nell'elenco che desideri scaricare.
4. Scegli Download (Scarica).
5. Apri la cartella in cui desideri scaricare il lessico.
6. Scegli Save (Salva).

AWS CLI

In Amazon Polly è disponibile l'operazione dell'API GetLexicon per recuperare i contenuti di un lessico di pronuncia archiviato nel tuo account in una determinata regione.

Il `get-lexicon` AWS CLI comando seguente recupera il contenuto del `example` lessico.

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

Se nel tuo account non è ancora archiviato alcun lessico, puoi utilizzare l'operazione `PutLexicon` per archiviarne uno. Per ulteriori informazioni, consulta Caricamento di un lessico.

Di seguito è riportata una risposta di esempio. Oltre ai contenuti del lessico, la risposta restituisce i metadata, ad esempio il codice della lingua a cui si applica il lessico, il numero di lessemi definito nel lessico, l'Amazon Resource Name (ARN) della risorsa e le dimensioni del lessico in byte. Il valore `LastModified` è un timestamp 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
    }
}
```

Le seguenti risorse contengono esempi di codice aggiuntivi per l' GetLexicon operazione:

- Esempio di Java: GetLexicon
- Esempio di Python (Boto3): GetLexicon

Eliminazione di un lessico

Nella procedura seguente viene descritto come eliminare un lessico. Dopo aver eliminato il lessico, devi aggiungerlo di nuovo prima di poterlo riutilizzare. Puoi eliminare uno o più lessici contemporaneamente selezionando le caselle di controllo accanto ai singoli lessici.

Console

Per eliminare un lessico

1. Accedi a AWS Management Console e apri la console Amazon Polly all'indirizzo. https://console.aws.amazon.com/polly/
2. Scegli la scheda Lexicons (Lessici).
3. Scegli uno o più lessici da eliminare dall'elenco.
4. Scegliere Delete (Elimina).
5. Inserisci il testo di conferma e poi scegli Delete (Elimina) per rimuovere il lessico dalla regione o Cancel (Annulla) per conservarlo.

AWS CLI

Amazon Polly fornisce l'operazione DeleteLexicon API per eliminare un lessico di pronuncia da una AWS regione specifica del tuo account. Quanto segue AWS CLI elimina il lessico specificato.

L' AWS CLI esempio seguente è formattato per Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione Unix barra rovesciata (\) al termine di ogni riga con un accento circonflesso (^) e usa virgolette (") attorno al testo di input con virgolette singole (') per i tag interni.

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

Le seguenti risorse contengono informazioni aggiuntive sull'operazione: DeleteLexicon

- Esempio di Java: DeleteLexicon
- Esempio di Python (Boto3): DeleteLexicon

File audio lunghi

Per creare file TTS per passaggi di testo lunghi, utilizza la funzionalità di sintesi asincrona di Amazon Polly. Questo utilizza i tre `SpeechSynthesisTask` APIs:

- `StartSpeechSynthesisTask`: avvia una nuova attività di sintesi.
- `GetSpeechSynthesisTask`: restituisce dettagli su un'attività di sintesi inviata in precedenza.
- `ListSpeechSynthesisTasks`: elenca tutte le attività di sintesi inviate.

L'operazione `SynthesizeSpeech` produce audio quasi in tempo reale, con latenza relativamente ridotta nella maggior parte dei casi. A questo scopo, l'operazione può sintetizzare solo 3000 caratteri.

La funzionalità di sintesi asincrona di Amazon Polly consente di risolvere il problema dell'elaborazione di un documento di testo di dimensioni maggiori, modificando il modo in cui il documento viene sintetizzato e restituito. Quando una richiesta di sintesi viene effettuata inviando testo di input mediante `StartSpeechSynthesisTask`, Amazon Polly mette in coda le richieste, quindi le elabora in modo asincrono in background non appena sono disponibili risorse del sistema. Amazon Polly carica quindi la conseguente sintesi vocale o i contrassegni vocali vengono trasmessi direttamente sul bucket Amazon Simple Storage Service (Amazon S3) (obbligatorio) e notifica la disponibilità del file completato tramite l'argomento SNS (opzionale).

In questo modo, tutte le funzionalità ad eccezione dell'elaborazione quasi in tempo reale sono disponibili per testi lunghi fino a 100.000 caratteri fatturabili (o 200.000 caratteri totali).

Per sintetizzare un documento utilizzando questo metodo, è necessario disporre di un bucket Amazon S3 scrivibile su cui è possibile salvare il file audio. Puoi ricevere una notifica quando l'audio sintetizzato è pronto, specificando un identificatore Argomento SNS opzionale. Quando l'attività di sintesi viene completata, Amazon Polly pubblicherà un messaggio su tale argomento. Questo messaggio può contenere inoltre utili informazioni sugli errori nel caso in cui l'attività di sintesi non abbia avuto esito positivo. Per eseguire questa operazione, accertati che l'utente che crea l'attività di sintesi possa anche pubblicare sull'argomento SNS. Consulta la documentazione di Amazon SNS per ulteriori informazioni su come creare e sottoscrivere un argomento SNS.

Encryption (Crittografia)

Se lo desideri, puoi memorizzare il file di output in un modulo crittografato nel bucket S3. Per fare ciò, abilita la crittografia del bucket Amazon S3 che utilizza una delle cifrature di blocco più complesse disponibili, lo standard di crittografia avanzata a 256 bit (AES-256).

Argomenti

- Configurazione della politica IAM per la sintesi asincrona
- Creazione di file audio lunghi

Configurazione della politica IAM per la sintesi asincrona

Per utilizzare la funzionalità di sintesi asincrona, è necessario disporre di una policy IAM che consenta quanto segue:

- utilizzo di nuove operazioni Amazon Polly
- scrittura sul bucket S3 di output
- pubblicazione sull'argomento SNS di stato [facoltativo]

La policy seguente concede solo le autorizzazioni necessarie per la sintesi asincrona e può essere collegata all'utente IAM.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "polly:StartSpeechSynthesisTask",
                "polly:GetSpeechSynthesisTask",
                "polly:ListSpeechSynthesisTasks"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::bucket-name/*"
        },
        {
            "Effect": "Allow",
            "Action": "sns:Publish",
            "Resource": "arn:aws:sns:us-east-1:account:topic"
```

```
        }
    ]
}
```

Creazione di file audio lunghi

Puoi utilizzare la console di Amazon Polly per creare sintesi vocali lunghe mediante sintesi asincrona con la stessa funzionalità utilizzabile con l'interfaccia a riga di comando di AWS CLI. Questa operazione viene eseguita utilizzando la scheda Text-to-Speech (Sintesi vocale) in modo molto analogo a qualsiasi altra sintesi.

Console

Anche l'altra funzionalità di sintesi asincrona è disponibile tramite la console. La scheda S3 synthesis tasks (Attività di sintesi S3) riflette la funzionalità `ListSpeechSynthesisTasks` visualizzando tutte le attività salvate nel bucket S3 e consentendo di filtrarle in base alle esigenze. Facendo clic su una specifica attività singola, ne vengono visualizzati i dettagli, in modo analogo alla funzionalità `GetSpeechSynthesisTask`.

Per sintetizzare un testo di grandi dimensioni utilizzando la console Amazon Polly

1. Accedi a AWS Management Console e apri la console Amazon Polly all'indirizzo. https://console.aws.amazon.com/polly/
2. Scegli la scheda Text-to-Speech (Sintesi vocale). Seleziona Long Form come motore, se appropriato.
3. Con SSML attivato o disattivato, digita o incolla il testo nella casella di input.
4. Scegli la lingua, la regione e la voce per il testo.
5. Scegli Save in S3 (Salva in S3).

   > **Note**
   >
   > Entrambe le opzioni Download e Ascolta sono disattivate se la lunghezza del testo supera il limite di 3.000 caratteri per il funzionamento in tempo reale`SynthesizeSpeech`.

6. La console apre un modulo in modo che sia possibile scegliere dove archiviare il file di output.

   a. Inserisci il nome del bucket Amazon S3 di destinazione.

b. Inserire facoltativamente la chiave di prefisso dell'output.

> **Note**
>
> Il bucket S3 di output deve essere scrivibile.

c. Se si desidera ricevere una notifica quando l'attività di sintesi viene completata, specificare l'identificatore argomento SNS facoltativo.

> **Note**
>
> L'SNS deve essere aperto per la pubblicazione da parte dell'utente della console corrente per utilizzare questa opzione. Per ulteriori informazioni, consulta Amazon Simple Notification Service (SNS).

d. Scegli Save in S3 (Salva in S3).

Per recuperare informazioni sulle attività di sintesi vocale

1. Nella console, scegliere la scheda S3 Synthesis Tasks (Attività sintesi vocale).
2. Le attività vengono visualizzate in ordine di data. Per filtrare le attività, in base allo stato, scegli All statuses (Tutti gli stati) e quindi scegli lo stato da utilizzare.
3. Per visualizzare i dettagli di un'attività specifica, scegliere il Task ID (ID attività) collegato.

AWS CLI

La funzionalità di sintesi asincrona di Amazon Polly ne utilizza tre `SpeechSynthesisTask` APIs per lavorare con grandi quantità di testo:

- `StartSpeechSynthesisTask`: avvia una nuova attività di sintesi.
- `GetSpeechSynthesisTask`: restituisce dettagli su un'attività di sintesi inviata in precedenza.
- `ListSpeechSynthesisTasks`: elenca tutte le attività di sintesi inviate.

Sintesi di grandi porzioni di testo (**`StartSpeechSynthesisTask`**)

Quando desideri creare un file audio di dimensioni superiori rispetto a quello che puoi creare con l'operazione `SynthesizeSpeech` in tempo reale, utilizza l'operazione

`StartSpeechSynthesisTask`. Oltre agli argomenti necessari per l'operazione `SynthesizeSpeech`, `StartSpeechSynthesisTask` richiede anche il nome di un bucket Amazon S3. Sono inoltre disponibili altri due argomenti facoltativi: un prefisso della chiave per il file di output e l'ARN per un argomento SNS se desideri ricevere la notifica di stato sull'attività.

- `OutputS3BucketName`: il nome del bucket Amazon S3; in cui la sintesi deve essere caricata. Questo bucket deve trovarsi nella stessa regione del servizio Amazon Polly. Inoltre, l'utente IAM utilizzato per effettuare la chiamata deve avere accesso al bucket. [Campo obbligatorio]
- `OutputS3KeyPrefix`: prefisso della chiave per il file di output. Utilizza questo parametro se desideri salvare i file di sintesi vocale di output in una chiave simile a una directory personalizzata nel tuo bucket. [Facoltativo].
- `SnsTopicArn`: L'argomento SNS ARN da utilizzare se si desidera ricevere notifiche sullo stato dell'attività. Questo argomento SNS deve trovarsi nella stessa regione del servizio Amazon Polly. Inoltre, l'utente IAM utilizzato per effettuare la chiamata deve avere accesso all'argomento. [Facoltativo].

Ad esempio, l'esempio seguente può essere utilizzato per eseguire il `start-speech-synthesis-task` AWS CLI comando nella regione Stati Uniti orientali (Ohio):

L' AWS CLI esempio seguente è formattato per Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione Unix barra rovesciata (\) al termine di ogni riga con un accento circonflesso (^) e usa virgolette (") attorno al testo di input con virgolette singole (') per i tag interni.

```
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
```

Questo determinerà una risposta simile a questa:

```
"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"
}
```

L'operazione `start-speech-synthesis-task` restituisce diversi nuovi campi:

- `OutputUri`: il percorso dei file di sintesi vocale di output.
- `TaskId`: un identificatore unico per l'attività di sintesi vocale generata da Amazon Polly.
- `CreationTime`: un timestamp che specifica quando l'attività è stata inviata inizialmente.
- `RequestCharacters`: il numero di caratteri fatturabili nell'attività.
- `TaskStatus`: fornisce informazioni sullo stato dell'attività inviata.

  Quando la tua attività è stata inviata, lo stato iniziale mostrerà `scheduled`. Quando Amazon Polly avvia l'elaborazione dell'attività, lo stato passerà a `inProgress` e in seguito a `completed` o `failed`. Se l'operazione non riesce, verrà restituito un messaggio di errore quando si chiama l' GetSpeechSynthesisTask operazione or. ListSpeechSynthesisTasks

Una volta completata l'attività, il file di sintesi vocale è disponibile nel percorso specificato in `OutputUri`.

Recupero di informazioni sull'attività di sintesi vocale

Puoi ottenere informazioni su un'attività, ad esempio gli errori, lo stato e così via, utilizzando l'operazione `GetSpeechSynthesisTask`. Per eseguire questa operazione, è necessario il `task-id` restituito da `StartSpeechSynthesisTask`.

Ad esempio, è possibile utilizzare l'esempio seguente per eseguire il `get-speech-synthesis-task` AWS CLI comando:

```
aws polly get-speech-synthesis-task \
--region us-east-2 \
--endpoint-url "https:// polly.us-east-2.amazonaws.com/" \
--task-id task identifier
```

Puoi inoltre elencare tutte le attività di sintesi vocale che hai eseguito nella regione corrente utilizzando l'operazione `ListSpeechSynthesisTasks`.

Ad esempio, l'esempio seguente può essere utilizzato per eseguire il `list-speech-synthesis-tasks` AWS CLI comando:

```
aws polly list-speech-synthesis-tasks \
--region us-east-2 \
--endpoint-url "https:// polly.us-east-2.amazonaws.com/"
```

Quote in Amazon Polly

Amazon Polly applica delle quote al traffico dei clienti rifiutando richieste eccessive. La quota predefinita per la richiesta `SynthesizeSpeech` con voci standard è di 80 transazioni al secondo (tps), in una singola regione, per una singola Account AWS. Se i limiti non sono aumentati e se ne hai generato 100 richieste `SynthesizeSpeech` al secondo utilizzando una voce standard, 80 richieste al secondo andrebbero a buon fine e 20 richieste al secondo verrebbero limitate da Amazon Polly. Queste richieste restituirebbero una risposta con lo stato HTTP 400 e un'intestazione di risposta che indica `ThrottlingException`. Amazon Polly inoltre limita il traffico verso tutte le operazioni in base alla frequenza delle richieste.

Esempi di limiti di sintesi vocale

- Sintetizza le prime 24 lettere dell'alfabeto inglese una lettera alla volta. Se la sintesi di ogni lettera richiedesse meno di 50 millisecondi, con un limite operativo di otto tps, la sintesi di 24 lettere richiederebbe almeno tre secondi. Durante quel periodo, potresti sintetizzare fino a otto lettere al secondo. Eventuali ulteriori richieste verrebbero limitate. Poiché le richieste durano poco tempo, verrebbero sintetizzate in serie senza sovrapposizioni.
- Sintetizza 16 paragrafi di testo. Se ogni paragrafo fosse sintetizzato e ricevuto completamente sul lato client in massimo due secondi, con un limite di operazioni di otto richieste simultanee, ci vorrebbero almeno quattro secondi per sintetizzare tutti i 16 articoli. Nel primo secondo, puoi avviare fino a otto richieste. Durante le richieste simultanee, qualsiasi tentativo di avviare una nuova sintesi verrebbe limitato a causa del limite di concorrenza. È possibile sintetizzare gli otto paragrafi rimanenti dopo i primi due secondi, al termine del primo batch di richieste.

Tieni a mente i seguenti limiti quando usi Amazon Polly.

Argomenti

- Regioni supportate
- Quote e velocità di limitazione
- Lessici di pronuncia
- SynthesizeSpeech Operazioni API
- SpeechSynthesisTask Operazioni API
- SSML (Speech Synthesis Markup Language)

Regioni supportate

Per un elenco delle AWS regioni in cui è disponibile Amazon Polly, consulta Amazon Polly Endpoints and Quotas nel. Riferimenti generali di Amazon Web Services

- Per le regioni che supportano le voci generative, consulta Generative voices.
- Per le regioni che supportano le voci in formato lungo, vedi Voci in formato lungo.
- Per le regioni che supportano le voci neurali, vedi the section called “Compatibilità tra funzionalità e regioni” per il TTS neurale.

Quote e velocità di limitazione

La tabella seguente definisce i tassi di limitazione per ogni operazione di Amazon Polly. È possibile utilizzare il AWS Management Console per richiedere aumenti delle quote per le quote regolabili quando necessario.

Operazione	Limite
Lessico	

`DeleteLexicon`
`PutLexicon`
`GetLexicon`
`ListLexicons`
	Ogni 2 transazioni al secondo (tps) di queste operazioni combinate.
	Il massimo aumento delle prestazioni consentito è pari a 4 tps.

Voce	
`DescribeVoices`	80 tps con un limite di aumento delle prestazioni pari a 100 tps
`SynthesizeSpeech`	Voce generativa: 8 tps Voce lunga: 8 tps con un limite di burst di 10 tps Voce neurale: 8 tps con un limite di aumento delle prestazioni pari a 10 tps

Operazione	Limite
	Voce standard: 80 tps con un limite di aumento delle prestazioni pari a 100 tps
`StartSpeechSynthesisTask`	Voce generativa: 1 tps Voce lunga: 1 tps Voce neurale: 10 tps Voce standard: 10 tps con un limite di aumento delle prestazioni pari a 12 tps
`GetSynthesizeSpeechTask` e `ListSynthesizeSpeechTask`	Il massimo consentito è pari a 10 tps

Richieste simultanee

Per quanto riguarda la voce generativa, Amazon Polly supporta fino a 26 richieste simultanee. Per le chiamate in formato esteso, Amazon Polly supporta fino a 26 richieste simultanee. Per voce neurale, Amazon Polly supporta 8 tps con un limite di burst di 10 tps, per un massimo di 18 richieste simultanee. Amazon Polly supporta anche i limiti per le richieste simultanee. Per voce standard, Amazon Polly supporta 80 tps per un massimo di 80 richieste simultanee.

Best practice per mitigare la limitazione

- Limitazioni di nuovi tentativi con backoff e jitter in modo da poter distribuire il carico su un breve periodo di tempo e gestire picchi di utilizzo imprevisti senza compromettere la disponibilità. AWS Code Sample Catalog è già configurato per eseguire questa operazione di default in molti linguaggi di programmazione. Visita comportamento dei nuovi tentativi di funzionalità per vedere i dettagli.
- Usa le metriche Amazon Polly. Amazon Polly pubblica automaticamente su per CloudWatch analizzare l'utilizzo corrente e prevedere la crescita dell'utilizzo.

> **Note**
>
> Prima di richiedere un aumento della quota (ove applicabile), calcola il fabbisogno di tps seguendo le linee guida in questa pagina. Amazon Polly si assicura solo le risorse di calcolo necessarie in base alla domanda dei clienti per mantenere bassi i costi.

Lessici di pronuncia

- Puoi archiviare fino a 100 lessici per account.
- I nomi dei lessici possono essere costituiti da stringhe alfanumeriche con una lunghezza massima di 20 caratteri.
- Ogni lessico può avere una dimensione massima di 40.000 caratteri. (Tieni presente che la dimensione del lessico influisce sulla latenza dell'operazione.) SynthesizeSpeech
- In un lessico puoi specificare fino a 100 caratteri per ogni sostituzione di <fonema> o <alias>.

Per informazioni sull'uso dei lessici, consulta Gestione dei lessici.

SynthesizeSpeech Operazioni API

Quando stimi l'utilizzo di `SynthesizeSpeech`, tieni presente che l'audio prodotto da Amazon Polly, in particolare per le applicazioni interattive, richiede in genere almeno alcuni secondi per essere riprodotto. Questo riduce il tasso di richieste a `SynthesizeSpeech`, anche per un numero significativo di consumatori simultanei. Inoltre, Amazon Polly limita le richieste `SynthesizeSpeech` in base al numero di richieste simultanee che sintetizza. Non esiste un'impostazione separata per le richieste simultanee. Il limite di richieste simultanee ha sempre lo stesso valore del numero di tps consentiti e si adatta ad esso.

Esempio di applicazione per un breve racconto. Puoi usare Amazon Polly per creare un'applicazione che riproduca una serie di racconti. Con questo tipo di app, inizia la riproduzione della prima storia, poi quella successiva e così via, fino a quando un utente non chiude l'applicazione. Ogni storia richiederebbe circa 0,5 secondi per essere sintetizzata e 10 secondi per essere riprodotta. In questo scenario, potresti aspettarti una chiamata a `SynthesizeSpeech` ogni 10 secondi trascorsi dal cliente nell'utilizzo dell'applicazione. Ciò si tradurrebbe in una chiamata al secondo per ogni 10 clienti che utilizzano contemporaneamente l'applicazione. Se avessi 1000 clienti che

utilizzano contemporaneamente l'applicazione, potresti aspettarti una tariffa media di chiamata a `SynthesizeSpeech` di sole 100 transazioni al secondo.

Nota i seguenti limiti correlati all'uso dell'operazione dell'API `SynthesizeSpeech`:

- Il testo di input può avere una dimensione massima di 3000 caratteri fatturati (6000 caratteri totali). I tag SSML non vengono conteggiati come caratteri fatturati.
- Puoi specificare fino a cinque lessici da applicare al testo di input.
- Lo streaming dell'audio di output (sintesi) è limitato a 10 minuti. Una volta raggiunto questo limite, qualsiasi sintesi vocale rimanente viene interrotta.

Per ulteriori informazioni, consulta SynthesizeSpeech.

> **Note**
>
> Alcune limitazioni dell'operazione API `SynthesizeSpeech` possono essere ignorate utilizzando l'operazione API `StartSythensizeSpeechTask`. Per ulteriori informazioni, consulta File audio lunghi.

SpeechSynthesisTask Operazioni API

Nota i seguenti limiti correlati all'uso delle operazioni API `StartSpeechSynthesisTask`, `GetSpeechSynthesisTask` e `ListSpeechSynthesisTasks`:

- Il testo di input può avere una dimensione massima di 100,000 caratteri fatturati (200,000 caratteri totali). I tag SSML non vengono conteggiati come caratteri fatturati.
- Puoi specificare fino a cinque lessici da applicare al testo di input.

SSML (Speech Synthesis Markup Language)

Nota i seguenti limiti correlati all'uso di SSML:

- Non sono supportati i tag `<audio>`, `<lexicon>`, `<lookup>` e `<voice>`.
- Gli elementi `<break>` possono specificare una durata massima di 10 secondi ciascuno.
- Il tag `<prosody>` non supporta i valori per l'attributo di velocità inferiori a -80%.

Per ulteriori informazioni, consulta Generazione vocale da documenti SSML.

Codice di esempio e applicazioni per Amazon Polly

In questa sezione vengono forniti esempi di codice e applicazioni che è possibile utilizzare per esplorare Amazon Polly.

L'argomento Sample Code (Codice di esempio) contiene frammenti di codice organizzati in base al linguaggio di programmazione e separati in esempi per le diverse funzionalità di Amazon Polly. L'argomento Example Application (Applicazione di esempio) contiene le applicazioni, organizzate in base al linguaggio di programmazione, che possono essere utilizzate in modo indipendente per esplorare Amazon Polly.

Prima di iniziare a utilizzare questi esempi, è consigliabile leggere Come funziona Amazon Polly e seguire le fasi descritte in Guida introduttiva ad Amazon Polly.

Argomenti

- Esempi di Java
- Esempi Python
- Esempio di Java
- Esempio di Python (HTML5 Client e Python Server)
- Esempio iOS
- Esempio Android

Esempi di Java

I seguenti esempi di codice mostrano come utilizzare le applicazioni basate su Java per eseguire diverse attività con Amazon Polly. Questi esempi non sono completi, ma possono essere inclusi in applicazioni Java più grandi che utilizzano AWS SDK per Java.

Frammenti di codice

- DeleteLexicon
- DescribeVoices
- GetLexicon
- ListLexicons
- PutLexicon

- StartSpeechSynthesisTask
- Contrassegni vocali
- SynthesizeSpeech

DeleteLexicon

Il seguente esempio di codice Java mostra come utilizzare le applicazioni basate su Java per eliminare un lessico specifico archiviato in una regione. AWS Un lessico che è stato eliminato non è disponibile per la sintesi vocale, né può essere recuperato utilizzando o. `GetLexicon` `ListLexicon` APIs

Per ulteriori informazioni su questa operazione, consulta il riferimento per l'API `DeleteLexicon`.

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

Il seguente esempio di codice Java mostra come utilizzare le applicazioni basate su Java per ottenere un elenco delle voci disponibili per l'uso al momento della richiesta di sintesi vocale. È anche possibile specificare un codice della lingua per filtrare le voci disponibili. Se ad esempio si specifica it-IT, l'operazione restituisce un elenco di tutte le voci disponibili per l'italiano.

Per ulteriori informazioni su questa operazione, consulta il riferimento per l'API `DescribeVoices`.

```
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

Il seguente esempio di codice Java mostra come utilizzare le applicazioni basate su Java per produrre il contenuto di un lessico di pronuncia specifico memorizzato in una regione. AWS

Per ulteriori informazioni su questa operazione, consulta il riferimento per l'API `GetLexicon`.

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

Il seguente esempio di codice Java mostra come utilizzare le applicazioni basate su Java per produrre un elenco di lessici di pronuncia memorizzati in una regione. AWS

Per ulteriori informazioni su questa operazione, consulta il riferimento per l'API `ListLexicons`.

```
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

Il seguente esempio di codice Java mostra come utilizzare le applicazioni basate su Java per memorizzare un lessico di pronuncia in una regione. AWS

Per ulteriori informazioni su questa operazione, consulta il riferimento per l'API `PutLexicon`.

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

Il seguente esempio di codice Java mostra come usare le applicazioni basate su Java per eseguire una sintesi vocale lunga (fino a 100.000 caratteri fatturati) e archiviarla direttamente in un bucket Amazon S3.

Per ulteriori informazioni, consulta il riferimento per l'API `StartSpeechSynthesisTask`.

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();
    }

}
```

Contrassegni vocali

Il seguente esempio di codice mostra come utilizzare le applicazioni basate su Java per sintetizzare i contrassegni vocali per il testo immesso. Questa funzionalità utilizza l'API. SynthesizeSpeech

Per ulteriori informazioni su questa funzionalità, consulta Segni vocali.

Per ulteriori informazioni sull'API, consulta il riferimento per l'API `SynthesizeSpeech`.

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

Il seguente esempio di codice Java mostra come utilizzare le applicazioni basate su Java per eseguire la sintesi vocale con testi più corti per l'elaborazione in tempo reale.

Per ulteriori informazioni, consulta il riferimento per l'API SynthesizeSpeech.

```
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);
        }
    }
```

```
}
```

Esempi Python

I seguenti esempi di codice mostrano come utilizzare le applicazioni basate su Python (boto3) per eseguire diverse attività con Amazon Polly. Questi esempi non sono completi, ma possono essere inclusi in applicazioni Python più grandi che utilizzano AWS SDK for Python (Boto).

Frammenti di codice

- DeleteLexicon
- GetLexicon
- ListLexicon
- PutLexicon
- StartSpeechSynthesisTask
- SynthesizeSpeech

DeleteLexicon

Il seguente esempio di codice Python usa AWS SDK for Python (Boto) per eliminare un lessico nella regione specificata nella configurazione locale. AWS Nell'esempio viene eliminato solo il lessico specificato e viene richiesto di confermare se desideri procedere prima che il lessico venga effettivamente eliminato.

Il seguente esempio di codice utilizza credenziali predefinite memorizzate nel file di configurazione AWS SDK. Per informazioni sulla creazione del file di configurazione, consulta Configurazione del AWS CLI.

Per ulteriori informazioni su questa operazione, consulta il riferimento per l'API `DeleteLexicon`.

```
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

Il seguente codice Python utilizza il AWS SDK for Python (Boto) per recuperare tutti i lessici memorizzati in una regione. AWS Nell'esempio viene accettato un nome di lessico come parametro della riga di comando e viene recuperato solo tale lessico, stampando il percorso temporaneo in cui è stato salvato in locale.

Il seguente esempio di codice utilizza credenziali predefinite memorizzate nel file di configurazione SDK. AWS Per informazioni sulla creazione del file di configurazione, consulta Configurazione del AWS CLI.

Per ulteriori informazioni su questa operazione, consulta il riferimento per l'API `GetLexicon`.

```
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

Il seguente esempio di codice Python usa AWS SDK for Python (Boto) per elencare i lessici del tuo account nella regione specificata nella configurazione locale. AWS Per informazioni sulla creazione del file di configurazione, consulta Configurazione del AWS CLI.

Per ulteriori informazioni su questa operazione, consulta il riferimento per l'API `ListLexicons`.

```
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

Il seguente esempio di codice mostra come utilizzare le applicazioni basate su Python (boto3) per memorizzare un lessico di pronuncia in una regione. AWS

Per ulteriori informazioni su questa operazione, consulta il riferimento per l'API `PutLexicon`.

Tieni presente quanto segue:

- Devi aggiornare il codice fornendo un nome file di lessico locale e un nome di lessico archiviato.
- Nell'esempio si presuppone che i file di lessico sono stati creati in una sottodirectory denominata `pls`. Dovrai aggiornare il percorso come opportuno.

Il seguente esempio di codice utilizza credenziali predefinite memorizzate nel file di configurazione SDK. AWS Per informazioni sulla creazione del file di configurazione, consulta Configurazione del AWS CLI.

Per ulteriori informazioni su questa operazione, consulta il riferimento per l'API `PutLexicon`.

```
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

Il seguente esempio di codice Python usa AWS SDK for Python (Boto) per elencare i lessici del tuo account nella regione specificata nella configurazione locale. AWS Per informazioni sulla creazione del file di configurazione, consulta Configurazione del AWS CLI.

Per ulteriori informazioni, consulta il riferimento per l'API `StartSpeechSynthesisTask`.

```
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

Il seguente esempio di codice Python utilizza la AWS SDK for Python (Boto) sintesi vocale con testi più brevi per un'elaborazione quasi in tempo reale. Per ulteriori informazioni, consulta il riferimento per l'operazione API SynthesizeSpeech.

In questo esempio viene utilizzata una breve stringa di testo semplice. È possibile utilizzare il testo SSML per un maggiore controllo sull'output. Per ulteriori informazioni, consulta Generazione vocale da documenti 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()
```

Esempio di Java

Questo esempio mostra come utilizzare Amazon Polly per lo streaming vocale da un'applicazione basata su Java. Nell'esempio viene utilizzato AWS SDK per Java per leggere il testo specificato utilizzando una voce selezionata da un elenco.

Il codice mostrato copre le attività principali, ma esegue solo un controllo minimo degli errori. Se Amazon Polly rileva un errore, l'applicazione viene terminata.

Per eseguire questa applicazione di esempio, è necessario quanto elencato di seguito:

- Java Development Kit (JDK) di Java 8
- AWS SDK per Java
- Apache Maven

Per testare l'applicazione

1. Verifica che la variabile di ambiente JAVA_HOME sia impostata per il JDK.

   Ad esempio, se hai installato JDK 1.8.0_121 su Windows nel percorso `C:\Program Files\Java\jdk1.8.0_121`, digita quanto segue al prompt dei comandi:

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

   Se hai installato JDK 1.8.0_121 su Linux nel percorso `/usr/lib/jvm/java8-openjdk-amd64`, digita quanto segue al prompt dei comandi:

   ```
   export JAVA_HOME=/usr/lib/jvm/java8-openjdk-amd64
   ```

2. Imposta le variabili di ambiente Maven per eseguire Maven dalla riga di comando.

   Ad esempio, se hai installato Maven 3.3.9 su Windows nel percorso `C:\Program Files\apache-maven-3.3.9`, digita quanto segue al prompt dei comandi:

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

   Se hai installato Maven 3.3.9 su Linux nel percorso `/home/ec2-user/opt/apache-maven-3.3.9`, digita quanto segue al prompt dei comandi:

   ```
   export M2_HOME=/home/ec2-user/opt/apache-maven-3.3.9
   export M2=$M2_HOME/bin
   export PATH=$M2:$PATH
   ```

3. Crea una nuova directory denominata `polly-java-demo`.
4. Nella directory `polly-java-demo` crea un nuovo file denominato `pom.xml` e incollavi il seguente codice:

   ```
   <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. Crea una nuova directory denominata `polly` nel percorso `src/main/java/com/amazonaws/demos`.

6. Nella directory `polly` crea un nuovo file di origine Java denominato `PollyDemo.java` e incolla il seguente codice:

```
package com.amazonaws.demos.polly;

import java.io.IOException;
import java.io.InputStream;

import com.amazonaws.ClientConfiguration;
import com.amazonaws.auth.DefaultAWSCredentialsProviderChain;
import com.amazonaws.regions.Region;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.polly.AmazonPollyClient;
import com.amazonaws.services.polly.model.DescribeVoicesRequest;
import com.amazonaws.services.polly.model.DescribeVoicesResult;
import com.amazonaws.services.polly.model.OutputFormat;
import com.amazonaws.services.polly.model.SynthesizeSpeechRequest;
import com.amazonaws.services.polly.model.SynthesizeSpeechResult;
import com.amazonaws.services.polly.model.Voice;

import javazoom.jl.player.advanced.AdvancedPlayer;
import javazoom.jl.player.advanced.PlaybackEvent;
import javazoom.jl.player.advanced.PlaybackListener;

public class PollyDemo {

 private final AmazonPollyClient polly;
 private final Voice voice;
 private static final String SAMPLE = "Congratulations. You have successfully built
 this working demo
 of Amazon Polly in Java. Have fun building voice enabled apps with Amazon Polly
 (that's me!), and always
 look at the AWS website for tips and tricks on using Amazon Polly and other great
 services from AWS";

 public PollyDemo(Region region) {
  // create an Amazon Polly client in a specific region
  polly = new AmazonPollyClient(new DefaultAWSCredentialsProviderChain(),
  new ClientConfiguration());
  polly.setRegion(region);
  // Create describe voices request.
  DescribeVoicesRequest describeVoicesRequest = new DescribeVoicesRequest();
```

```
 // Synchronously ask Amazon Polly to describe available TTS voices.
 DescribeVoicesResult describeVoicesResult =
polly.describeVoices(describeVoicesRequest);
 voice = describeVoicesResult.getVoices().get(0);
}

public InputStream synthesize(String text, OutputFormat format) throws IOException
{
 SynthesizeSpeechRequest synthReq =
 new SynthesizeSpeechRequest().withText(text).withVoiceId(voice.getId())
   .withOutputFormat(format).withEngine("neural");
 SynthesizeSpeechResult synthRes = polly.synthesizeSpeech(synthReq);

 return synthRes.getAudioStream();
}

public static void main(String args[]) throws Exception {
 //create the test class
 PollyDemo helloWorld = new PollyDemo(Region.getRegion(Regions.US_EAST_1));
 //get the audio stream
 InputStream speechStream = helloWorld.synthesize(SAMPLE, OutputFormat.Mp3);

 //create an MP3 player
 AdvancedPlayer player = new AdvancedPlayer(speechStream,
   javazoom.jl.player.FactoryRegistry.systemRegistry().createAudioDevice());

 player.setPlayBackListener(new PlaybackListener() {
  @Override
  public void playbackStarted(PlaybackEvent evt) {
   System.out.println("Playback started");
   System.out.println(SAMPLE);
  }

  @Override
  public void playbackFinished(PlaybackEvent evt) {
   System.out.println("Playback finished");
  }
 });


 // play it!
 player.play();

}
```

```
}
```

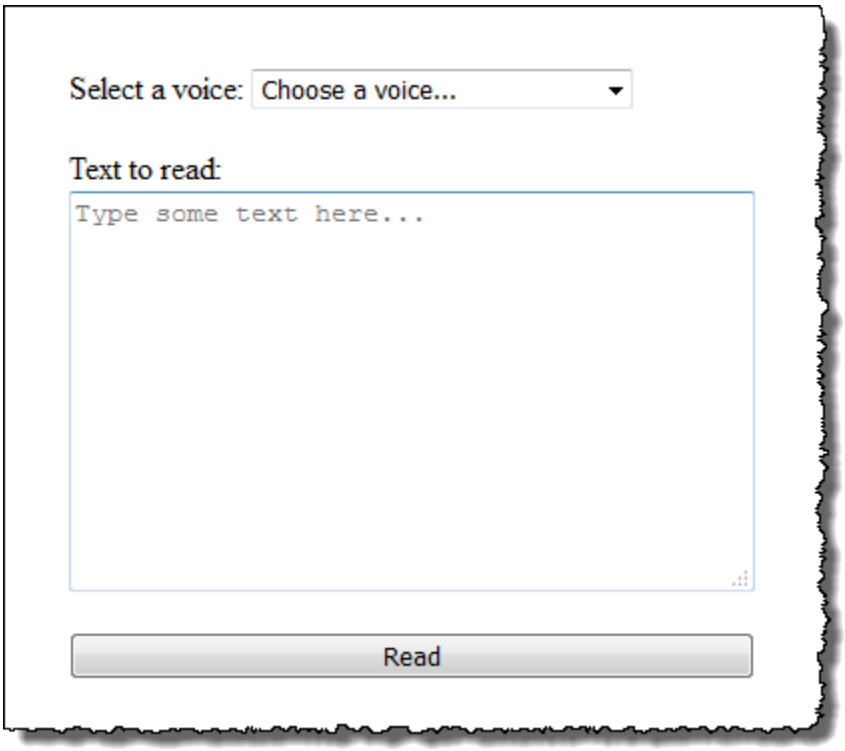
7. Torna alla directory `polly-java-demo` per pulire, compilare ed eseguire la demo:

```
mvn clean compile exec:java
```

Esempio di Python (HTML5 Client e Python Server)

Questa applicazione di esempio è costituita da quanto elencato di seguito:

- Un server HTTP 1.1 che utilizza la codifica di trasferimento in blocchi HTTP (vedi Codifica di trasferimento in blocchi)
- Una semplice interfaccia HTML5 utente che interagisce con il server HTTP 1.1 (mostrato di seguito):



L'obiettivo di questo esempio è mostrare come utilizzare Amazon Polly per lo streaming vocale da un'applicazione basata su browser HTML5 . L'esecuzione del flusso audio prodotto da Amazon Polly durante la sintesi del testo è l'approccio consigliato per i casi d'uso in cui i tempi di risposta

costituiscono un fattore importante (ad esempio, sistemi di dialogo, programmi di lettura dello schermo e così via).

Per eseguire questa applicazione di esempio è necessario quanto elencato di seguito:

- Browser Web conforme agli standard HTML5 e EcmaScript 5 (ad esempio, Chrome 23.0 o versioni successive, Firefox 21.0 o versioni successive, Internet Explorer 9.0 o versioni successive)
- Versione di Python successiva alla 3.0

Per testare l'applicazione

1. Salva il codice server con il nome `server.py`. Per il codice, consulta Esempio in Python: Python Server Code (server.py).
2. Salva il codice client come. HTML5 `index.html` Per il codice, consulta Esempio in Python: interfaccia HTML5 utente (index.html).
3. Esegui il comando riportato di seguito dal percorso in cui hai salvato server.py per avviare l'applicazione (in alcuni sistemi potrebbe essere necessario utilizzare `python3` anziché `python` quando si esegue il comando).

   ```
   $ python  server.py
   ```

   Dopo l'avvio dell'applicazione, un URL viene visualizzato sul terminale.
4. Apri l'URL mostrato nel terminale in un browser Web.

   Puoi passare l'indirizzo e la porta al server dell'applicazione in modo che li utilizzi come parametro per `server.py`. Per ulteriori informazioni, esegui `python server.py -h`.
5. Per ascoltare la sintesi vocale, scegli una voce dall'elenco, digita del testo, quindi scegli Read. La riproduzione della sintesi vocale viene avviata non appena Amazon Polly trasferisce i primi blocchi utilizzabili di dati audio.
6. Per arrestare il server Python al termine del test dell'applicazione, premere Ctrl+C nel terminale in cui il server è in esecuzione.

> **Note**
>
> Il server crea un client Boto3 utilizzando AWS SDK for Python (Boto). Il client utilizza le credenziali memorizzate nel file di AWS configurazione sul tuo computer per firmare e

autenticare le richieste ad Amazon Polly. Per ulteriori informazioni su come creare il file di AWS configurazione e memorizzare le credenziali, consulta la sezione Configurazione nella Guida per l'utente. AWS Command Line InterfaceAWS Command Line Interface

Esempio in Python: interfaccia HTML5 utente (index.html)

Questa sezione fornisce il codice per il HTML5 client descritto inEsempio di Python (HTML5 Client e Python Server).

```
<html>

<head>
    <title>Text-to-Speech Example Application</title>
    <script>
        /*
         * This sample code requires a web browser with support for both the
         * HTML5 and ECMAScript 5 standards; the following is a non-comprehensive
         * list of compliant browsers and their minimum version:
         *
         * - Chrome 23.0+
         * - Firefox 21.0+
         * - Internet Explorer 9.0+
         * - Edge 12.0+
         * - Opera 15.0+
         * - Safari 6.1+
         * - Android (stock web browser) 4.4+
         * - Chrome for Android 51.0+
         * - Firefox for Android 48.0+
         * - Opera Mobile 37.0+
         * - iOS (Safari Mobile and Chrome) 3.2+
         * - Internet Explorer Mobile 10.0+
         * - Blackberry Browser 10.0+
         */

        // Mapping of the OutputFormat parameter of the SynthesizeSpeech API
        // and the audio format strings understood by the browser
        var AUDIO_FORMATS = {
            'ogg_vorbis': 'audio/ogg',
            'mp3': 'audio/mpeg',
            'pcm': 'audio/wave; codecs=1'
        };
```

```
/**
 * Handles fetching JSON over HTTP
 */
function fetchJSON(method, url, onSuccess, onError) {
    var request = new XMLHttpRequest();
    request.open(method, url, true);
    request.onload = function () {
        // If loading is complete
        if (request.readyState === 4) {
            // if the request was successful
            if (request.status === 200) {
                var data;

                // Parse the JSON in the response
                try {
                    data = JSON.parse(request.responseText);
                } catch (error) {
                    onError(request.status, error.toString());
                }

                onSuccess(data);
            } else {
                onError(request.status, request.responseText)
            }
        }
    };

    request.send();
}

/**
 * Returns a list of audio formats supported by the browser
 */
function getSupportedAudioFormats(player) {
    return Object.keys(AUDIO_FORMATS)
        .filter(function (format) {
            var supported = player.canPlayType(AUDIO_FORMATS[format]);
            return supported === 'probably' || supported === 'maybe';
        });
}

// Initialize the application when the DOM is loaded and ready to be
// manipulated
```

```
        document.addEventListener("DOMContentLoaded", function () {
            var input = document.getElementById('input'),
                voiceMenu = document.getElementById('voice'),
                text = document.getElementById('text'),
                player = document.getElementById('player'),
                submit = document.getElementById('submit'),
                supportedFormats = getSupportedAudioFormats(player);

            // Display a message and don't allow submitting the form if the
            // browser doesn't support any of the available audio formats
            if (supportedFormats.length === 0) {
                submit.disabled = true;
                alert('The web browser in use does not support any of the' +
                      ' available audio formats. Please try with a different' +
                      ' one.');
            }

            // Play the audio stream when the form is submitted successfully
            input.addEventListener('submit', function (event) {
                // Validate the fields in the form, display a message if
                // unexpected values are encountered
                if (voiceMenu.selectedIndex <= 0 || text.value.length === 0) {
                    alert('Please fill in all the fields.');
                } else {
                    var selectedVoice = voiceMenu
                                            .options[voiceMenu.selectedIndex]
                                            .value;

                    // Point the player to the streaming server
                    player.src = '/read?voiceId=' +
                        encodeURIComponent(selectedVoice) +
                        '&text=' + encodeURIComponent(text.value) +
                        '&outputFormat=' + supportedFormats[0];
                    player.play();
                }

                // Stop the form from submitting,
                // Submitting the form is allowed only if the browser doesn't
                // support Javascript to ensure functionality in such a case
                event.preventDefault();
            });

            // Load the list of available voices and display them in a menu
            fetchJSON('GET', '/voices',
```

```
                // If the request succeeds
                function (voices) {
                    var container = document.createDocumentFragment();

                    // Build the list of options for the menu
                    voices.forEach(function (voice) {
                        var option = document.createElement('option');
                        option.value = voice['Id'];
                        option.innerHTML = voice['Name'] + ' (' +
                            voice['Gender'] + ', ' +
                            voice['LanguageName'] + ')';
                        container.appendChild(option);
                    });

                    // Add the options to the menu and enable the form field
                    voiceMenu.appendChild(container);
                    voiceMenu.disabled = false;
                },
                // If the request fails
                function (status, response) {
                    // Display a message in case loading data from the server
                    // fails
                    alert(status + ' - ' + response);
                });
        });

    </script>
    <style>
        #input {
            min-width: 100px;
            max-width: 600px;
            margin: 0 auto;
            padding: 50px;
        }

        #input div {
            margin-bottom: 20px;
        }

        #text {
            width: 100%;
            height: 200px;
            display: block;
        }
```

```
        #submit {
            width: 100%;
        }
    </style>
</head>

<body>
    <form id="input" method="GET" action="/read">
        <div>
            <label for="voice">Select a voice:</label>
            <select id="voice" name="voiceId" disabled>
                <option value="">Choose a voice...</option>
            </select>
        </div>
        <div>
            <label for="text">Text to read:</label>
            <textarea id="text" maxlength="1000" minlength="1" name="text"
                    placeholder="Type some text here..."></textarea>
        </div>
        <input type="submit" value="Read" id="submit" />
    </form>
    <audio id="player"></audio>
</body>

</html>
```

Esempio in Python: Python Server Code (server.py)

Questa sezione fornisce il codice per il server di Python descritto in Esempio di Python (HTML5 Client e Python Server).

```
"""
Example Python 2.7+/3.3+ Application

This application consists of a HTTP 1.1 server using the HTTP chunked transfer
coding (https://tools.ietf.org/html/rfc2616#section-3.6.1) and a minimal HTML5
user interface that interacts with it.

The goal of this example is to start streaming the speech to the client (the
HTML5 web UI) as soon as the first consumable chunk of speech is returned in
order to start playing the audio as soon as possible.
For use cases where low latency and responsiveness are strong requirements,
```

```
this is the recommended approach.

The service documentation contains examples for non-streaming use cases where
waiting for the speech synthesis to complete and fetching the whole audio stream
at once are an option.

To test the application, run 'python server.py' and then open the URL
displayed in the terminal in a web browser (see index.html for a list of
supported browsers). The address and port for the server can be passed as
parameters to server.py. For more information, run: 'python server.py -h'
"""
from argparse import ArgumentParser
from collections import namedtuple
from contextlib import closing
from io import BytesIO
from json import dumps as json_encode
import os
import sys

if sys.version_info >= (3, 0):
    from http.server import BaseHTTPRequestHandler, HTTPServer
    from socketserver import ThreadingMixIn
    from urllib.parse import parse_qs
else:
    from BaseHTTPServer import BaseHTTPRequestHandler, HTTPServer
    from SocketServer import ThreadingMixIn
    from urlparse import parse_qs

from boto3 import Session
from botocore.exceptions import BotoCoreError, ClientError

ResponseStatus = namedtuple("HTTPStatus",
                            ["code", "message"])

ResponseData = namedtuple("ResponseData",
                          ["status", "content_type", "data_stream"])

# Mapping the output format used in the client to the content type for the
# response
AUDIO_FORMATS = {"ogg_vorbis": "audio/ogg",
                 "mp3": "audio/mpeg",
                 "pcm": "audio/wave; codecs=1"}
CHUNK_SIZE = 1024
HTTP_STATUS = {"OK": ResponseStatus(code=200, message="OK"),
```

```
                "BAD_REQUEST": ResponseStatus(code=400, message="Bad request"),
                "NOT_FOUND": ResponseStatus(code=404, message="Not found"),
                "INTERNAL_SERVER_ERROR": ResponseStatus(code=500, message="Internal
 server error")}
PROTOCOL = "http"
ROUTE_INDEX = "/index.html"
ROUTE_VOICES = "/voices"
ROUTE_READ = "/read"


# Create a client using the credentials and region defined in the adminuser
# section of the AWS credentials and configuration files
session = Session(profile_name="adminuser")
polly = session.client("polly")


class HTTPStatusError(Exception):
    """Exception wrapping a value from http.server.HTTPStatus"""

    def __init__(self, status, description=None):
        """
        Constructs an error instance from a tuple of
        (code, message, description), see http.server.HTTPStatus
        """
        super(HTTPStatusError, self).__init__()
        self.code = status.code
        self.message = status.message
        self.explain = description


class ThreadedHTTPServer(ThreadingMixIn, HTTPServer):
    """An HTTP Server that handle each request in a new thread"""
    daemon_threads = True


class ChunkedHTTPRequestHandler(BaseHTTPRequestHandler):
    """"HTTP 1.1 Chunked encoding request handler"""
    # Use HTTP 1.1 as 1.0 doesn't support chunked encoding
    protocol_version = "HTTP/1.1"

    def query_get(self, queryData, key, default=""):
        """Helper for getting values from a pre-parsed query string"""
        return queryData.get(key, [default])[0]
```

```
    def do_GET(self):
        """Handles GET requests"""

        # Extract values from the query string
        path, _, query_string = self.path.partition('?')
        query = parse_qs(query_string)

        response = None

        print(u"[START]: Received GET for %s with query: %s" % (path, query))

        try:
            # Handle the possible request paths
            if path == ROUTE_INDEX:
                response = self.route_index(path, query)
            elif path == ROUTE_VOICES:
                response = self.route_voices(path, query)
            elif path == ROUTE_READ:
                response = self.route_read(path, query)
            else:
                response = self.route_not_found(path, query)

            self.send_headers(response.status, response.content_type)
            self.stream_data(response.data_stream)

        except HTTPStatusError as err:
            # Respond with an error and log debug
            # information
            if sys.version_info >= (3, 0):
                self.send_error(err.code, err.message, err.explain)
            else:
                self.send_error(err.code, err.message)

            self.log_error(u"%s %s %s - [%d] %s", self.client_address[0],
                           self.command, self.path, err.code, err.explain)

        print("[END]")

    def route_not_found(self, path, query):
        """Handles routing for unexpected paths"""
        raise HTTPStatusError(HTTP_STATUS["NOT_FOUND"], "Page not found")

    def route_index(self, path, query):
        """Handles routing for the application's entry point'"""
```

```
        try:
            return ResponseData(status=HTTP_STATUS["OK"], content_type="text_html",
                                # Open a binary stream for reading the index
                                # HTML file
                                data_stream=open(os.path.join(sys.path[0],
                                                              path[1:]), "rb"))
        except IOError as err:
            # Couldn't open the stream
            raise HTTPStatusError(HTTP_STATUS["INTERNAL_SERVER_ERROR"],
                                  str(err))

    def route_voices(self, path, query):
        """Handles routing for listing available voices"""
        params = {}
        voices = []

        while True:
            try:
                # Request list of available voices, if a continuation token
                # was returned by the previous call then use it to continue
                # listing
                response = polly.describe_voices(**params)
            except (BotoCoreError, ClientError) as err:
                # The service returned an error
                raise HTTPStatusError(HTTP_STATUS["INTERNAL_SERVER_ERROR"],
                                      str(err))

            # Collect all the voices
            voices.extend(response.get("Voices", []))

            # If a continuation token was returned continue, stop iterating
            # otherwise
            if "NextToken" in response:
                params = {"NextToken": response["NextToken"]}
            else:
                break

        json_data = json_encode(voices)
        bytes_data = bytes(json_data, "utf-8") if sys.version_info >= (3, 0) \
            else bytes(json_data)

        return ResponseData(status=HTTP_STATUS["OK"],
                            content_type="application/json",
                            # Create a binary stream for the JSON data
```

```
                            data_stream=BytesIO(bytes_data))

    def route_read(self, path, query):
        """Handles routing for reading text (speech synthesis)"""
        # Get the parameters from the query string
        text = self.query_get(query, "text")
        voiceId = self.query_get(query, "voiceId")
        outputFormat = self.query_get(query, "outputFormat")

        # Validate the parameters, set error flag in case of unexpected
        # values
        if len(text) == 0 or len(voiceId) == 0 or \
                outputFormat not in AUDIO_FORMATS:
            raise HTTPStatusError(HTTP_STATUS["BAD_REQUEST"],
                                  "Wrong parameters")
        else:
            try:
                # Request speech synthesis
                response = polly.synthesize_speech(Text=text,
                                                   VoiceId=voiceId,
                                                   OutputFormat=outputFormat,
                                                   Engine="neural")
            except (BotoCoreError, ClientError) as err:
                # The service returned an error
                raise HTTPStatusError(HTTP_STATUS["INTERNAL_SERVER_ERROR"],
                                      str(err))

            return ResponseData(status=HTTP_STATUS["OK"],
                                content_type=AUDIO_FORMATS[outputFormat],
                                # Access the audio stream in the response
                                data_stream=response.get("AudioStream"))

    def send_headers(self, status, content_type):
        """Send out the group of headers for a successful request"""
        # Send HTTP headers
        self.send_response(status.code, status.message)
        self.send_header('Content-type', content_type)
        self.send_header('Transfer-Encoding', 'chunked')
        self.send_header('Connection', 'close')
        self.end_headers()

    def stream_data(self, stream):
        """Consumes a stream in chunks to produce the response's output'"""
        print("Streaming started...")
```

```
        if stream:
            # Note: Closing the stream is important as the service throttles on
            # the number of parallel connections. Here we are using
            # contextlib.closing to ensure the close method of the stream object
            # will be called automatically at the end of the with statement's
            # scope.
            with closing(stream) as managed_stream:
                # Push out the stream's content in chunks
                while True:
                    data = managed_stream.read(CHUNK_SIZE)
                    self.wfile.write(b"%X\r\n%s\r\n" % (len(data), data))

                    # If there's no more data to read, stop streaming
                    if not data:
                        break

                # Ensure any buffered output has been transmitted and close the
                # stream
                self.wfile.flush()

            print("Streaming completed.")
        else:
            # The stream passed in is empty
            self.wfile.write(b"0\r\n\r\n")
            print("Nothing to stream.")

# Define and parse the command line arguments
cli = ArgumentParser(description='Example Python Application')
cli.add_argument(
    "-p", "--port", type=int, metavar="PORT", dest="port", default=8000)
cli.add_argument(
    "--host", type=str, metavar="HOST", dest="host", default="localhost")
arguments = cli.parse_args()

# If the module is invoked directly, initialize the application
if __name__ == '__main__':
    # Create and configure the HTTP server instance
    server = ThreadedHTTPServer((arguments.host, arguments.port),
                                ChunkedHTTPRequestHandler)
    print("Starting server, use <Ctrl-C> to stop...")
    print(u"Open {0}://{1}:{2}{3} in a web browser.".format(PROTOCOL,
                                                            arguments.host,
                                                            arguments.port,
```

```
                                                                ROUTE_INDEX))

    try:
        # Listen for requests indefinitely
        server.serve_forever()
    except KeyboardInterrupt:
        # A request to terminate has been received, stop the server
        print("\nShutting down...")
        server.socket.close()
```

Esempio iOS

Nel seguente esempio viene utilizzato l'SDK di iOS per consentire a Amazon Polly di leggere il testo specificato utilizzando una voce selezionata da un elenco di voci.

Il codice mostrato di seguito copre i principali processi, ma non gestisce gli errori. Per il codice completo, consulta AWS Mobile SDK for iOS Amazon Polly demo.

Inizializzazione

```
// 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()
```

Visualizza elenco di voci disponibili

```
// 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
})
```

Sintesi vocale

```
// 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
})
```

Esempio Android

Nel seguente esempio viene utilizzato l'SDK di Android per consentire a Amazon Polly di leggere il testo specificato utilizzando una voce selezionata da un elenco di voci.

Il codice mostrato di seguito copre i principali processi, ma non gestisce gli errori. Per il codice completo, consulta AWS Mobile SDK for Android Amazon Polly demo.

Inizializzazione

```
// 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);
```

Visualizza elenco di voci disponibili

```
// 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();
```

Visualizzazione dell'URL per lo streaming audio

```
// 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);
```

Riproduzione dell'audio sintetizzato

```
// 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();
    }
});
```

Esempi di codice per l'utilizzo di Amazon Polly AWS SDKs

I seguenti esempi di codice mostrano come usare Amazon Polly con un kit di sviluppo AWS software (SDK).

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le operazioni mostrino come richiamare le singole funzioni del servizio, è possibile visualizzarle contestualizzate negli scenari correlati.

Gli scenari sono esempi di codice che mostrano come eseguire un'attività specifica richiamando più funzioni all'interno dello stesso servizio o combinate con altri Servizi AWS.

Per un elenco completo di guide ed esempi di codice per sviluppatori AWS SDK, consulta. Utilizzo di Amazon Polly con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Esempi di codice

- Esempi di base per l'utilizzo di Amazon Polly AWS SDKs
  - Azioni per l'utilizzo di Amazon Polly AWS SDKs
    - Utilizzo DeleteLexicon con un AWS SDK o una CLI
    - Utilizzare DescribeVoices con un SDK AWS
    - Utilizzo GetLexicon con un AWS SDK o una CLI
    - Utilizzo GetSpeechSynthesisTask con un AWS SDK o una CLI
    - Utilizzo ListLexicons con un AWS SDK o una CLI
    - Utilizzo PutLexicon con un AWS SDK o una CLI
    - Utilizzo StartSpeechSynthesisTask con un AWS SDK o una CLI
    - Utilizzare SynthesizeSpeech con un SDK AWS
- Scenari per l'utilizzo di Amazon Polly AWS SDKs
  - Converti testo in voce e viceversa utilizzando un SDK AWS
  - Crea un'applicazione di sincronizzazione labiale con Amazon Polly utilizzando un SDK AWS
  - Crea un'applicazione che analizza il feedback dei clienti e sintetizza l'audio

Esempi di base per l'utilizzo di Amazon Polly AWS SDKs

I seguenti esempi di codice mostrano come utilizzare le nozioni di base di Amazon AWS SDKs Polly con.

Esempi

- Azioni per l'utilizzo di Amazon Polly AWS SDKs
  - Utilizzo DeleteLexicon con un AWS SDK o una CLI
  - Utilizzare DescribeVoices con un SDK AWS
  - Utilizzo GetLexicon con un AWS SDK o una CLI
  - Utilizzo GetSpeechSynthesisTask con un AWS SDK o una CLI
  - Utilizzo ListLexicons con un AWS SDK o una CLI
  - Utilizzo PutLexicon con un AWS SDK o una CLI
  - Utilizzo StartSpeechSynthesisTask con un AWS SDK o una CLI
  - Utilizzare SynthesizeSpeech con un SDK AWS

Azioni per l'utilizzo di Amazon Polly AWS SDKs

I seguenti esempi di codice mostrano come eseguire singole azioni di Amazon Polly con. AWS SDKs Ogni esempio include un collegamento a GitHub, dove puoi trovare le istruzioni per la configurazione e l'esecuzione del codice.

Questi estratti chiamano l'API Amazon Polly e sono estratti di codice da programmi più grandi che devono essere eseguiti nel contesto. È possibile visualizzare le azioni nel contesto in Scenari per l'utilizzo di Amazon Polly AWS SDKs .

Gli esempi seguenti includono solo le operazioni più comunemente utilizzate. Per un elenco completo, consulta l'Amazon Polly API Reference.

Esempi

- Utilizzo DeleteLexicon con un AWS SDK o una CLI
- Utilizzare DescribeVoices con un SDK AWS
- Utilizzo GetLexicon con un AWS SDK o una CLI
- Utilizzo GetSpeechSynthesisTask con un AWS SDK o una CLI
- Utilizzo ListLexicons con un AWS SDK o una CLI

- Utilizzo PutLexicon con un AWS SDK o una CLI
- Utilizzo StartSpeechSynthesisTask con un AWS SDK o una CLI
- Utilizzare SynthesizeSpeech con un SDK AWS

Utilizzo **`DeleteLexicon`** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `DeleteLexicon`.

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
        }
    }
```

- Per i dettagli sull'API, consulta la DeleteLexiconsezione AWS SDK per .NET API Reference.

CLI

AWS CLI

Per eliminare un lessico

L'`delete-lexicon`esempio seguente elimina il lessico specificato.

```
aws polly delete-lexicon \
```

```
    --name w3c
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Utilizzo dell' DeleteLexicon operazione nella Amazon Polly Developer Guide.

- Per i dettagli sull'API, consulta DeleteLexicon AWS CLICommand Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di Amazon Polly con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`DescribeVoices`** con un SDK AWS

Gli esempi di codice seguenti mostrano come utilizzare `DescribeVoices`.

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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}");
        }
    }
```

- Per i dettagli sull'API, consulta la DescribeVoicessezione AWS SDK per .NET API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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(voicesRequest);
            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);
        }
    }
}
```

- Per i dettagli sull'API, consulta la DescribeVoicessezione AWS SDK for Java 2.x API Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta DescribeVoices AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta la DescribeVoicessezione AWS SDK per Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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(())
}
```

- Per i dettagli sulle API, consulta la DescribeVoicesguida di riferimento all'API AWS SDK for Rust.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di Amazon Polly con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **GetLexicon** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `GetLexicon`.

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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);
            }
        }
    }
```

- Per i dettagli sull'API, consulta la GetLexiconsezione AWS SDK per .NET API Reference.

CLI

AWS CLI

Per recuperare il contenuto di un lessico

L'`get-lexicon`esempio seguente recupera il contenuto del lessico di pronuncia specificato.

```
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
    }
}
```

Per ulteriori informazioni, consulta Utilizzo dell' GetLexicon operazione nella Amazon Polly Developer Guide.

- Per i dettagli sull'API, consulta GetLexicon AWS CLICommand Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta GetLexicon AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di Amazon Polly con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetSpeechSynthesisTask`** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `GetSpeechSynthesisTask`.

CLI

AWS CLI

Per ottenere informazioni su un'attività di sintesi vocale

L'`get-speech-synthesis-task`esempio seguente recupera informazioni sull'attività di sintesi vocale specificata.

```
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"
    }
}
```

Per ulteriori informazioni, consulta Creazione di file audio lunghi nella Amazon Polly Developer Guide.

- Per i dettagli sull'API, consulta GetSpeechSynthesisTask AWS CLICommand Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta GetSpeechSynthesisTask AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di Amazon Polly con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListLexicons`** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `ListLexicons`.

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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}");
                        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}");
            }
        }
    }
```

- Per i dettagli sull'API, consulta la ListLexiconssezione AWS SDK per .NET API Reference.

CLI

AWS CLI

Per elencare i tuoi lessici

L'`list-lexicons`esempio seguente elenca i tuoi lessici di pronuncia.

```
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
            }
        }
    ]
}
```

Per ulteriori informazioni, consulta Utilizzo dell' ListLexicons operazione nella Amazon Polly Developer Guide.

- Per i dettagli sull'API, consulta ListLexicons AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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);
        }
    }
}
```

- Per i dettagli sull'API, consulta la ListLexiconssezione AWS SDK for Java 2.x API Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta ListLexicons AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta la ListLexiconssezione AWS SDK per Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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(())
}
```

- Per i dettagli sulle API, consulta la ListLexiconsguida di riferimento all'API AWS SDK for Rust.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di Amazon Polly con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`PutLexicon`** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `PutLexicon`.

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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);
            }
        }
    }
```

- Per i dettagli sull'API, consulta la PutLexiconsezione AWS SDK per .NET API Reference.

CLI

AWS CLI

Per memorizzare un lessico

L'`put-lexicon`esempio seguente memorizza il lessico di pronuncia specificato. Il `example.pls` file specifica un lessico conforme al W3C PLS.

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

Contenuto di `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>
}
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Utilizzo dell' PutLexicon operazione nella Amazon Polly Developer Guide.

- Per i dettagli sull'API, consulta PutLexicon AWS CLICommand Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta PutLexicon AWSSDK for Python (Boto3) API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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(())
}
```

- Per i dettagli sulle API, consulta la PutLexiconguida di riferimento all'API AWS SDK for Rust.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di Amazon Polly con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`StartSpeechSynthesisTask`** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `StartSpeechSynthesisTask`.

CLI

AWS CLI

Per sintetizzare il testo

L'`start-speech-synthesis-task`esempio seguente sintetizza il testo `text_file.txt` e memorizza il MP3 file risultante nel bucket specificato.

```
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"
    }
}
```

Per ulteriori informazioni, consulta Creazione di file audio lunghi nella Amazon Polly Developer Guide.

- Per i dettagli sull'API, consulta StartSpeechSynthesisTask AWS CLICommand Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta StartSpeechSynthesisTask AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di Amazon Polly con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`SynthesizeSpeech`** con un SDK AWS

Gli esempi di codice seguenti mostrano come utilizzare `SynthesizeSpeech`.

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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.");
        }
    }
```

Sintetizza il parlato dal testo utilizzando i segni vocali con Amazon Polly utilizzando un SDK. AWS

```
    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}");
            }
        }
    }
```

- Per i dettagli sulle API, consulta SynthesizeSpeechla sezione API Reference.AWS SDK per .NET

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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;
    }
}
```

- Per i dettagli sull'API, consulta la SynthesizeSpeechsezione AWS SDK for Java 2.x API Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta SynthesizeSpeech AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta la SynthesizeSpeechsezione AWS SDK per Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice 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(())
}
```

- Per i dettagli sulle API, consulta la SynthesizeSpeechguida di riferimento all'API AWS SDK for Rust.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di Amazon Polly con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Scenari per l'utilizzo di Amazon Polly AWS SDKs

I seguenti esempi di codice mostrano come implementare scenari comuni in Amazon Polly con. AWS SDKs Questi scenari mostrano come eseguire attività specifiche richiamando più funzioni all'interno di Amazon Polly o combinandole con altre. Servizi AWS Ogni scenario include un collegamento al codice sorgente completo, dove è possibile trovare le istruzioni su come configurare ed eseguire il codice.

Gli scenari sono relativi a un livello intermedio di esperienza per aiutarti a comprendere le azioni di servizio nel contesto.

Esempi

- Converti testo in voce e viceversa utilizzando un SDK AWS
- Crea un'applicazione di sincronizzazione labiale con Amazon Polly utilizzando un SDK AWS
- Crea un'applicazione che analizza il feedback dei clienti e sintetizza l'audio

Converti testo in voce e viceversa utilizzando un SDK AWS

L'esempio di codice seguente mostra come:

- Utilizzare Amazon Polly per sintetizzare un file di input in testo normale (UTF-8) in un file audio.
- Carica il file audio in un bucket Amazon S3.
- Utilizzare Amazon Transcribe per convertire il file audio in testo.
- Visualizzare il testo.

Rust

SDK per Rust

Utilizza Amazon Polly per sintetizzare un file di input di testo normale (UTF-8) in un file audio, caricare il file audio in un bucket Amazon S3, utilizzare Amazon Transcribe per convertire il file audio in testo e visualizzare il testo.

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, consulta l'esempio completo su GitHub.

Servizi utilizzati in questo esempio

- Amazon Polly
- Amazon S3
- Amazon Transcribe

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di Amazon Polly con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Crea un'applicazione di sincronizzazione labiale con Amazon Polly utilizzando un SDK AWS

Il seguente esempio di codice mostra come creare un'applicazione di sincronizzazione labiale con Amazon Polly.

Python

SDK per Python (Boto3)

Mostra come usare Amazon Polly e Tkinter per creare un'applicazione di sincronizzazione labiale che mostra un volto animato che parla insieme al parlato sintetizzato da Amazon Polly. La sincronizzazione labiale viene eseguita richiedendo ad Amazon Polly un elenco di visemi corrispondenti al parlato sintetizzato.

- Ottieni metadati vocali da Amazon Polly e visualizzali in un'applicazione Tkinter.
- Ottieni audio vocale sintetizzato e indicatori vocali viseme corrispondenti da Amazon Polly.
- Riproduci l'audio con movimenti della bocca sincronizzati su un volto animato.
- Invia attività di sintesi asincrone per testi lunghi e recupera l'output da un bucket Amazon Simple Storage Service (Amazon S3).

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, consulta l'esempio completo su. GitHub

Servizi utilizzati in questo esempio

- Amazon Polly

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di Amazon Polly con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Crea un'applicazione che analizza il feedback dei clienti e sintetizza l'audio

Il seguente esempio di codice spiega come creare un'applicazione che analizza schede dei commenti dei clienti, le traduce dalla loro lingua originale, ne determina la valutazione e genera un file audio dal testo tradotto.

.NET

SDK per .NET

Questa applicazione di esempio analizza e archivia le schede di feedback dei clienti. In particolare, soddisfa l'esigenza di un hotel fittizio a New York City. L'hotel riceve feedback dagli ospiti in varie lingue sotto forma di schede di commento fisiche. Tale feedback viene caricato nell'app tramite un client Web. Dopo aver caricato l'immagine di una scheda di commento, vengono eseguiti i seguenti passaggi:

- Il testo viene estratto dall'immagine utilizzando Amazon Textract.
- Amazon Comprehend determina il sentiment del testo estratto e la sua lingua.
- Il testo estratto viene tradotto in inglese utilizzando Amazon Translate.
- Amazon Polly sintetizza un file audio dal testo estratto.

L'app completa può essere implementata con AWS CDK. Per il codice sorgente e le istruzioni di distribuzione, consulta il progetto in GitHub.

Servizi utilizzati in questo esempio

- Amazon Comprehend
- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

Java

SDK per Java 2.x

Questa applicazione di esempio analizza e archivia le schede di feedback dei clienti. In particolare, soddisfa l'esigenza di un hotel fittizio a New York City. L'hotel riceve feedback dagli ospiti in varie lingue sotto forma di schede di commento fisiche. Tale feedback viene caricato nell'app tramite un client Web. Dopo aver caricato l'immagine di una scheda di commento, vengono eseguiti i seguenti passaggi:

- Il testo viene estratto dall'immagine utilizzando Amazon Textract.
- Amazon Comprehend determina il sentiment del testo estratto e la sua lingua.
- Il testo estratto viene tradotto in inglese utilizzando Amazon Translate.
- Amazon Polly sintetizza un file audio dal testo estratto.

L'app completa può essere implementata con AWS CDK. Per il codice sorgente e le istruzioni di distribuzione, consulta il progetto in GitHub.

Servizi utilizzati in questo esempio

- Amazon Comprehend
- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

JavaScript

SDK per JavaScript (v3)

Questa applicazione di esempio analizza e archivia le schede di feedback dei clienti. In particolare, soddisfa l'esigenza di un hotel fittizio a New York City. L'hotel riceve feedback dagli ospiti in varie lingue sotto forma di schede di commento fisiche. Tale feedback viene caricato nell'app tramite un client Web. Dopo aver caricato l'immagine di una scheda di commento, vengono eseguiti i seguenti passaggi:

- Il testo viene estratto dall'immagine utilizzando Amazon Textract.
- Amazon Comprehend determina il sentiment del testo estratto e la sua lingua.

- Il testo estratto viene tradotto in inglese utilizzando Amazon Translate.
- Amazon Polly sintetizza un file audio dal testo estratto.

L'app completa può essere implementata con AWS CDK. Per il codice sorgente e le istruzioni di distribuzione, consulta il progetto in. GitHub I seguenti estratti mostrano come AWS SDK per JavaScript viene utilizzato all'interno delle funzioni 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 };
};
```

Servizi utilizzati in questo esempio

- Amazon Comprehend
- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

Ruby

SDK per Ruby

Questa applicazione di esempio analizza e archivia le schede di feedback dei clienti. In particolare, soddisfa l'esigenza di un hotel fittizio a New York City. L'hotel riceve feedback dagli ospiti in varie lingue sotto forma di schede di commento fisiche. Tale feedback viene caricato nell'app tramite un client Web. Dopo aver caricato l'immagine di una scheda di commento, vengono eseguiti i seguenti passaggi:

- Il testo viene estratto dall'immagine utilizzando Amazon Textract.
- Amazon Comprehend determina il sentiment del testo estratto e la sua lingua.
- Il testo estratto viene tradotto in inglese utilizzando Amazon Translate.

- Amazon Polly sintetizza un file audio dal testo estratto.

L'app completa può essere implementata con AWS CDK. Per il codice sorgente e le istruzioni di distribuzione, consulta il progetto in. GitHub

Servizi utilizzati in questo esempio

- Amazon Comprehend
- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di Amazon Polly con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Sicurezza in Amazon Polly

La sicurezza del cloud AWS è la massima priorità. In qualità di AWS cliente, puoi beneficiare di un data center e di un'architettura di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra AWS te e te. Il modello di responsabilità condivisa descrive questo aspetto come sicurezza del cloud e sicurezza nel cloud:

- Sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi nel AWS cloud. AWS ti fornisce anche servizi che puoi utilizzare in modo sicuro. I revisori esterni testano e verificano regolarmente l'efficacia della nostra sicurezza nell'ambito dei AWS Programmi di AWS conformità dei Programmi di conformità dei di . Per ulteriori informazioni sui programmi di conformità che si applicano ad Amazon Polly, consulta AWS Services in Scope by Compliance Program Program.
- Sicurezza nel cloud: la tua responsabilità è determinata dal AWS servizio che utilizzi. L'utente è anche responsabile di altri fattori, tra cui la riservatezza dei dati, i requisiti dell'azienda e le leggi e le normative applicabili.

Questa documentazione aiuta a comprendere come applicare il modello di responsabilità condivisa quando si utilizza Amazon Polly. Gli argomenti seguenti descrivono come configurare Amazon Polly per soddisfare gli obiettivi di sicurezza e conformità. Scopri anche come utilizzare altri AWS servizi che ti aiutano a monitorare e proteggere le tue risorse Amazon Polly.

Argomenti

- Protezione dei dati in Amazon Polly
- Identity and Access Management in Amazon Polly
- Registrazione e monitoraggio in Amazon Polly
- Convalida della conformità per Amazon Polly
- Resilienza in Amazon Polly
- Sicurezza dell'infrastruttura in Amazon Polly
- Best practice di sicurezza per Amazon Polly
- Uso di Amazon Polly con endpoint VPC dell'interfaccia

Protezione dei dati in Amazon Polly

Amazon Polly è conforme al modello di responsabilità AWS condivisa, che include regolamenti e linee guida per la protezione dei dati. AWS è responsabile della protezione dell'infrastruttura globale che gestisce tutti i AWS servizi. AWS mantiene il controllo sui dati ospitati su questa infrastruttura, compresi i controlli di configurazione di sicurezza per la gestione dei contenuti e dei dati personali dei clienti. AWS i clienti e i partner APN, che agiscono in qualità di titolari o incaricati del trattamento dei dati, sono responsabili di tutti i dati personali che inseriscono nel AWS Cloud.

Ai fini della protezione dei dati, ti consigliamo di proteggere le credenziali AWS dell'account e di configurare i singoli utenti con AWS Identity and Access Management (IAM), in modo che a ciascun utente vengano concesse solo le autorizzazioni necessarie per svolgere le proprie mansioni lavorative. Ti suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- SSL/TLS Da utilizzare per comunicare con AWS le risorse.
- Configura l'API e la registrazione delle attività degli utenti con AWS CloudTrail.
- Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno AWS dei servizi.

Consigliamo di non inserire mai informazioni identificative sensibili, ad esempio i numeri di account dei clienti, in campi a formato libero come un campo Nome. Ciò include quando lavori con Amazon Polly o altri AWS servizi utilizzando la console, l'API o. AWS CLI AWS SDKs Gli eventuali dati immessi in Amazon Polly o altri servizi potrebbero essere prelevati per l'inserimento nei log di diagnostica. Quando fornisci un URL a un server esterno, non includere informazioni sulle credenziali nell'URL per convalidare la tua richiesta a tale server.

Per ulteriori informazioni sulla protezione dei dati, consulta il post del blog AWS Modello di responsabilità condivisa e GDPR su AWS Security Blog.

Crittografia dei dati inattivi

L'output della sintesi vocale Amazon Polly può essere salvato sul tuo sistema. Puoi anche chiamare Amazon Polly, e quindi crittografare il file con qualsiasi chiave di crittografia di tua scelta e archiviarlo in Amazon Simple Storage Service (Amazon S3) o in un altro archivio sicuro. L'operazione the section called “SynthesizeSpeech” Amazon Polly è senza stato e non è associata a un'identità del cliente. Non puoi recuperarla successivamente da Amazon Polly.

Crittografia in transito

Tutti gli invii di testo sono protetti da TLS durante il transito. Amazon Polly non conserva il contenuto degli invii di testo.

Riservatezza del traffico Internet

L'accesso ad Amazon Polly avviene tramite AWS console, CLI o. SDKs Le comunicazioni utilizzano la crittografia delle sessioni TLS (Transport Layer Security) per la riservatezza e le firme digitali per l'autenticazione e l'integrità.

Identity and Access Management in Amazon Polly

AWS Identity and Access Management (IAM) è uno strumento Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle risorse. AWS Gli amministratori IAM controllano chi è autenticato (accesso effettuato) e autorizzato (dispone di autorizzazioni) a utilizzare risorse Amazon Polly. IAM è un software Servizio AWS che puoi utilizzare senza costi aggiuntivi.

Argomenti

- Destinatari
- Autenticazione con identità
- Gestione dell'accesso con policy
- Come Amazon Polly funziona con IAM
- Esempi di policy basate su identità Amazon Polly
- Autorizzazioni API Amazon Polly: informazioni di riferimento su operazioni, autorizzazioni e risorse
- Risoluzione dei problemi relativi all'identità e all'accesso di Amazon Polly

Destinatari

Il modo in cui utilizzi AWS Identity and Access Management (IAM) varia a seconda del lavoro svolto in Amazon Polly.

Utente del servizio: se utilizzi il servizio Amazon Polly per eseguire il tuo lavoro, l'amministratore fornisce le credenziali e le autorizzazioni necessarie. All'aumentare del numero di caratteristiche Amazon Polly utilizzate per il lavoro, potrebbero essere necessarie ulteriori autorizzazioni. La

comprensione della gestione dell'accesso ti consente di richiedere le autorizzazioni corrette all'amministratore. Se non riesci ad accedere a una funzionalità in Amazon Polly, consulta Risoluzione dei problemi relativi all'identità e all'accesso di Amazon Polly.

Service administrator (Amministratore del servizio): se sei il responsabile delle risorse Amazon Polly presso la tua azienda, probabilmente disponi dell'accesso completo ad Amazon Polly. Il compito dell'utente è determinare le caratteristiche e le risorse di Amazon SNS a cui gli utenti del servizio devono accedere. Devi inviare le richieste all'amministratore IAM per cambiare le autorizzazioni degli utenti del servizio. Esamina le informazioni contenute in questa pagina per comprendere i concetti di base relativi a IAM. Per ulteriori informazioni su come la tua azienda può utilizzare IAM con Amazon Polly, consulta Come Amazon Polly funziona con IAM.

Amministratore IAM: se sei un amministratore IAM, potresti essere interessato a ottenere informazioni su come scrivere policy per gestire l'accesso ad Amazon Polly. Per visualizzare policy basate su identità di esempio di Amazon Polly che possono essere utilizzate in IAM, consulta Esempi di policy basate su identità Amazon Polly.

Autenticazione con identità

L'autenticazione è il modo in cui accedi AWS utilizzando le tue credenziali di identità. Devi essere autenticato (aver effettuato l' Utente root dell'account AWS accesso AWS) come utente IAM o assumendo un ruolo IAM.

Puoi accedere AWS come identità federata utilizzando le credenziali fornite tramite una fonte di identità. AWS IAM Identity Center Gli utenti (IAM Identity Center), l'autenticazione Single Sign-On della tua azienda e le tue credenziali di Google o Facebook sono esempi di identità federate. Se accedi come identità federata, l'amministratore ha configurato in precedenza la federazione delle identità utilizzando i ruoli IAM. Quando accedi AWS utilizzando la federazione, assumi indirettamente un ruolo.

A seconda del tipo di utente, puoi accedere al AWS Management Console o al portale di AWS accesso. Per ulteriori informazioni sull'accesso a AWS, vedi Come accedere al tuo Account AWS nella Guida per l'Accedi ad AWS utente.

Se accedi a AWS livello di codice, AWS fornisce un kit di sviluppo software (SDK) e un'interfaccia a riga di comando (CLI) per firmare crittograficamente le tue richieste utilizzando le tue credenziali. Se non utilizzi AWS strumenti, devi firmare tu stesso le richieste. Per ulteriori informazioni sul metodo consigliato per la firma delle richieste, consulta Signature Version 4 AWS per le richieste API nella Guida per l'utente IAM.

A prescindere dal metodo di autenticazione utilizzato, potrebbe essere necessario specificare ulteriori informazioni sulla sicurezza. Ad esempio, ti AWS consiglia di utilizzare l'autenticazione a più fattori (MFA) per aumentare la sicurezza del tuo account. Per ulteriori informazioni, consulta Autenticazione a più fattori nella Guida per l'utente di AWS IAM Identity Center e Utilizzo dell'autenticazione a più fattori (MFA)AWS in IAM nella Guida per l'utente IAM.

Account AWS utente root

Quando si crea un account Account AWS, si inizia con un'identità di accesso che ha accesso completo a tutte Servizi AWS le risorse dell'account. Questa identità è denominata utente Account AWS root ed è accessibile effettuando l'accesso con l'indirizzo e-mail e la password utilizzati per creare l'account. Si consiglia vivamente di non utilizzare l'utente root per le attività quotidiane. Conserva le credenziali dell'utente root e utilizzale per eseguire le operazioni che solo l'utente root può eseguire. Per un elenco completo delle attività che richiedono l'accesso come utente root, consulta la sezione Attività che richiedono le credenziali dell'utente root nella Guida per l'utente IAM.

Identità federata

Come procedura consigliata, richiedi agli utenti umani, compresi gli utenti che richiedono l'accesso come amministratore, di utilizzare la federazione con un provider di identità per accedere Servizi AWS utilizzando credenziali temporanee.

Un'identità federata è un utente dell'elenco utenti aziendale, di un provider di identità Web AWS Directory Service, della directory Identity Center o di qualsiasi utente che accede utilizzando le Servizi AWS credenziali fornite tramite un'origine di identità. Quando le identità federate accedono Account AWS, assumono ruoli e i ruoli forniscono credenziali temporanee.

Per la gestione centralizzata degli accessi, consigliamo di utilizzare AWS IAM Identity Center. Puoi creare utenti e gruppi in IAM Identity Center oppure puoi connetterti e sincronizzarti con un set di utenti e gruppi nella tua fonte di identità per utilizzarli su tutte le tue applicazioni. Account AWS Per ulteriori informazioni su IAM Identity Center, consulta Cos'è IAM Identity Center? nella Guida per l'utente di AWS IAM Identity Center .

Utenti e gruppi IAM

Un utente IAM è un'identità interna Account AWS che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ove possibile, consigliamo di fare affidamento a credenziali temporanee invece di creare utenti IAM con credenziali a lungo termine come le password e le chiavi di accesso. Tuttavia, se si hanno casi d'uso specifici che richiedono credenziali a lungo termine con

utenti IAM, si consiglia di ruotare le chiavi di accesso. Per ulteriori informazioni, consulta la pagina Rotazione periodica delle chiavi di accesso per casi d'uso che richiedono credenziali a lungo termine nella Guida per l'utente IAM.

Un gruppo IAM è un'identità che specifica un insieme di utenti IAM. Non è possibile eseguire l'accesso come gruppo. È possibile utilizzare gruppi per specificare le autorizzazioni per più utenti alla volta. I gruppi semplificano la gestione delle autorizzazioni per set di utenti di grandi dimensioni. Ad esempio, potresti avere un gruppo denominato IAMAdminse concedere a quel gruppo le autorizzazioni per amministrare le risorse IAM.

Gli utenti sono diversi dai ruoli. Un utente è associato in modo univoco a una persona o un'applicazione, mentre un ruolo è destinato a essere assunto da chiunque ne abbia bisogno. Gli utenti dispongono di credenziali a lungo termine permanenti, mentre i ruoli forniscono credenziali temporanee. Per ulteriori informazioni, consulta Casi d'uso per utenti IAM nella Guida per l'utente IAM.

Ruoli IAM

Un ruolo IAM è un'identità interna all'utente Account AWS che dispone di autorizzazioni specifiche. È simile a un utente IAM, ma non è associato a una persona specifica. Per assumere temporaneamente un ruolo IAM in AWS Management Console, puoi passare da un ruolo utente a un ruolo IAM (console). Puoi assumere un ruolo chiamando un'operazione AWS CLI o AWS API o utilizzando un URL personalizzato. Per ulteriori informazioni sui metodi per l'utilizzo dei ruoli, consulta Utilizzo di ruoli IAM nella Guida per l'utente IAM.

I ruoli IAM con credenziali temporanee sono utili nelle seguenti situazioni:

- Accesso utente federato: per assegnare le autorizzazioni a una identità federata, è possibile creare un ruolo e definire le autorizzazioni per il ruolo. Quando un'identità federata viene autenticata, l'identità viene associata al ruolo e ottiene le autorizzazioni da esso definite. Per ulteriori informazioni sulla federazione dei ruoli, consulta Create a role for a third-party identity provider (federation) nella Guida per l'utente IAM. Se utilizzi IAM Identity Center, configura un set di autorizzazioni. IAM Identity Center mette in correlazione il set di autorizzazioni con un ruolo in IAM per controllare a cosa possono accedere le identità dopo l'autenticazione. Per informazioni sui set di autorizzazioni, consulta Set di autorizzazioni nella Guida per l'utente di AWS IAM Identity Center .
- Autorizzazioni utente IAM temporanee: un utente IAM o un ruolo può assumere un ruolo IAM per ottenere temporaneamente autorizzazioni diverse per un'attività specifica.

- Accesso multi-account: è possibile utilizzare un ruolo IAM per permettere a un utente (un principale affidabile) con un account diverso di accedere alle risorse nell'account. I ruoli sono lo strumento principale per concedere l'accesso multi-account. Tuttavia, con alcuni Servizi AWS, è possibile allegare una policy direttamente a una risorsa (anziché utilizzare un ruolo come proxy). Per informazioni sulle differenze tra ruoli e policy basate su risorse per l'accesso multi-account, consulta Accesso a risorse multi-account in IAM nella Guida per l'utente IAM.
- Accesso a più servizi: alcuni Servizi AWS utilizzano le funzionalità di altri Servizi AWS. Ad esempio, quando effettui una chiamata in un servizio, è normale che quel servizio esegua applicazioni in Amazon EC2 o archivi oggetti in Amazon S3. Un servizio può eseguire questa operazione utilizzando le autorizzazioni dell'entità chiamante, utilizzando un ruolo di servizio o utilizzando un ruolo collegato al servizio.
  - Sessioni di accesso inoltrato (FAS): quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama an Servizio AWS, combinate con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli delle policy relative alle richieste FAS, consulta Forward access sessions.
  - Ruolo di servizio: un ruolo di servizio è un ruolo IAM che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione Create a role to delegate permissions to an Servizio AWS nella Guida per l'utente IAM.
  - Ruolo collegato al servizio: un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.
- Applicazioni in esecuzione su Amazon EC2: puoi utilizzare un ruolo IAM per gestire le credenziali temporanee per le applicazioni in esecuzione su un' EC2 istanza e che AWS CLI effettuano richieste AWS API. È preferibile archiviare le chiavi di accesso all'interno dell' EC2 istanza. Per assegnare un AWS ruolo a un' EC2 istanza e renderlo disponibile per tutte le sue applicazioni, create un profilo di istanza collegato all'istanza. Un profilo di istanza contiene il ruolo e consente ai programmi in esecuzione sull' EC2 istanza di ottenere credenziali temporanee. Per ulteriori

informazioni, consulta Utilizzare un ruolo IAM per concedere le autorizzazioni alle applicazioni in esecuzione su EC2 istanze Amazon nella IAM User Guide.

Gestione dell'accesso con policy

Puoi controllare l'accesso AWS creando policy e collegandole a AWS identità o risorse. Una policy è un oggetto AWS che, se associato a un'identità o a una risorsa, ne definisce le autorizzazioni. AWS valuta queste politiche quando un principale (utente, utente root o sessione di ruolo) effettua una richiesta. Le autorizzazioni nelle policy determinano l'approvazione o il rifiuto della richiesta. La maggior parte delle politiche viene archiviata AWS come documenti JSON. Per ulteriori informazioni sulla struttura e sui contenuti dei documenti delle policy JSON, consulta Panoramica delle policy JSON nella Guida per l'utente IAM.

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Per concedere agli utenti l'autorizzazione a eseguire operazioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM. L'amministratore può quindi aggiungere le policy IAM ai ruoli e gli utenti possono assumere i ruoli.

Le policy IAM definiscono le autorizzazioni relative a un'operazione, a prescindere dal metodo utilizzato per eseguirla. Ad esempio, supponiamo di disporre di una policy che consente l'operazione `iam:GetRole`. Un utente con tale policy può ottenere informazioni sul ruolo dall' AWS Management Console AWS CLI, dall'o dall' AWS API.

Policy basate sull'identità

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le operazioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente nella Guida per l'utente IAM.

Le policy basate su identità possono essere ulteriormente classificate come policy inline o policy gestite. Le policy inline sono integrate direttamente in un singolo utente, gruppo o ruolo. Le politiche gestite sono politiche autonome che puoi allegare a più utenti, gruppi e ruoli nel tuo Account AWS. Le politiche gestite includono politiche AWS gestite e politiche gestite dai clienti. Per informazioni su

come scegliere tra una policy gestita o una policy inline, consulta Scelta fra policy gestite e policy inline nella Guida per l'utente IAM.

Policy basate sulle risorse

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Esempi di policy basate sulle risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le operazioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario specificare un principale in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Le policy basate sulle risorse sono policy inline che si trovano in tale servizio. Non puoi utilizzare le policy AWS gestite di IAM in una policy basata sulle risorse.

Liste di controllo degli accessi () ACLs

Le liste di controllo degli accessi (ACLs) controllano quali principali (membri dell'account, utenti o ruoli) dispongono delle autorizzazioni per accedere a una risorsa. ACLs sono simili alle politiche basate sulle risorse, sebbene non utilizzino il formato del documento di policy JSON.

Amazon S3 e Amazon VPC sono esempi di servizi che supportano. AWS WAF ACLs Per ulteriori informazioni ACLs, consulta la panoramica della lista di controllo degli accessi (ACL) nella Amazon Simple Storage Service Developer Guide.

Altri tipi di policy

AWS supporta tipi di policy aggiuntivi e meno comuni. Questi tipi di policy possono impostare il numero massimo di autorizzazioni concesse dai tipi di policy più comuni.

- Limiti delle autorizzazioni: un limite delle autorizzazioni è una funzionalità avanzata nella quale si imposta il numero massimo di autorizzazioni che una policy basata su identità può concedere a un'entità IAM (utente o ruolo IAM). È possibile impostare un limite delle autorizzazioni per un'entità. Le autorizzazioni risultanti sono l'intersezione delle policy basate su identità dell'entità e i relativi limiti delle autorizzazioni. Le policy basate su risorse che specificano l'utente o il ruolo nel campo `Principal`sono condizionate dal limite delle autorizzazioni. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni sui limiti delle autorizzazioni, consulta Limiti delle autorizzazioni per le entità IAM nella Guida per l'utente IAM.

- Politiche di controllo del servizio (SCPs): SCPs sono politiche JSON che specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa (OU) in. AWS Organizations AWS Organizations è un servizio per il raggruppamento e la gestione centralizzata di più di proprietà dell' Account AWS azienda. Se abiliti tutte le funzionalità di un'organizzazione, puoi applicare le politiche di controllo del servizio (SCPs) a uno o tutti i tuoi account. L'SCP limita le autorizzazioni per le entità presenti negli account dei membri, inclusa ciascuna di esse. Utente root dell'account AWS Per ulteriori informazioni su Organizations and SCPs, consulta le politiche di controllo dei servizi nella Guida AWS Organizations per l'utente.
- Politiche di controllo delle risorse (RCPs): RCPs sono politiche JSON che puoi utilizzare per impostare le autorizzazioni massime disponibili per le risorse nei tuoi account senza aggiornare le politiche IAM allegate a ciascuna risorsa di tua proprietà. L'RCP limita le autorizzazioni per le risorse negli account dei membri e può influire sulle autorizzazioni effettive per le identità, incluse le Utente root dell'account AWS, indipendentemente dal fatto che appartengano o meno all'organizzazione. Per ulteriori informazioni su Organizations e RCPs, incluso un elenco di Servizi AWS tale supporto RCPs, vedere Resource control policies (RCPs) nella Guida per l'AWS Organizations utente.
- Policy di sessione: le policy di sessione sono policy avanzate che vengono trasmesse come parametro quando si crea in modo programmatico una sessione temporanea per un ruolo o un utente federato. Le autorizzazioni della sessione risultante sono l'intersezione delle policy basate su identità del ruolo o dell'utente e le policy di sessione. Le autorizzazioni possono anche provenire da una policy basata su risorse. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni, consulta Policy di sessione nella Guida per l'utente IAM.

Più tipi di policy

Quando più tipi di policy si applicano a una richiesta, le autorizzazioni risultanti sono più complicate da comprendere. Per scoprire come si AWS determina se consentire o meno una richiesta quando sono coinvolti più tipi di policy, consulta la logica di valutazione delle policy nella IAM User Guide.

Come Amazon Polly funziona con IAM

Prima di utilizzare IAM per gestire l'accesso ad Amazon Polly, scopri quali funzioni IAM sono disponibili per l'uso con Amazon Polly.

Funzionalità di IAM che puoi utilizzare con Amazon Polly

Funzionalità IAM	Supporto di Amazon Polly
Policy basate su identità	Sì
Policy basate su risorse	No
Azioni di policy	Sì
Risorse relative alle policy	Sì
Chiavi di condizione della policy (specifica del servizio)	No
ACLs	No
ABAC (tag nelle policy)	No
Credenziali temporanee	Sì
Sessioni di accesso diretto (FAS) per Amazon Polly	Sì
Ruoli di servizio	No
Ruoli collegati al servizio	No

Per avere una visione di alto livello di come Amazon Polly e AWS altri servizi funzionano con la maggior parte delle funzionalità IAM, AWS consulta i servizi che funzionano con IAM nella IAM User Guide.

Policy basate su identità per Amazon Polly

Supporta le policy basate su identità: sì

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le operazioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente nella Guida per l'utente IAM.

Con le policy basate su identità di IAM, è possibile specificare quali operazioni e risorse sono consentite o respinte, nonché le condizioni in base alle quali le operazioni sono consentite o respinte. Non è possibile specificare l'entità principale in una policy basata sull'identità perché si applica all'utente o al ruolo a cui è associato. Per informazioni su tutti gli elementi utilizzabili in una policy JSON, consulta Guida di riferimento agli elementi delle policy JSON IAM nella Guida per l'utente di IAM.

Esempi di policy basate su identità Amazon Polly

Per visualizzare esempi di policy basate su identità di Amazon Polly, consulta Esempi di policy basate su identità Amazon Polly.

Policy basate su risorse all'interno di Amazon Polly

Supporta le policy basate su risorse: no

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Esempi di policy basate sulle risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le operazioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario specificare un principale in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Per consentire l'accesso multi-account, puoi specificare un intero account o entità IAM in un altro account come principale in una policy basata sulle risorse. L'aggiunta di un principale multi-account a una policy basata sulle risorse rappresenta solo una parte della relazione di trust. Quando il principale e la risorsa sono diversi Account AWS, un amministratore IAM dell'account affidabile deve inoltre concedere all'entità principale (utente o ruolo) l'autorizzazione ad accedere alla risorsa. L'autorizzazione viene concessa collegando all'entità una policy basata sull'identità. Tuttavia, se una policy basata su risorse concede l'accesso a un principale nello stesso account, non sono richieste ulteriori policy basate su identità. Per ulteriori informazioni, consulta Accesso a risorse multi-account in IAM nella Guida per l'utente IAM.

Operazioni delle policy per Amazon Polly

Supporta le operazioni di policy: si

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento `Action`di una policy JSON descrive le operazioni che è possibile utilizzare per consentire o negare l'accesso a un criterio. Le azioni politiche in genere hanno lo stesso nome dell'operazione AWS API associata. Ci sono alcune eccezioni, ad esempio le operazioni di sola autorizzazione che non hanno un'operazione API corrispondente. Esistono anche alcune operazioni che richiedono più operazioni in una policy. Queste operazioni aggiuntive sono denominate operazioni dipendenti.

Includi le operazioni in una policy per concedere le autorizzazioni a eseguire l'operazione associata.

Per visualizzare un elenco di operazioni di Amazon EC2, consulta Operazioni definite da Amazon EC2 nella Guida di riferimento per l'autorizzazione al servizio.

Le operazioni delle policy in Amazon Polly utilizzano il seguente prefisso prima dell'operazione:

```
polly
```

Per specificare più operazioni in una sola istruzione, occorre separarle con la virgola.

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

Per visualizzare esempi di policy basate su identità di Amazon Polly, consulta Esempi di policy basate su identità Amazon Polly.

Risorse delle policy per Amazon Polly

Supporta le risorse di policy: sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento JSON `Resource`della policy specifica l'oggetto o gli oggetti ai quali si applica l'operazione. Le istruzioni devono includere un elemento `Resource`o un elemento `NotResource`. Come best practice, specifica una risorsa utilizzando il suo nome della risorsa Amazon (ARN). È possibile eseguire questa operazione per operazioni che supportano un tipo di risorsa specifico, note come autorizzazioni a livello di risorsa.

Per le azioni che non supportano le autorizzazioni a livello di risorsa, ad esempio le operazioni di elenco, utilizza un carattere jolly (*) per indicare che l'istruzione si applica a tutte le risorse.

```
"Resource": "*"
```

Per visualizzare un elenco dei tipi di risorse Amazon Polly e relativi ARNs, consulta Risorse definite da Amazon Polly nel Service Authorization Reference. Per informazioni sulle operazioni con cui è possibile specificare l'ARN di ogni risorsa, consulta Operazioni definite da Amazon Polly.

Per visualizzare esempi di policy basate su identità di Amazon Polly, consulta Esempi di policy basate su identità Amazon Polly.

Chiavi di condizione delle policy per Amazon Polly

Supporta le chiavi delle condizioni delle policy specifiche del servizio: No

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento `Condition`(o blocco `Condition`) consente di specificare le condizioni in cui un'istruzione è in vigore. L'elemento `Condition`è facoltativo. È possibile compilare espressioni condizionali che utilizzano operatori di condizione, ad esempio uguale a o minore di, per soddisfare la condizione nella policy con i valori nella richiesta.

Se specifichi più elementi `Condition`in un'istruzione o più chiavi in un singolo elemento `Condition`, questi vengono valutati da AWS utilizzando un'operazione `AND`logica. Se si specificano più valori per una singola chiave di condizione, AWS valuta la condizione utilizzando un'operazione logica. `OR` Tutte le condizioni devono essere soddisfatte prima che le autorizzazioni dell'istruzione vengano concesse.

È possibile anche utilizzare variabili segnaposto quando specifichi le condizioni. Ad esempio, è possibile autorizzare un utente IAM ad accedere a una risorsa solo se è stata taggata con il relativo nome utente IAM. Per ulteriori informazioni, consulta Elementi delle policy IAM: variabili e tag nella Guida per l'utente di IAM.

AWS supporta chiavi di condizione globali e chiavi di condizione specifiche del servizio. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di contesto delle condizioni AWS globali nella Guida per l'utente IAM.

Per visualizzare un elenco di chiavi di condizione Amazon Polly, consulta Chiavi di condizione per Amazon Polly nella Guida di riferimento per l'autorizzazione al servizio. Per scoprire con quali azioni e risorse puoi utilizzare una chiave di condizione, consulta Operazioni definite da Amazon Polly.

Per visualizzare esempi di policy basate su identità di Amazon Polly, consulta Esempi di policy basate su identità Amazon Polly.

ACLs in Amazon Polly

Supporti ACLs: no

Le liste di controllo degli accessi (ACLs) controllano quali principali (membri dell'account, utenti o ruoli) dispongono delle autorizzazioni per accedere a una risorsa. ACLs sono simili alle politiche basate sulle risorse, sebbene non utilizzino il formato del documento di policy JSON.

ABAC con Amazon Polly

Supporta ABAC (tag nelle politiche): No

Il controllo dell'accesso basato su attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi. In AWS, questi attributi sono chiamati tag. Puoi allegare tag a entità IAM (utenti o ruoli) e a molte AWS risorse. L'assegnazione di tag alle entità e alle risorse è il primo passaggio di ABAC. In seguito, vengono progettate policy ABAC per consentire operazioni quando il tag dell'entità principale corrisponde al tag sulla risorsa a cui si sta provando ad accedere.

La strategia ABAC è utile in ambienti soggetti a una rapida crescita e aiuta in situazioni in cui la gestione delle policy diventa impegnativa.

Per controllare l'accesso basato su tag, fornisci informazioni sui tag nell'elemento condizione di una policy utilizzando le chiavi di condizione `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`*o `aws:TagKeys`.

Se un servizio supporta tutte e tre le chiavi di condizione per ogni tipo di risorsa, il valore per il servizio è Yes (Sì). Se un servizio supporta tutte e tre le chiavi di condizione solo per alcuni tipi di risorsa, allora il valore sarà Parziale.

Per ulteriori informazioni su ABAC, consulta Definizione delle autorizzazioni con autorizzazione ABAC nella Guida per l'utente IAM. Per visualizzare un tutorial con i passaggi per l'impostazione di ABAC, consulta Utilizzo del controllo degli accessi basato su attributi (ABAC) nella Guida per l'utente di IAM.

Utilizzo di credenziali temporanee con Amazon Polly

Supporta le credenziali temporanee: sì

Alcuni Servizi AWS non funzionano quando accedi utilizzando credenziali temporanee. Per ulteriori informazioni, incluse quelle che Servizi AWS funzionano con credenziali temporanee, consulta la sezione relativa alla Servizi AWS compatibilità con IAM nella IAM User Guide.

Stai utilizzando credenziali temporanee se accedi AWS Management Console utilizzando qualsiasi metodo tranne nome utente e password. Ad esempio, quando accedete AWS utilizzando il link Single Sign-On (SSO) della vostra azienda, tale processo crea automaticamente credenziali temporanee. Le credenziali temporanee vengono create in automatico anche quando accedi alla console come utente e poi cambi ruolo. Per ulteriori informazioni sullo scambio dei ruoli, consulta Passaggio da un ruolo utente a un ruolo IAM (console) nella Guida per l'utente IAM.

È possibile creare manualmente credenziali temporanee utilizzando l'API or. AWS CLI AWS È quindi possibile utilizzare tali credenziali temporanee per accedere. AWS AWS consiglia di generare dinamicamente credenziali temporanee anziché utilizzare chiavi di accesso a lungo termine. Per ulteriori informazioni, consulta Credenziali di sicurezza provvisorie in IAM.

Sessioni di accesso diretto (FAS) tra servizi per Amazon Polly

Supporta l'inoltro delle sessioni di accesso (FAS): sì

Quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama an Servizio AWS, in combinazione con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli delle policy relative alle richieste FAS, consulta Forward access sessions.

Ruoli di servizio per Amazon Polly

Supporta i ruoli di servizio: no

Un ruolo di servizio è un ruolo IAM che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per

ulteriori informazioni, consulta la sezione Create a role to delegate permissions to an Servizio AWS nella Guida per l'utente IAM.

> **Warning**
>
> La modifica delle autorizzazioni per un ruolo di servizio potrebbe compromettere la funzionalità di Amazon Polly. Modifica i ruoli del servizio solo quando Amazon Polly fornisce le indicazioni per farlo.

Ruoli collegati ai servizi per Amazon Polly

Supporta i ruoli collegati ai servizi: no

Un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.

Per ulteriori informazioni su come creare e gestire i ruoli collegati ai servizi, consulta Servizi AWS supportati da IAM. Trova un servizio nella tabella che include un `Yes` nella colonna Service-linked role (Ruolo collegato ai servizi). Scegli il collegamento Sì per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Ruoli IAM di Amazon Polly

Per concedere autorizzazioni multiaccount, puoi collegare una policy di autorizzazione basata su identità a un ruolo IAM. Ad esempio, l'amministratore dell'account A può creare un ruolo per concedere autorizzazioni su più account a un altro account (ad esempio, l' AWS account B) o a un AWS servizio nel modo seguente:

1. L'amministratore dell'account A crea un ruolo IAM e attribuisce una policy di autorizzazione al ruolo che concede le autorizzazioni per le risorse nell'account A.
2. L'amministratore dell'account A collega una policy di attendibilità al ruolo, identificando l'account B come principale per tale ruolo.
3. L'amministratore dell'account B può quindi delegare le autorizzazioni per assumere il ruolo a qualsiasi utente dell'account B. In questo modo gli utenti dell'account B possono creare o accedere alle risorse nell'account A. Il responsabile della politica di fiducia può anche essere

un responsabile del AWS servizio se si desidera concedere a un servizio le autorizzazioni per assumere il ruolo. AWS

Per ulteriori informazioni sull'uso di IAM per delegare le autorizzazioni, consulta Access Management nella IAM User Guide (Guida per l'utente di IAM).

Di seguito è riportata una policy di esempio che concede autorizzazioni per inserire e ricevere lessici, nonché per elencare i lessici attualmente disponibili.

Amazon Polly supporta le policy basate su identità per le operazioni a livello di risorsa. In alcuni casi, la risorsa può essere limitata da un ARN. Questo è vero per le operazioni `SynthesizeSpeech`, `StartSpeechSynthesisTask`, `PutLexicon`, `GetLexicon` e `DeleteLexicon`. In questi casi, il valore `Resource` è indicato dall'ARN. L'elemento `arn:aws:polly:`*`us-east-2`*`:`*`account-id`*`:lexicon/*` come valore `Resource` specifica ad esempio autorizzazioni su tutti i lessici di proprietà nella regione `us-east-2`.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPut-Get-ListActions",
            "Effect": "Allow",
            "Action": [
                "polly:PutLexicon",
                "polly:GetLexicon",
                "polly:ListLexicons"
            ],
            "Resource": "arn:aws:polly:us-east-2:111122223333:lexicon/*"
        }
    ]
}
```

Tuttavia, non tutte le operazioni utilizzano. ARNs Questo è il caso con le operazioni `DescribeVoices`,`ListLexicons`,`GetSpeechSynthesisTasks` e `ListSpeechSynthesisTasks`.

Per ulteriori informazioni su utenti, gruppi, ruoli e autorizzazioni, consulta Identità (utenti, gruppi e ruoli) nella Guida per l'utente di IAM.

Esempi di policy basate su identità Amazon Polly

Per impostazione predefinita, gli utenti e i ruoli non dispongono dell'autorizzazione per creare o modificare risorse Amazon Polly. Inoltre, non possono eseguire attività utilizzando AWS Management Console, AWS Command Line Interface (AWS CLI) o AWS l'API. Per concedere agli utenti l'autorizzazione a eseguire operazioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM. L'amministratore può quindi aggiungere le policy IAM ai ruoli e gli utenti possono assumere i ruoli.

Per informazioni su come creare una policy basata su identità IAM utilizzando questi documenti di policy JSON di esempio, consulta Creazione di policy IAM (console) nella Guida per l'utente IAM.

Per informazioni dettagliate sulle azioni e sui tipi di risorse definiti da Amazon Polly, incluso il formato di ARNs per ogni tipo di risorsa, consulta Azioni, risorse e chiavi di condizione per Amazon Polly nel Service Authorization Reference.

Argomenti

- Best practice per le policy
- Utilizzo della console Amazon Polly
- Consentire agli utenti di visualizzare le loro autorizzazioni
- AWS politiche gestite (predefinite) per Amazon Polly
- Esempi di policy gestite dal cliente

Best practice per le policy

Le policy basate su identità determinano se qualcuno può creare, accedere o eliminare risorse Amazon Polly nell'account. Queste azioni possono comportare costi aggiuntivi per l' Account AWS. Quando crei o modifichi policy basate su identità, segui queste linee guida e raccomandazioni:

- Inizia con le politiche AWS gestite e passa alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le policy AWS gestite che concedono le autorizzazioni per molti casi d'uso comuni. Sono disponibili nel tuo. Account AWS Ti consigliamo di ridurre ulteriormente le autorizzazioni definendo politiche gestite dai AWS clienti specifiche per i tuoi casi d'uso. Per ulteriori informazioni, consulta Policy gestite da AWSo Policy gestite da AWS per le funzioni dei processi nella Guida per l'utente IAM.

- Applica le autorizzazioni con privilegio minimo: quando imposti le autorizzazioni con le policy IAM, concedi solo le autorizzazioni richieste per eseguire un'attività. È possibile farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegi minimi. Per ulteriori informazioni sull'utilizzo di IAM per applicare le autorizzazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente IAM.
- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso: per limitare l'accesso a operazioni e risorse è possibile aggiungere una condizione alle tue policy. Ad esempio, è possibile scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzando SSL. Puoi anche utilizzare le condizioni per concedere l'accesso alle azioni del servizio se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio AWS CloudFormation. Per ulteriori informazioni, consulta la sezione Elementi delle policy JSON di IAM: condizione nella Guida per l'utente IAM.
- Utilizzo di IAM Access Analyzer per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali: IAM Access Analyzer convalida le policy nuove ed esistenti in modo che aderiscano alla sintassi della policy IAM (JSON) e alle best practice di IAM. IAM Access Analyzer offre oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Per ulteriori informazioni, consulta Convalida delle policy per il Sistema di analisi degli accessi IAM nella Guida per l'utente IAM.
- Richiedi l'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o un utente root nel Account AWS tuo, attiva l'MFA per una maggiore sicurezza. Per richiedere la MFA quando vengono chiamate le operazioni API, aggiungi le condizioni MFA alle policy. Per ulteriori informazioni, consulta Protezione dell'accesso API con MFA nella Guida per l'utente IAM.

Per maggiori informazioni sulle best practice in IAM, consulta Best practice di sicurezza in IAM nella Guida per l'utente di IAM.

Utilizzo della console Amazon Polly

Per accedere alla console Amazon Polly, è necessario disporre di un set di autorizzazioni minimo. Queste autorizzazioni devono consentirti di elencare e visualizzare i dettagli sulle risorse Amazon Polly presenti nel tuo. Account AWS Se crei una policy basata sull'identità più restrittiva rispetto alle autorizzazioni minime richieste, la console non funzionerà nel modo previsto per le entità (utenti o ruoli) associate a tale policy.

Non è necessario consentire autorizzazioni minime di console per gli utenti che effettuano chiamate solo verso AWS CLI o l'API. AWS Al contrario, concedi l'accesso solo alle operazioni che corrispondono all'operazione API che stanno cercando di eseguire.

Per garantire che utenti e ruoli possano continuare a utilizzare la console Amazon Polly, collega anche Amazon `ConsoleAccess` Polly `ReadOnly` AWS o la policy gestita alle entità. Per ulteriori informazioni, consulta Aggiunta di autorizzazioni a un utente nella Guida per l'utente IAM.

Per utilizzare la console Amazon Polly, concedi le autorizzazioni a tutti gli Amazon Polly. APIs Non sono necessarie autorizzazioni aggiuntive Per ottenere la funzionalità completa della console, è possibile utilizzare le seguenti policy:.

JSON

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

Consentire agli utenti di visualizzare le loro autorizzazioni

Questo esempio mostra in che modo è possibile creare una policy che consente agli utenti IAM di visualizzare le policy inline e gestite che sono cpllegate alla relativa identità utente. Questa politica include le autorizzazioni per completare questa azione sulla console o utilizzando l'API o a livello di codice. AWS CLI AWS

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
```

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

AWS politiche gestite (predefinite) per Amazon Polly

AWS affronta molti casi d'uso comuni fornendo politiche IAM autonome create e amministrate da. AWS Queste policy AWS gestite concedono le autorizzazioni necessarie per i casi d'uso comuni in modo da evitare di dover esaminare quali autorizzazioni sono necessarie. Per ulteriori informazioni, consulta Policy gestite da AWS nella Guida per l'utente di IAM.

Le seguenti politiche AWS gestite, che puoi allegare agli utenti del tuo account, sono specifiche di Amazon Polly:

- AmazonPollyReadOnlyAccess— Garantisce l'accesso in sola lettura alle risorse, consente di elencare lessici, recuperare lessici, elencare le voci disponibili e sintetizzare il parlato (inclusa l'applicazione di lessici al parlato sintetizzato).
- AmazonPollyFullAccess— Garantisce l'accesso completo alle risorse e a tutte le operazioni supportate.

> **Note**
>
> Per esaminare queste policy di autorizzazione, accedi alla console IAM ed esegui la ricerca delle policy specifiche.

È inoltre possibile creare policy IAM personalizzate per concedere le autorizzazioni per operazioni e risorse Amazon Polly. Puoi associare queste policy personalizzate agli utenti o ai gruppi IAM che richiedono tali autorizzazioni.

Esempi di policy gestite dal cliente

Questa sezione include policy utente di esempio che concedono autorizzazioni per diverse operazioni Amazon Polly. Queste politiche funzionano quando utilizzi AWS SDKs o il AWS CLI. Quando usi la console, concedi le autorizzazioni a tutti gli Amazon APIs Polly.

> **Note**
>
> Tutti gli esempi utilizzano la regione us-east-2 e contengono account fittizi. IDs

Esempi

- Esempio 1: consenti tutte le operazioni Amazon Polly
- Esempio 2: consenti tutte le azioni di Amazon Polly tranne DeleteLexicon
- Esempio 3: Consenti DeleteLexicon
- Esempio 4: consenti Delete Lexicon in una regione specifica
- Esempio 5: Consenti DeleteLexicon il lessico specificato

Esempio 1: consenti tutte le operazioni Amazon Polly

Dopo aver effettuato l'accesso (consultare Guida introduttiva ad Amazon Polly), è necessario creare un utente amministratore per gestire l'account, ad esempio per creare utenti e gestirne le autorizzazioni.

È possibile creare un utente che dispone delle autorizzazioni per tutte le operazioni Amazon Polly. Considerare questo utente come un amministratore specifico del servizio per l'utilizzo con Amazon Polly. È possibile collegare la policy di autorizzazione seguente a tale utente.

JSON

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

Esempio 2: consenti tutte le azioni di Amazon Polly tranne DeleteLexicon

La policy di autorizzazione seguente concede le autorizzazioni utente per eseguire tutte le operazioni ad eccezione di `DeleteLexicon`, con le autorizzazioni per l'eliminazione negate esplicitamente in tutte le regioni.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Sid": "AllowAllActions-DenyDelete",
      "Effect": "Allow",
      "Action": [
         "polly:DescribeVoices",
         "polly:GetLexicon",
         "polly:PutLexicon",
         "polly:SynthesizeSpeech",
         "polly:ListLexicons"],
      "Resource": "*"
      }
      {
      "Sid": "DenyDeleteLexicon",
      "Effect": "Deny",
      "Action": [
         "polly:DeleteLexicon"],
      "Resource": "*"
      }
   ]
```

```
}
```

Esempio 3: Consenti DeleteLexicon

La policy di autorizzazione seguente concede le autorizzazioni utente per eliminare qualsiasi lessico di proprietà indipendentemente dal progetto o dalla regione in cui si trova.

JSON

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

Esempio 4: consenti Delete Lexicon in una regione specifica

La policy di autorizzazione seguente concede le autorizzazioni utente per eliminare qualsiasi lessico in tutti i progetti che si trovano in una regione specifica (in questo caso, us-east-2).

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
      "Sid": "AllowDeleteSpecifiedRegion",
      "Effect": "Allow",
      "Action": [
         "polly:DeleteLexicon"],
      "Resource": "arn:aws:polly:us-east-2:123456789012:lexicon/*"
      }
   ]
}
```

Esempio 5: Consenti DeleteLexicon il lessico specificato

La policy di autorizzazione seguente concede le autorizzazioni utente per eliminare un lessico specifico di proprietà (in questo caso, myLexicon) in una regione specifica (in questo caso, us-east-2).

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
      "Sid": "AllowDeleteForSpecifiedLexicon",
      "Effect": "Allow",
      "Action": [
         "polly:DeleteLexicon"],
      "Resource": "arn:aws:polly:us-east-2:123456789012:lexicon/myLexicon"
      }
   ]
}
```

Autorizzazioni API Amazon Polly: informazioni di riferimento su operazioni, autorizzazioni e risorse

Quando configuri una politica di autorizzazioni che puoi allegare a un'identità IAM (politiche basate sull'identità), puoi utilizzare il seguente elenco di come riferimento. L' include ogni operazione dell'API Amazon Polly, le azioni corrispondenti per le quali è possibile concedere le autorizzazioni per eseguire l'azione e la AWS risorsa per la quale è possibile concedere le autorizzazioni. Puoi specificare le azioni nel campo `Action` della policy e il valore della risorsa nel campo `Resource`.

Puoi utilizzare le chiavi di condizione AWS-wide nelle tue politiche di Amazon Polly per esprimere le condizioni. Per un elenco completo delle chiavi AWS-wide, consulta le chiavi disponibili nella IAM User Guide.

> **Note**
>
> Per specificare un'operazione, utilizza il prefisso `polly` seguito dal nome dell'operazione API (ad esempio, `polly:GetLexicon`).

Amazon Polly supporta le policy basate su identità per le operazioni a livello di risorsa. Di conseguenza, il valore `Resource` è indicato dall'ARN. L'elemento `arn:aws:polly:us-east-2:account-id:lexicon/*` come valore `Resource` specifica ad esempio autorizzazioni su tutti i lessici di proprietà nella regione `us-east-2`.

Poiché Amazon Polly non supporta le autorizzazioni per operazioni a livello di risorsa, la maggior parte delle policy specifica un carattere jolly (*) come il valore `Resource`. Se è necessario limitare le autorizzazioni a una regione specifica, tuttavia, il carattere jolly è sostituito con l'ARN appropriato: `arn:aws:polly:region:account-id:lexicon/*`.

API Amazon Polly e autorizzazioni richieste per le operazioni

azione API: DeleteLexicon

Autorizzazioni richieste (azione API): `polly:DeleteLexicon`

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

azione API: DescribeVoices

Autorizzazioni richieste (azione API): `polly:DescribeVoices`

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

azione API: GetLexicon

Autorizzazioni richieste (azione API): `polly:GetLexicon`

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

azione API: ListLexicons

Autorizzazioni richieste (azione API): `polly:ListLexicons`

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

azione API: PutLexicon

Autorizzazioni richieste (azione API): `polly:ListLexicons`

Risorse: *

azione API: SynthesizeSpeech

Autorizzazioni richieste (azione API): `polly:SynthesizeSpeech`

Risorse: *

Risoluzione dei problemi relativi all'identità e all'accesso di Amazon Polly

Utilizza le informazioni seguenti per diagnosticare e risolvere i problemi comuni che possono verificarsi durante l'utilizzo di Amazon Polly e IAM.

Argomenti

- Non dispongo dell'autorizzazione per eseguire un'operazione in Amazon Polly
- Non sono autorizzato a eseguire iam: PassRole
- Desidero consentire a persone esterne Account AWS a me di accedere alle mie risorse Amazon Polly

Non dispongo dell'autorizzazione per eseguire un'operazione in Amazon Polly

Se ricevi un errore che indica che non sei autorizzato a eseguire un'operazione, le tue policy devono essere aggiornate per poter eseguire l'operazione.

L'errore di esempio seguente si verifica quando l'utente IAM `mateojackson` prova a utilizzare la console per visualizzare i dettagli relativi a una risorsa *`my-example-widget`* fittizia ma non dispone di autorizzazioni `polly:`*`GetWidget`* fittizie.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 polly:GetWidget on resource: my-example-widget
```

In questo caso, la policy per l'utente `mateojackson` deve essere aggiornata per consentire l'accesso alla risorsa *`my-example-widget`* utilizzando l'azione `polly:`*`GetWidget`*.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Non sono autorizzato a eseguire iam: PassRole

Se ricevi un errore che indica che non si è autorizzati a eseguire l'operazione `iam:PassRole`, è necessario aggiornare le policy per poter passare un ruolo ad Amazon Polly.

Alcuni Servizi AWS consentono di passare un ruolo esistente a quel servizio invece di creare un nuovo ruolo di servizio o un ruolo collegato al servizio. Per eseguire questa operazione, è necessario disporre delle autorizzazioni per trasmettere il ruolo al servizio.

Il seguente esempio di errore si verifica quando un utente IAM denominato `marymajor` cerca di utilizzare la console per eseguire un'operazione in Amazon Polly. Tuttavia, l'azione richiede che il servizio disponga delle autorizzazioni concesse da un ruolo di servizio. Mary non dispone delle autorizzazioni per passare il ruolo al servizio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In questo caso, le policy di Mary devono essere aggiornate per poter eseguire l'operazione `iam:PassRole`.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Desidero consentire a persone esterne Account AWS a me di accedere alle mie risorse Amazon Polly

È possibile creare un ruolo con il quale utenti in altri account o persone esterne all'organizzazione possono accedere alle tue risorse. È possibile specificare chi è attendibile per l'assunzione del ruolo. Per i servizi che supportano politiche basate sulle risorse o liste di controllo degli accessi (ACLs), puoi utilizzare tali politiche per consentire alle persone di accedere alle tue risorse.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Per sapere se Amazon Polly supporta queste caratteristiche, consulta Come Amazon Polly funziona con IAM.
- Per scoprire come fornire l'accesso alle tue risorse attraverso Account AWS le risorse di tua proprietà, consulta Fornire l'accesso a un utente IAM in un altro Account AWS di tua proprietà nella IAM User Guide.
- Per scoprire come fornire l'accesso alle tue risorse a terze parti Account AWS, consulta Fornire l'accesso a soggetti Account AWS di proprietà di terze parti nella Guida per l'utente IAM.
- Per informazioni su come fornire l'accesso tramite la federazione delle identità, consulta Fornire l'accesso a utenti autenticati esternamente (Federazione delle identità) nella Guida per l'utente IAM.

- Per informazioni sulle differenze di utilizzo tra ruoli e policy basate su risorse per l'accesso multi-account, consulta Accesso a risorse multi-account in IAM nella Guida per l'utente IAM.

Registrazione e monitoraggio in Amazon Polly

Il monitoraggio è importante per garantire l'affidabilità, la disponibilità e le prestazioni delle applicazioni Amazon Polly. Per monitorare le chiamate all'API Amazon Polly, puoi utilizzare. AWS CloudTrail Per monitorare lo stato dei tuoi lavori, usa Amazon CloudWatch Logs.

- Amazon CloudWatch Alarms: utilizzando gli CloudWatch allarmi, controlli una singola metrica per un periodo di tempo specificato. Se la metrica supera una determinata soglia, viene inviata una notifica a un argomento o AWS Auto Scaling una politica di Amazon Simple Notification Service. CloudWatchgli allarmi non richiamano azioni quando una metrica si trova in uno stato particolare. È necessario invece cambiare lo stato e mantenerlo per un numero di periodi specificato. Per ulteriori informazioni, consulta Integrazione CloudWatch con Amazon Polly.
- CloudTrail log: CloudTrail fornisce un registro delle azioni intraprese da un utente, un ruolo o un AWS servizio in Amazon Polly. Utilizzando le informazioni raccolte da CloudTrail, puoi determinare la richiesta inviata ad Amazon Polly. Puoi determinare l'indirizzo IP da cui è stata effettuata la richiesta, l'autore della richiesta, il momento in cui è stata effettuata e i dettagli aggiuntivi. Per ulteriori informazioni, consulta Registrazione delle chiamate API Amazon Polly con AWS CloudTrail.

Convalida della conformità per Amazon Polly

I revisori di terze parti valutano la sicurezza e la conformità di Amazon Polly nell'ambito di AWS diversi programmi di conformità. Questi includono SOC, PCI, FedRAMP, HIPAA e altri.

Per un elenco dei AWS servizi che rientrano nell'ambito di specifici programmi di conformità, consulta AWS Services in Scope by Compliance Program AWS Program. Per informazioni generali, vedere Programmi di AWS conformità Programmi di di .

È possibile scaricare report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta Download di report in AWS Artifact.

La tua responsabilità di conformità quando utilizzi Amazon Polly è determinata dalla sensibilità dei tuoi dati, dagli obiettivi di conformità della tua azienda e dalle leggi e dai regolamenti applicabili. AWS fornisce le seguenti risorse per contribuire alla conformità:

- Security and Compliance Quick Start Guides (Guide Quick Start Sicurezza e compliance): queste guide alla distribuzione illustrano considerazioni relative all'architettura e forniscono procedure per la distribuzione di ambienti di base incentrati sulla sicurezza e sulla conformità su AWS.
- Whitepaper sull'architettura per la sicurezza e la conformità HIPAA: questo white paper descrive come le aziende possono utilizzare per creare applicazioni conformi allo standard HIPAA. AWS
- AWS Risorse per la conformità Risorse per la conformità: questa raccolta di potrebbe riguardare il settore e la località in cui operate.
- Valutazione delle risorse con le regole nella Guida per gli AWS Config sviluppatori: il AWS Config servizio valuta la conformità delle configurazioni delle risorse alle pratiche interne, alle linee guida del settore e alle normative.
- AWS Security Hub— Questo AWS servizio offre una visione completa dello stato di sicurezza dell'utente, AWS che consente di verificare la conformità agli standard e alle best practice del settore della sicurezza.

Resilienza in Amazon Polly

L'infrastruttura AWS globale è costruita attorno a AWS regioni e zone di disponibilità. AWS Le regioni forniscono più zone di disponibilità fisicamente separate e isolate, collegate con reti a bassa latenza, ad alto throughput e altamente ridondanti. Con le zone di disponibilità, puoi progettare e gestire applicazioni e database che eseguono automaticamente il failover tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture a data center singolo o multiplo tradizionali.

Per ulteriori informazioni su AWS regioni e zone di disponibilità, consulta Global Infrastructure.AWS

Sicurezza dell'infrastruttura in Amazon Polly

In quanto servizio gestito, Amazon Polly è protetto dalle procedure di sicurezza di rete AWS globali descritte nel white paper Amazon Web Services: Overview of Security Processes.

Utilizzi chiamate API AWS pubblicate per accedere ad Amazon Polly attraverso la rete. I client devono supportare Transport Layer Security (TLS) 1.0 o versioni successive. È consigliabile TLS 1.2 o versioni successive. I client devono, inoltre, supportare le suite di cifratura con PFS (Perfect Forward Secrecy), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Inoltre, le richieste devono essere firmate utilizzando un ID chiave di accesso e una chiave di accesso segreta associata a un principale IAM. O puoi utilizzare AWS Security Token Service (AWS STS) per generare credenziali di sicurezza temporanee per sottoscrivere le richieste.

Best practice di sicurezza per Amazon Polly

La tua fiducia, la tua privacy e la sicurezza dei tuoi contenuti sono le nostre maggiori priorità. Implementiamo controlli tecnici e fisici responsabili e sofisticati progettati per impedire l'accesso non autorizzato o la divulgazione dei tuoi contenuti e garantire che il nostro utilizzo rispetti i nostri impegni nei tuoi confronti. Per ulteriori informazioni, consulta le Domande frequenti sulla privacy dei dati in AWS.

Amazon Polly non conserva il contenuto degli invii di testo.

Per un'ampia panoramica sulla AWS sicurezza, tra cui conformità, test di penetrazione, bollettini e risorse, visita il AWS sito Web di Cloud Security.

Uso di Amazon Polly con endpoint VPC dell'interfaccia

Se utilizzi Amazon Virtual Private Cloud (Amazon VPC) per ospitare AWS le tue risorse, puoi stabilire una connessione privata tra il tuo VPC e Amazon Polly. Puoi utilizzare questa connessione per sintetizzare la voce con Amazon Polly senza passare per la rete Internet pubblica.

Amazon VPC è un AWS servizio che puoi utilizzare per avviare AWS risorse in una rete virtuale definita dall'utente. Con un VPC;, detieni il controllo delle impostazioni della rete, come l'intervallo di indirizzi IP, le sottoreti, le tabelle di routing e i gateway di rete. Per collegare il tuo VPC a Amazon Polly, definisci un endpoint VPC dell'interfaccia per Amazon Polly. Questo tipo di endpoint consente di collegare il VPC a Servizi AWS. L'endpoint offre una connettività scalabile e affidabile a Amazon Polly senza richiedere un gateway Internet, un'istanza Network Address Translation (NAT) o una connessione VPN. Per ulteriori informazioni, consulta la sezione Che cos'è Amazon VPC? nella Guida per l'utente di Amazon VPC.

Gli endpoint VPC di interfaccia sono alimentati da AWS PrivateLink, una AWS tecnologia che consente la comunicazione privata tra Servizi AWS l'utilizzo di un'interfaccia di rete elastica con indirizzi IP privati. Per ulteriori informazioni, vedere New - AWS PrivateLink for. Servizi AWS

Le fasi seguenti sono per gli utenti Amazon VPC. Per ulteriori informazioni, consulta l'argomento relativo alle nozioni di base nella Guida per l'utente di Amazon VPC.

Disponibilità

Gli endpoint VPC sono supportati in tutte le regioni in cui è supportato Amazon Polly. Per ulteriori informazioni su AWS regioni e zone di disponibilità, vedere AWS Global Infrastructure.

Creazione di un endpoint VPC per Amazon Polly

Per iniziare a usare Amazon Polly con il VPC, crea un endpoint VPC dell'interfaccia per Amazon Polly. Il servizio da scegliere è com.amazonaws. ***Region***.polly. Non è necessario modificare alcuna impostazione per Amazon Polly. Per ulteriori informazioni, consulta Creazione di un endpoint di interfaccia nella Guida per l'utente di Amazon VPC.

Verifica della connessione tra il VPC e Amazon Polly

Dopo aver creato l'endpoint, è possibile testare la connessione.

Per verificare la connessione tra il VPC e l'endpoint Amazon Polly

1. Connettiti a un' EC2 istanza Amazon che risiede nel tuo VPC. Per informazioni sulla connessione, consulta Connessione all'istanza Linux o Connessione all'istanza Windows nella EC2 documentazione di Amazon.
2. Dall'istanza, utilizza `aws polly describe-voices` della AWS CLI per elencare le voci Amazon Polly disponibili.

Se la risposta al comando include l'elenco di voci Amazon Polly disponibili, il comando è riuscito e l'endpoint VPC funziona correttamente.

Controllo dell'accesso all'endpoint Amazon Polly

Una policy endpoint VPC è una policy della risorsa IAM che viene collegata a un endpoint durante la creazione o la modifica dell'endpoint. Se non colleghi una policy durante la creazione di un endpoint, viene collegata una policy predefinita che consente l'accesso completo al servizio. Una policy endpoint non esclude né sostituisce policy dell'utente IAM o policy specifiche del servizio. Si tratta di una policy separata per controllare l'accesso dall'endpoint al servizio specificato.

Le policy endpoint devono essere scritte in formato JSON.

Per ulteriori informazioni, consulta Controllo degli accessi ai servizi con endpoint VPC nella Guida per l'utente di Amazon VPC.

Di seguito è riportato un esempio di una policy dell'endpoint per Amazon Polly. Questa policy consente agli utenti che si connettono ad Amazon Polly tramite il VPC di descrivere le voci ed eseguire al sintesi vocale con Amazon Polly e impedisce loro di eseguire altre azioni Amazon Polly.

```
{
  "Statement": [
    {
      "Sid": "SynthesisAndDescribeVoicesOnly",
      "Principal": "*",
      "Action": [
        "polly:DescribeVoices",
        "polly:SynthesizeSpeech"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Per modificare la policy di endpoint VPC per Amazon Polly

1. Apri la console Amazon VPC in /vpc. https://console.aws.amazon.com
2. Nel pannello di navigazione, seleziona Endpoint.
3. Se non hai già creato l'endpoint per Amazon Polly, scegli Create Endpoint (Crea endpoint). Quindi seleziona com.amazonaws. *`Region`*.polly e scegli Crea endpoint.
4. Seleziona com.amazonaws. ***`Region`***.polly endpoint e scegli la scheda Policy nella metà inferiore dello schermo.
5. Scegli Edit Policy (Modifica policy) e apporta le modifiche alla policy.

Supporto delle chiavi di contesto VPC

Amazon Polly supporta le `aws:SourceVpc` chiavi `aws:SourceVpce` contestuali che possono limitare l'accesso a endpoint VPC specifici VPCs o specifici. Queste chiavi funzionano solo se l'utente utilizza endpoint VPC. Per ulteriori informazioni, consulta Chiavi disponibili per alcuni servizi nella Guida per l'utente di IAM.

Registrazione delle chiamate API Amazon Polly con AWS CloudTrail

Amazon Polly è integrato con AWS CloudTrail, un servizio che fornisce una registrazione delle azioni intraprese da un utente, ruolo o AWS servizio in Amazon Polly. CloudTrail acquisisce tutte le chiamate API per Amazon Polly come eventi. Le chiamate acquisite includono le chiamate dalla console Amazon Polly e le chiamate di codice alle operazioni API Amazon Polly. Se crei un trail, puoi abilitare la distribuzione continua di CloudTrail eventi a un bucket Amazon S3, inclusi gli eventi per Amazon Polly. Se non configuri un percorso, puoi comunque visualizzare gli eventi più recenti nella CloudTrail console nella cronologia degli eventi. Utilizzando le informazioni raccolte da CloudTrail, puoi determinare la richiesta che è stata effettuata ad Amazon Polly, l'indirizzo IP da cui è stata effettuata, chi ha effettuato la richiesta, quando è stata effettuata e ulteriori dettagli.

Per ulteriori informazioni CloudTrail, incluso come configurarlo e abilitarlo, consulta la Guida per l'AWS CloudTrail utente.

Informazioni su Amazon Polly in CloudTrail

CloudTrail è abilitato sul tuo AWS account al momento della creazione dell'account. Quando si verifica un'attività di evento supportata in Amazon Polly, tale attività viene registrata in un CloudTrail evento insieme ad altri eventi di AWS servizio nella cronologia degli eventi. Puoi visualizzare, cercare e scaricare eventi recenti nel tuo AWS account. Per ulteriori informazioni, consulta Visualizzazione degli eventi con la cronologia degli CloudTrail eventi.

Per una registrazione continua degli eventi nel tuo AWS account, inclusi gli eventi per Amazon Polly, crea un percorso. Un trail consente di CloudTrail inviare file di log a un bucket Amazon S3. Per impostazione predefinita, quando crei un percorso nella console, il percorso si applica a tutte le AWS regioni. Il trail registra gli eventi di tutte le regioni della AWS partizione e consegna i file di log al bucket Amazon S3 specificato. Inoltre, puoi configurare altri AWS servizi per analizzare ulteriormente e agire in base ai dati sugli eventi raccolti nei log. CloudTrail Per ulteriori informazioni, consulta gli argomenti seguenti:

- Panoramica della creazione di un trail
- CloudTrail Servizi e integrazioni supportati
- Configurazione delle notifiche Amazon SNS per CloudTrail

- Ricezione di file di CloudTrail registro da più regioni e ricezione di file di CloudTrail registro da più account

Amazon Polly supporta la registrazione delle seguenti azioni come eventi nei CloudTrail file di registro:

- DeleteLexicon
- DescribeVoices
- GetLexicon
- GetSpeechSynthesisTask
- ListLexicons
- ListSpeechSynthesisTasks
- PutLexicon
- StartSpeechSynthesisTask
- SynthesizeSpeech

Ogni evento o voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di determinare quanto segue:

- Se la richiesta è stata effettuata con credenziali utente root o utente AWS Identity and Access Management (IAM).
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata effettuata da un altro AWS servizio.

Per ulteriori informazioni, consulta Elemento CloudTrail userIdentity.

Esempio: voci del file di log di Amazon Polly

Un trail è una configurazione che consente la distribuzione di eventi come file di log in un bucket Amazon S3 specificato dall'utente. CloudTrail i file di registro contengono una o più voci di registro. Un evento rappresenta una singola richiesta proveniente da qualsiasi fonte e include informazioni sull'azione richiesta, la data e l'ora dell'azione, i parametri della richiesta e così via. CloudTrail i file di

registro non sono una traccia ordinata dello stack delle chiamate API pubbliche, quindi non vengono visualizzati in un ordine specifico.

L'esempio seguente mostra una voce di CloudTrail registro che dimostra il. `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"
            }
        }

    ]
}
```

Integrazione CloudWatch con Amazon Polly

Quando interagisci con Amazon Polly, invia le seguenti metriche e dimensioni ogni minuto. CloudWatch Per visualizzare i parametri per Amazon Polly, puoi utilizzare le procedure seguenti.

Puoi monitorare Amazon Polly utilizzando CloudWatch, che raccoglie ed elabora i dati grezzi di Amazon Polly in metriche leggibili quasi in tempo reale. Queste statistiche vengono registrate per un periodo di due settimane, per permettere l'accesso a elementi di tipo `historical information` e per offrire una prospettiva migliore sulle prestazioni del servizio o dell'applicazione Web. Per impostazione predefinita, i dati metrici di Amazon Polly vengono inviati CloudWatch a intervalli di 1 minuto. Per ulteriori informazioni, consulta What Is Amazon CloudWatch nella Amazon CloudWatch User Guide.

Ottenere CloudWatch metriche (console)

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, seleziona Parametri.
3. Nel riquadro CloudWatch Metriche per categoria, nella categoria delle metriche per Amazon Polly, seleziona una categoria di metriche, quindi nel riquadro superiore scorri verso il basso per visualizzare l'elenco completo delle metriche.

Ottenere le metriche su CloudWatch AWS CLI

Il codice seguente consente di visualizzare i parametri disponibili per Amazon Polly.

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

Il comando precedente restituisce un elenco di parametri di Amazon Polly analogo al seguente. L'elemento `MetricName` identifica il parametro.

```
{
    "Metrics": [
        {
            "Namespace": "AWS/Polly",
            "Dimensions": [
```

```
                {
                    "Name": "Operation",
                    "Value": "SynthesizeSpeech"
                }
            ],
            "MetricName": "ResponseLatency"
        },
        {
            "Namespace": "AWS/Polly",
            "Dimensions": [
                {
                    "Name": "Operation",
                    "Value": "SynthesizeSpeech"
                }
            ],
            "MetricName": "RequestCharacters"
        }
```

Per ulteriori informazioni, GetMetricStatisticsconsulta Amazon CloudWatch API Reference.

Parametri di Amazon Polly

Amazon Polly produce i seguenti parametri per ogni richiesta. Queste metriche vengono aggregate e inviate a intervalli di un minuto CloudWatch dove sono disponibili.

Parametro	Descrizione
`RequestCharacters`	Il numero minimo di caratteri della richiesta. Si tratta solo di caratteri fatturabili e non include i tag SSML. Dimensione valida: operazione Statistiche valide: minimo, massimo, medio, somma SampleCount Unità: numero
`ResponseLatency`	La latenza tra il momento in cui è stata effettuata la richiesta e l'inizio della risposta di streaming. Dimensioni valide: operazione

Parametro	Descrizione
	Statistiche valide: minimo, massimo, medio, SampleCount Unità: microsecondi
`2XXCount`	Codice di livello HTTP 200 restituito in caso di risposta riuscita. Dimensioni valide: operazione Statistiche valide: media, SampleCount, somma Unità: numero
`4XXCount`	Codice di errore di livello HTTP 400 restituito in caso di errore. Per ogni risposta riuscita, viene emesso uno zero (0). Dimensioni valide: operazione Statistiche valide: media, SampleCount, somma Unità: numero
`5XXCount`	Codice di errore di livello HTTP 500 restituito in caso di errore. Per ogni risposta riuscita, viene emesso uno zero (0). Dimensioni valide: operazione Statistiche valide: media, SampleCount, somma Unità: numero

Dimensioni per i parametri Amazon Polly

Le metriche di Amazon Polly utilizzano lo spazio dei nomi AWS/Polly e forniscono parametri per la seguente dimensione:

Dimensione	Descrizione
`Operation`	I parametri sono raggruppati in base al metodo API a cui fanno riferimento. I valori possibili sono `SynthesizeSpeech` , `PutLexicon` , `DescribeVoices` e così via.

Informazioni di riferimento delle API Amazon Polly

Amazon Polly è un servizio Web che semplifica la sintesi vocale dal testo.

Il servizio Amazon Polly fornisce operazioni API per la sintesi di voci di alta qualità da testo semplice e Speech Synthesis Markup Language (SSML), oltre a gestire i lessici delle pronunce che consentono di ottenere i migliori risultati per il dominio dell'applicazione.

Le chiamate API autenticate devono essere registrate tramite il processo di firma Signature Version 4. Per ulteriori informazioni, consulta Signing API Requests in. AWSRiferimenti generali di Amazon Web Services

Argomenti

- Azioni
- Tipi di dati
- Errori comuni
- Parametri comuni

Azioni

Sono supportate le operazioni seguenti:

- DeleteLexicon
- DescribeVoices
- GetLexicon
- GetSpeechSynthesisTask
- ListLexicons
- ListSpeechSynthesisTasks
- PutLexicon
- StartSpeechSynthesisTask
- SynthesizeSpeech

DeleteLexicon

Elimina il lessico di pronuncia specificato archiviato in una regione Regione AWS. Un lessico eliminato non è disponibile per la sintesi vocale, né è possibile recuperarlo utilizzando o. `GetLexicon` `ListLexicon` APIs

Per ulteriori informazioni, consulta Managing Lexicons (Gestione di lessici).

Sintassi della richiesta

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

Parametri della richiesta URI

La richiesta utilizza i seguenti parametri URI.

LexiconName

Nome del lessico da eliminare. Deve essere un lessico esistente nella regione.

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

Campo obbligatorio: sì

Corpo della richiesta

La richiesta non ha un corpo della richiesta.

Sintassi della risposta

```
HTTP/1.1 200
```

Elementi di risposta

Se l'operazione riesce, il servizio invia una risposta HTTP 200 con un corpo HTTP vuoto.

Errori

LexiconNotFoundException

Amazon Polly non è in grado di individuare il lessico specificato. Ciò potrebbe essere causato da un lessico mancante, il suo nome è errato o specificando un lessico che si trova in una regione diversa.

Verificare che il lessico esista, sia nella regione (consulta ListLexicons) e che hai digitato il suo nome sia corretto. Allora try again.

Codice di stato HTTP: 404

ServiceFailureException

Una condizione sconosciuta ha causato un errore di servizio.

Codice di stato HTTP: 500

Vedi anche

Per ulteriori informazioni sull'utilizzo di questa API in una delle lingue specifiche, consulta quanto segue AWS SDKs:

- Interfaccia a riga di comando AWS
- AWS SDK per.NET
- AWS SDK per C++
- AWS SDK per Go v2
- AWS SDK per Java V2
- AWS SDK per V3 JavaScript
- AWS SDK per Kotlin
- AWS SDK per PHP V3
- AWS SDK per Python
- AWS SDK per Ruby V3

DescribeVoices

Restituisce l'elenco di voci che sono disponibili per l'uso quando si richiede la sintesi vocale. Ogni voce parla una lingua specificata, è maschile o femminile ed è identificata da un ID, che è la versione ASCII del nome della voce.

Quando si sintetizza il discorso (`SynthesizeSpeech`), si fornisce l'ID vocale per la voce desiderata dall'elenco delle voci restituite da `DescribeVoices`.

Ad esempio, si desidera che l'applicazione per il lettore di notizie legga le notizie in una lingua specifica, ma dando a un utente la possibilità di scegliere la voce. Utilizzando l'operazione `DescribeVoices` è possibile fornire all'utente un elenco di voci disponibili tra cui selezionare.

È anche possibile specificare un codice della lingua per filtrare le voci disponibili. Se ad esempio si specifica `en-US`, l'operazione restituisce un elenco di tutte le voci disponibili per l'inglese Stati Uniti.

Questa operazione necessita delle autorizzazioni a eseguire l'operazione `polly:DescribeVoices`.

Sintassi della richiesta

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

Parametri della richiesta URI

La richiesta utilizza i seguenti parametri URI.

Engine

Speciifica il motore (`standard`,`neural`, `long-form` or`generative`) utilizzato da Amazon Polly durante l'elaborazione del testo di input per la sintesi vocale.

Valori validi: `standard | neural | long-form | generative`

IncludeAdditionalLanguageCodes

Valore booleano che indica se restituire voci bilingue che utilizzano la lingua specificata come lingua aggiuntiva. Ad esempio, se richiedi tutte le lingue che usano l'inglese statunitense (es-US) e c'è una voce italiana che parla sia italiano (it-IT) che inglese americano, tale voce verrà inclusa se specifichi `yes` ma non se si specifica no.

LanguageCode

Il tag di identificazione della lingua (codice ISO 639 per il nome della lingua-codice paese ISO 3166) per filtrare l'elenco delle voci restituite. Se non specifichi questo parametro facoltativo, vengono restituite tutte le voci disponibili.

Valori validi: `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`

NextToken

Un token di impaginazione opaco restituito dal precedente operazione `DescribeVoices`. Se presente, indica dove continuare l'inserzione.

Limitazioni di lunghezza: lunghezza minima di 0. Lunghezza massima di 4096.

Corpo della richiesta

La richiesta non ha un corpo della richiesta.

Sintassi della risposta

```
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" ]
      }
   ]
}
```

Elementi di risposta

Se l'operazione riesce, il servizio restituisce una risposta HTTP 200.

I dati seguenti vengono restituiti in formato JSON mediante il servizio.

NextToken

Il token di impaginazione da utilizzare nella richiesta successiva per continuare l'elenco delle voci. `NextToken` viene restituito solo se la risposta viene troncata.

Tipo: stringa

Limitazioni di lunghezza: lunghezza minima di 0. Lunghezza massima di 4096.

Voices

Un elenco di voci con le loro proprietà.

Tipo: matrice di oggetti Voice

Errori

InvalidNextTokenException

Non NextToken è valido. Verificare che sia stato digitato correttamente, quindi riprovare.

Codice di stato HTTP: 400

ServiceFailureException

Una condizione sconosciuta ha causato un errore di servizio.

Codice di stato HTTP: 500

Vedi anche

Per ulteriori informazioni sull'utilizzo di questa API in una delle lingue specifiche AWS SDKs, consulta quanto segue:

- Interfaccia a riga di comando AWS
- AWS SDK per.NET

- AWS SDK per C++
- AWS SDK per Go v2
- AWS SDK per Java V2
- AWS SDK per V3 JavaScript
- AWS SDK per Kotlin
- AWS SDK per PHP V3
- AWS SDK per Python
- AWS SDK per Ruby V3

GetLexicon

Restituisce i contenuti del lessico di pronuncia specificato archiviato in una regione Regione AWS. Per ulteriori informazioni, consulta Managing Lexicons (Gestione di lessici).

Sintassi della richiesta

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

Parametri della richiesta URI

La richiesta utilizza i seguenti parametri URI.

LexiconName

Nome del lessico.

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

Campo obbligatorio: sì

Corpo della richiesta

La richiesta non ha un corpo della richiesta.

Sintassi della risposta

```
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
    }
}
```

Elementi di risposta

Se l'operazione riesce, il servizio restituisce una risposta HTTP 200.

I dati seguenti vengono restituiti in formato JSON mediante il servizio.

Lexicon

Oggetto lessico che fornisce nome e contenuto della stringa del lessico.

Tipo: oggetto Lexicon

LexiconAttributes

Metadati del lessico, compreso l'alfabetico fonetico utilizzato, il codice della lingua, il lessico ARN, il numero di lessemi definiti nel lessico e la dimensione del lessico in byte.

Tipo: oggetto LexiconAttributes

Errori

LexiconNotFoundException

Amazon Polly non è in grado di individuare il lessico specificato. Ciò potrebbe essere causato da un lessico mancante, il suo nome è errato o specificando un lessico che si trova in una regione diversa.

Verificare che il lessico esista, sia nella regione (consulta ListLexicons) e che hai digitato il suo nome sia corretto. Allora try again.

Codice di stato HTTP: 404

ServiceFailureException

Una condizione sconosciuta ha causato un errore di servizio.

Codice di stato HTTP: 500

Vedi anche

Per ulteriori informazioni sull'utilizzo di questa API in una delle lingue specifiche AWS SDKs, consulta quanto segue:

- Interfaccia a riga di comando AWS
- AWS SDK per.NET
- AWS SDK per C++
- AWS SDK per Go v2
- AWS SDK per Java V2
- AWS SDK per V3 JavaScript
- AWS SDK per Kotlin
- AWS SDK per PHP V3
- AWS SDK per Python
- AWS SDK per Ruby V3

GetSpeechSynthesisTask

Recupera un SpeechSynthesisTask oggetto specifico in base al relativo TaskID. Questo oggetto contiene informazioni sull'attività di sintesi vocale data, incluso lo stato dell'attività, e un collegamento al S3 Bucket contenente l'output dell'attività.

Sintassi della richiesta

```
GET /v1/synthesisTasks/TaskId HTTP/1.1
```

Parametri della richiesta URI

La richiesta utilizza i seguenti parametri URI.

TaskId

Il identificatore unico per il processo di sintesi vocale generato da Amazon Polly.

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

Campo obbligatorio: sì

Corpo della richiesta

La richiesta non ha un corpo della richiesta.

Sintassi della risposta

```
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"
    }
}
```

Elementi di risposta

Se l'operazione riesce, il servizio restituisce una risposta HTTP 200.

I dati seguenti vengono restituiti in formato JSON mediante il servizio.

SynthesisTask

SynthesisTask oggetto che fornisce informazioni sull'attività richiesta, tra cui il formato di output, l'ora di creazione, lo stato dell'attività e così via.

Tipo: oggetto SynthesisTask

Errori

InvalidTaskIdException

L'ID attività fornito non è valido. Fornisci un ID attività valido e riprova.

Codice di stato HTTP: 400

ServiceFailureException

Una condizione sconosciuta ha causato un errore di servizio.

Codice di stato HTTP: 500

SynthesisTaskNotFoundException

Impossibile trovare l'attività Sintesi vocale con ID attività richiesto.

Codice di stato HTTP: 400

Vedi anche

Per ulteriori informazioni sull'utilizzo di questa API in una delle lingue specifiche AWS SDKs, consulta quanto segue:

- Interfaccia a riga di comando AWS
- AWS SDK per.NET
- AWS SDK per C++
- AWS SDK per Go v2
- AWS SDK per Java V2
- AWS SDK per V3 JavaScript
- AWS SDK per Kotlin
- AWS SDK per PHP V3
- AWS SDK per Python
- AWS SDK per Ruby V3

ListLexicons

Restituisce un elenco di lessici di pronuncia archiviati in una regione Regione AWS. Per ulteriori informazioni, consulta Managing Lexicons (Gestione di lessici).

Sintassi della richiesta

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

Parametri della richiesta URI

La richiesta utilizza i seguenti parametri URI.

NextToken

Un token di impaginazione opaco restituito dal precedente operazione `ListLexicons`. Se presente, indica dove continuare l'elenco dei lessiconi.

Limitazioni di lunghezza: lunghezza minima di 0. Lunghezza massima di 4096.

Corpo della richiesta

La richiesta non ha un corpo della richiesta.

Sintassi della risposta

```
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"
}
```

Elementi di risposta

Se l'operazione riesce, il servizio restituisce una risposta HTTP 200.

I dati seguenti vengono restituiti in formato JSON mediante il servizio.

Lexicons

Un elenco di nomi e attributi del lessico.

Tipo: matrice di oggetti LexiconDescription

NextToken

Il token di impaginazione da utilizzare nella richiesta successiva per continuare l'elenco dei lessici. `NextToken` viene restituito solo se la risposta viene troncata.

Tipo: stringa

Limitazioni di lunghezza: lunghezza minima di 0. Lunghezza massima di 4096.

Errori

InvalidNextTokenException

Non NextToken è valido. Verificare che sia stato digitato correttamente, quindi riprovare.

Codice di stato HTTP: 400

ServiceFailureException

Una condizione sconosciuta ha causato un errore di servizio.

Codice di stato HTTP: 500

Vedi anche

Per ulteriori informazioni sull'utilizzo di questa API in una delle lingue specifiche AWS SDKs, consulta quanto segue:

- Interfaccia a riga di comando AWS
- AWS SDK per.NET
- AWS SDK per C++
- AWS SDK per Go v2
- AWS SDK per Java V2
- AWS SDK per V3 JavaScript
- AWS SDK per Kotlin
- AWS SDK per PHP V3
- AWS SDK per Python
- AWS SDK per Ruby V3

ListSpeechSynthesisTasks

Restituisce un elenco di SpeechSynthesisTask oggetti ordinati in base alla data di creazione. Questa operazione consente di filtrare le attività in base al relativo stato, ad esempio consentendo agli utenti di elencare solo le attività completate.

Sintassi della richiesta

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

Parametri della richiesta URI

La richiesta utilizza i seguenti parametri URI.

MaxResults

Numero massimo di attività di sintesi vocale restituite in un'operazione List.

Intervallo valido: valore minimo di 1. valore massimo pari a 100.

NextToken

Il token di impaginazione da utilizzare nella richiesta successiva per continuare l'elenco delle attività di sintesi vocale.

Limitazioni di lunghezza: lunghezza minima di 0. Lunghezza massima di 4096.

Status

Stato delle attività di sintesi vocale restituite in un'operazione List

Valori validi: `scheduled | inProgress | completed | failed`

Corpo della richiesta

La richiesta non ha un corpo della richiesta.

Sintassi della risposta

```
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"
      }
   ]
}
```

Elementi di risposta

Se l'operazione riesce, il servizio restituisce una risposta HTTP 200.

I dati seguenti vengono restituiti in formato JSON mediante il servizio.

NextToken

Un token di impaginazione opaco restituito dall'operazione List precedente in questa richiesta. Se presente, indica dove continuare l'inserzione.

Tipo: stringa

Limitazioni di lunghezza: lunghezza minima di 0. Lunghezza massima di 4096.

SynthesisTasks

Elenco di SynthesisTask oggetti che fornisce informazioni sull'attività specificata nella richiesta di elenco, inclusi il formato di output, l'ora di creazione, lo stato dell'attività e così via.

Tipo: matrice di oggetti SynthesisTask

Errori

InvalidNextTokenException

Non NextToken è valido. Verificare che sia stato digitato correttamente, quindi riprovare.

Codice di stato HTTP: 400

ServiceFailureException

Una condizione sconosciuta ha causato un errore di servizio.

Codice di stato HTTP: 500

Vedi anche

Per ulteriori informazioni sull'utilizzo di questa API in una delle lingue specifiche AWS SDKs, consulta quanto segue:

- Interfaccia a riga di comando AWS
- AWS SDK per.NET
- AWS SDK per C++
- AWS SDK per Go v2
- AWS SDK per Java V2
- AWS SDK per V3 JavaScript
- AWS SDK per Kotlin
- AWS SDK per PHP V3
- AWS SDK per Python
- AWS SDK per Ruby V3

PutLexicon

Archivia un lessico di pronuncia in una regione Regione AWS. Se un lessico con lo stesso nome esiste già nella regione, viene sovrascritto dal nuovo lessico. Le operazioni del lessico hanno una certa coerenza, pertanto potrebbe essere necessario del tempo prima che il lessico sia disponibile per l'operazione. SynthesizeSpeech

Per ulteriori informazioni, consulta Managing Lexicons (Gestione di lessici).

Sintassi della richiesta

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

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

Parametri della richiesta URI

La richiesta utilizza i seguenti parametri URI.

LexiconName

Nome del lessico. Il nome deve seguire il formato espresso normale [0-9A-Za-z]{1,20}. Cioè, il nome è una stringa alfanumerica con distinzione tra maiuscole e minuscole da 20 caratteri.

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

Campo obbligatorio: sì

Corpo della richiesta

La richiesta accetta i seguenti dati in formato JSON.

Content

Contenuto del lessico PLS come dati stringa.

Tipo: stringa

Campo obbligatorio: sì

Sintassi della risposta

```
HTTP/1.1 200
```

Elementi di risposta

Se l'operazione riesce, il servizio invia una risposta HTTP 200 con un corpo HTTP vuoto.

Errori

InvalidLexiconException

Amazon Polly non è in grado di individuare il lessico specificato. Verificare che il nome del lessico sia stato digitato correttamente, quindi riprovare.

Codice di stato HTTP: 400

LexiconSizeExceededException

La dimensione massima del lessico specificato verrebbe superata da questa operazione.

Codice di stato HTTP: 400

MaxLexemeLengthExceededException

La dimensione massima del lessema verrebbe superata da questa operazione.

Codice di stato HTTP: 400

MaxLexiconsNumberExceededException

Il numero massimo di lessici verrebbe superato da questa operazione.

Codice di stato HTTP: 400

ServiceFailureException

Una condizione sconosciuta ha causato un errore di servizio.

Codice di stato HTTP: 500

UnsupportedPlsAlphabetException

L'alfabeto specificato dal lessico non è un alfabeto supportato. I valori validi sono `x-sampa` e `ipa`.

Codice di stato HTTP: 400

UnsupportedPlsLanguageException

La lingua specificata nel lessico non è supportata. Per un elenco di linguaggi supportati, consulta Lexicon Attributes (Attributi del lessico).

Codice di stato HTTP: 400

Vedi anche

Per ulteriori informazioni sull'utilizzo di questa API in una delle lingue AWS SDKs specifiche, consulta quanto segue:

- Interfaccia a riga di comando AWS
- AWS SDK per.NET
- AWS SDK per C++
- AWS SDK per Go v2
- AWS SDK per Java V2
- AWS SDK per V3 JavaScript
- AWS SDK per Kotlin
- AWS SDK per PHP V3
- AWS SDK per Python
- AWS SDK per Ruby V3

StartSpeechSynthesisTask

Consente la creazione di un'attività di sintesi asincrona, avviando un nuovo `SpeechSynthesisTask`. Questa operazione richiede tutte le informazioni standard necessarie per la sintesi vocale, oltre al nome di un bucket Amazon S3 affinché il servizio archivi l'output dell'attività di sintesi e due parametri opzionali (`OutputS3KeyPrefix` e `SnsTopicArn`). Una volta creata l'attività di sintesi, questa operazione restituirà un oggetto `SpeechSynthesisTask`, che includerà un identificatore di questa attività e lo stato corrente. L'oggetto `SpeechSynthesisTask` è disponibile per 72 ore dopo l'avvio dell'attività di sintesi asincrona.

Sintassi della richiesta

```
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"
}
```

Parametri della richiesta URI:

La richiesta non utilizza parametri URI.

Corpo della richiesta

La richiesta accetta i seguenti dati in formato JSON.

Engine

Speciifica il motore (`standard`,`neural`, `long-form` or`generative`) per Amazon Polly da utilizzare durante l'elaborazione del testo di input per la sintesi vocale. Se si utilizza una voce non supportata per il motore selezionato, si verificherà un errore.

Tipo: stringa

Valori validi: `standard | neural | long-form | generative`

Campo obbligatorio: no

LanguageCode

Codice lingua facoltativo per la richiesta di sintesi vocale. Ciò è necessario solo se si utilizza una voce bilingue, come Aditi, che può essere utilizzata sia per l'inglese indiano (en-IN) che per l'hindi (hi-IN).

Se viene utilizzata una voce bilingue e non viene specificato alcun codice lingua, Amazon Polly utilizza la lingua predefinita della voce bilingue. La lingua predefinita per ogni voce è quella restituita dall'DescribeVoicesoperazione per il `LanguageCode` parametro. Ad esempio, se non viene specificato alcun codice di lingua, Aditi utilizzerà l'inglese indiano anziché l'hindi.

Tipo: stringa

Valori validi: `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`

Campo obbligatorio: no

LexiconNames

Elenco di uno o più nomi di lessico di pronuncia che si desidera applicare il servizio durante la sintesi. I lessici vengono applicati solo se la lingua del lessico corrisponde alla lingua della voce.

Tipo: matrice di stringhe

Membri della matrice: numero massimo di 5 elementi.

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

Campo obbligatorio: no

OutputFormat

Il formato in cui verrà codificato l'output restituito. Per il flusso audio, questo sarà mp3, ogg_vorbis, o pcm. Per i segni vocali, questo sarà json.

Tipo: stringa

Valori validi: `json | mp3 | ogg_vorbis | pcm`

Campo obbligatorio: sì

OutputS3BucketName

Nome del bucket Amazon S3 in cui viene salvato il file di output.

Tipo: stringa

Modello: `^[a-z0-9][\.\-a-z0-9]{1,61}[a-z0-9]$`

Campo obbligatorio: sì

OutputS3KeyPrefix

Prefisso della chiave Amazon S3 per il file vocale di output.

Tipo: stringa

Modello: `^[0-9a-zA-Z\/\!\-_\.\*\'\(\):;\$@=+\,\?&]{0,800}$`

Campo obbligatorio: no

SampleRate

La frequenza audio specificata in Hz.

I valori validi per mp3 e ogg_vorbis sono "8000", "16000", "22050" e "24000". Il valore predefinito per le voci standard è "22050". Il valore predefinito per le voci neurali è "24000". Il valore predefinito per le voci lunghe è «24000". Il valore predefinito per le voci generative è «24000".

I valori validi per pcm sono "8000" e "16000". Il valore predefinito è "16000".

Tipo: string

Campo obbligatorio: no

SnsTopicArn

ARN per l'argomento SNS utilizzato facoltativamente per la notifica di stato per un'attività di sintesi vocale.

Tipo: stringa

Modello: `^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)$`

Campo obbligatorio: no

SpeechMarkTypes

Tipo di segni vocali restituiti per il testo di input.

Tipo: matrice di stringhe

Membri della matrice: numero massimo di 4 elementi.

Valori validi: `sentence | ssml | viseme | word`

Campo obbligatorio: no

Text

Testo di input da sintetizzare. Se specificate ssml come TextType, seguite il formato SSML per il testo di input.

Tipo: stringa

Campo obbligatorio: sì

TextType

Specifica se il testo di input è testo normale o SSML. Il valore di default è testo normale.

Tipo: stringa

Valori validi: `ssml | text`

Campo obbligatorio: no

VoiceId

ID vocale da utilizzare per la sintesi.

Tipo: stringa

Valori validi: `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`

Campo obbligatorio: sì

Sintassi della risposta

```
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"
    }
}
```

Elementi di risposta

Se l'operazione riesce, il servizio restituisce una risposta HTTP 200.

I dati seguenti vengono restituiti in formato JSON mediante il servizio.

SynthesisTask

SynthesisTask oggetto che fornisce informazioni e attributi su un compito di sintesi vocale appena inviato.

Tipo: oggetto SynthesisTask

Errori

EngineNotSupportedException

Questo motore non è compatibile con la voce designata. Scegli una nuova voce compatibile con il motore o cambia il motore e riavvia l'operazione.

Codice di stato HTTP: 400

InvalidS3BucketException

Il nome del bucket Amazon S3 non è valido. Controlla il tuo input con i requisiti di denominazione del S3 Bucket e riprova.

Codice di stato HTTP: 400

InvalidS3KeyException

Il prefisso della chiave Amazon S3 non è valido. Fornisci un nome chiave oggetto S3 valido.

Codice di stato HTTP: 400

InvalidSampleRateException

La frequenza di campionamento specificata non è valida.

Codice di stato HTTP: 400

InvalidSnsTopicArnException

L'ARN dell'argomento SNS fornito non è valido. Fornisci un ARN di argomento SNS valido e riprova.

Codice di stato HTTP: 400

InvalidSsmlException

L'SSML fornito non è valido. Verifica la sintassi SSML, l'ortografia dei tag e dei valori, quindi riprova.

Codice di stato HTTP: 400

LanguageNotSupportedException

La lingua specificata non è attualmente supportata da Amazon Polly in questa capacità.

Codice di stato HTTP: 400

LexiconNotFoundException

Amazon Polly non è in grado di individuare il lessico specificato. Ciò potrebbe essere causato da un lessico mancante, il suo nome è errato o specificando un lessico che si trova in una regione diversa.

Verificare che il lessico esista, sia nella regione (consulta ListLexicons) e che hai digitato il suo nome sia corretto. Allora try again.

Codice di stato HTTP: 404

MarksNotSupportedForFormatException

I contrassegni vocali non sono supportati per `OutputFormat` selezionato. I contrassegni vocali sono disponibili solo per il contenuto in formato `json`.

Codice di stato HTTP: 400

ServiceFailureException

Una condizione sconosciuta ha causato un errore di servizio.

Codice di stato HTTP: 500

SsmlMarksNotSupportedForTextTypeException

I contrassegni vocali SSML non sono supportati per l'input di tipo testo normale.

Codice di stato HTTP: 400

TextLengthExceededException

Il valore del parametro "Testo" è più lungo dei limiti accettati. Per il API `SynthesizeSpeech`, il limite per il testo di input è un massimo di 6000 caratteri totali, di cui non più di 3000 caratteri possono essere fatturati. Per il API `StartSpeechSynthesisTask`, il massimo è 200.000 caratteri, di cui non più di 100.000 caratteri possono essere fatturati. I tag SSML non vengono conteggiati come caratteri fatturati.

Codice di stato HTTP: 400

Vedi anche

Per ulteriori informazioni sull'utilizzo di questa API in una delle lingue specifiche AWS SDKs, consulta quanto segue:

- Interfaccia a riga di comando AWS
- AWS SDK per.NET
- AWS SDK per C++
- AWS SDK per Go v2
- AWS SDK per Java V2
- AWS SDK per V3 JavaScript
- AWS SDK per Kotlin
- AWS SDK per PHP V3
- AWS SDK per Python
- AWS SDK per Ruby V3

SynthesizeSpeech

Sintetizza input UTF-8, testo normale o SSML in un flusso di byte. L'input SSML deve essere valido e ben formato SSML. Alcuni alfabeti potrebbero non essere disponibili con tutte le voci (ad esempio, il cirillico potrebbe non essere letto affatto dalle voci inglesi) a meno che non venga utilizzata la mappatura dei fonemi. Per ulteriori informazioni, consulta How it Works (Come funziona).

Sintassi della richiesta

```
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"
}
```

Parametri della richiesta URI:

La richiesta non utilizza parametri URI.

Corpo della richiesta

La richiesta accetta i seguenti dati in formato JSON.

Engine

Speciifica il motore (`standard`, `neurallong-form`, `ogenerative`) per Amazon Polly da utilizzare durante l'elaborazione del testo di input per la sintesi vocale. Fornisci un motore supportato dalla voce selezionata. Se non fornisci un motore, per impostazione predefinita viene selezionato il motore standard. Se una voce selezionata non è supportata dal motore standard, verrà generato un errore. Per informazioni sulle voci di Amazon Polly e sulle voci disponibili per ogni motore, consulta Available Voices.

Tipo: stringa

Valori validi: `standard | neural | long-form | generative`

Campo obbligatorio: no

LanguageCode

Codice della lingua facoltativo per la richiesta di sintesi vocale. Ciò è necessario solo se si utilizza una voce bilingue, come Aditi, che può essere utilizzata sia per l'inglese indiano (en-IN) che per l'hindi (hi-IN).

Se viene utilizzata una voce bilingue e non viene specificato alcun codice lingua, Amazon Polly utilizza la lingua predefinita della voce bilingue. La lingua predefinita per ogni voce è quella restituita dall'DescribeVoicesoperazione per il `LanguageCode` parametro. Ad esempio, se non viene specificato alcun codice di lingua, Aditi utilizzerà l'inglese indiano anziché l'hindi.

Tipo: stringa

Valori validi: `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`

Campo obbligatorio: no

LexiconNames

Elenco di uno o più nomi di lessico di pronuncia che si desidera applicare il servizio durante la sintesi. I lessici vengono applicati solo se la lingua del lessico corrisponde alla lingua della voce. Per informazioni sulla memorizzazione dei lessici, vedere PutLexicon.

Tipo: matrice di stringhe

Membri della matrice: numero massimo di 5 elementi.

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

Campo obbligatorio: no

OutputFormat

Il formato in cui verrà codificato l'output restituito. Per il flusso audio, questo sarà mp3, ogg_vorbis, o pcm. Per i segni vocali, questo sarà json.

Quando si usa pcm, il contenuto restituito è audio/pcm in un formato con firma a 16 bit, 1 canale (mono), little-endian.

Tipo: stringa

Valori validi: `json | mp3 | ogg_vorbis | pcm`

Campo obbligatorio: sì

SampleRate

La frequenza audio specificata in Hz.

I valori validi per mp3 e ogg_vorbis sono «8000", «16000", «22050", «24000", «44100" e «48000". Il valore predefinito per le voci standard è "22050". Il valore predefinito per le voci neurali è "24000". Il valore predefinito per le voci lunghe è «24000". Il valore predefinito per le voci generative è «24000".

I valori validi per pcm sono "8000" e "16000". Il valore predefinito è "16000".

Tipo: string

Campo obbligatorio: no

SpeechMarkTypes

Tipo di segni vocali restituiti per il testo di input.

Tipo: matrice di stringhe

Membri della matrice: numero massimo di 4 elementi.

Valori validi: `sentence | ssml | viseme | word`

Campo obbligatorio: no

Text

Inserisci testo da sintetizzare. Se si specifica `ssml` come `TextType`, seguire il formato SSML per il testo di input.

Tipo: stringa

Campo obbligatorio: sì

TextType

Specifica se il testo di input è testo normale o SSML. Il valore di default è testo normale. Per ulteriori informazioni, consulta Using SSML (Utilizzo di SSML).

Tipo: stringa

Valori validi: `ssml | text`

Campo obbligatorio: no

VoiceId

ID vocale da utilizzare per la sintesi. È possibile ottenere un elenco delle voci disponibili IDs chiamando l'operazione. DescribeVoices

Tipo: stringa

Valori validi: `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`

Campo obbligatorio: sì

Sintassi della risposta

```
HTTP/1.1 200
Content-Type: ContentType
x-amzn-RequestCharacters: RequestCharacters

AudioStream
```

Elementi di risposta

Se l'operazione riesce, il servizio restituisce una risposta HTTP 200.

La risposta restituisce le seguenti intestazioni HTTP.

ContentType

Specifica il tipo di flusso audio. Questo dovrebbe riflettere il parametro `OutputFormat` nella richiesta.

- Se richiedi `mp3` come `OutputFormat`, il `ContentType` restituito è audio/mpeg.
- Se richiedi `ogg_vorbis` come `OutputFormat`, il `ContentType` restituito è audio/ogg.
- Se lo richiedi `pcm` come`OutputFormat`, il risultato `ContentType` restituito è audio/pcm in un formato con firma a 16 bit, 1 canale (mono), little-endian.
- Se lo richiedi `json` come, il risultato restituito è application/`OutputFormat`. `ContentType` x-json-stream

RequestCharacters

Numero di caratteri sintetizzati.

La risposta restituisce quanto segue come corpo HTTP.

AudioStream

Flusso contenente il discorso sintetizzato.

Errori

EngineNotSupportedException

Questo motore non è compatibile con la voce designata. Scegli una nuova voce compatibile con il motore o cambia il motore e riavvia l'operazione.

Codice di stato HTTP: 400

InvalidSampleRateException

La frequenza di campionamento specificata non è valida.

Codice di stato HTTP: 400

InvalidSsmlException

L'SSML fornito non è valido. Verifica la sintassi SSML, l'ortografia dei tag e dei valori, quindi riprova.

Codice di stato HTTP: 400

LanguageNotSupportedException

La lingua specificata non è attualmente supportata da Amazon Polly in questa capacità.

Codice di stato HTTP: 400

LexiconNotFoundException

Amazon Polly non è in grado di individuare il lessico specificato. Ciò potrebbe essere causato da un lessico mancante, il suo nome è errato o specificando un lessico che si trova in una regione diversa.

Verificare che il lessico esista, sia nella regione (consulta ListLexicons) e che hai digitato il suo nome sia corretto. Allora try again.

Codice di stato HTTP: 404

MarksNotSupportedForFormatException

I contrassegni vocali non sono supportati per `OutputFormat` selezionato. I contrassegni vocali sono disponibili solo per il contenuto in formato `json`.

Codice di stato HTTP: 400

ServiceFailureException

Una condizione sconosciuta ha causato un errore di servizio.

Codice di stato HTTP: 500

SsmlMarksNotSupportedForTextTypeException

I contrassegni vocali SSML non sono supportati per l'input di tipo testo normale.

Codice di stato HTTP: 400

TextLengthExceededException

Il valore del parametro "Testo" è più lungo dei limiti accettati. Per il API `SynthesizeSpeech`, il limite per il testo di input è un massimo di 6000 caratteri totali, di cui non più di 3000 caratteri

possono essere fatturati. Per il API `StartSpeechSynthesisTask`, il massimo è 200.000 caratteri, di cui non più di 100.000 caratteri possono essere fatturati. I tag SSML non vengono conteggiati come caratteri fatturati.

Codice di stato HTTP: 400

Vedi anche

Per ulteriori informazioni sull'utilizzo di questa API in una delle lingue specifiche, consulta quanto segue AWS SDKs:

- Interfaccia a riga di comando AWS
- AWS SDK per.NET
- AWS SDK per C++
- AWS SDK per Go v2
- AWS SDK per Java V2
- AWS SDK per V3 JavaScript
- AWS SDK per Kotlin
- AWS SDK per PHP V3
- AWS SDK per Python
- AWS SDK per Ruby V3

Tipi di dati

Sono supportati i tipi di dati seguenti:

- Lexicon
- LexiconAttributes
- LexiconDescription
- SynthesisTask
- Voice

Lexicon

Fornisce il nome del lessico e il contenuto del lessico in formato stringa. Per ulteriori informazioni, consulta Pronunciation Lexicon Specification (PLS) Version 1.0.

Indice

Content

Contenuto lessico in formato stringa. Il contenuto di un lessico deve essere in formato PLS.

Tipo: string

Campo obbligatorio: no

Name

Nome del lessico.

Tipo: stringa

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

Campo obbligatorio: no

Vedi anche

Per ulteriori informazioni sull'utilizzo di questa API in una delle lingue specifiche AWS SDKs, consulta quanto segue:

- AWS SDK per C++
- AWS SDK per Java V2
- AWS SDK per Ruby V3

LexiconAttributes

Contiene metadati che descrivono il lessico, ad esempio il numero di lessemi, il codice della lingua e così via. Per ulteriori informazioni, consulta Managing Lexicons (Gestione di lessici).

Indice

Alphabet

Alfabeto fonetico usato nel lessico. I valori validi sono `ipa` e `x-sampa`.

Tipo: string

Campo obbligatorio: no

LanguageCode

Codice della lingua a cui si applica il lessico. Un lessico con un codice di lingua come "en" verrebbe applicato a tutte le lingue inglesi (en-GB, en-US, en-AUS, en-WLS e così via.

Tipo: stringa

Valori validi: `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`

Campo obbligatorio: no

LastModified

Data dell'ultima modifica del lessico (un valore timestamp).

Tipo: Timestamp

Campo obbligatorio: no

LexemesCount

Numero di lessemi nel lessico.

Tipo: integer

Campo obbligatorio: no

LexiconArn

Amazon Resource Name (ARN) del lessico.

Tipo: string

Campo obbligatorio: no

Size

Dimensione totale del lessico, espressa in caratteri.

Tipo: integer

Campo obbligatorio: no

Vedi anche

Per ulteriori informazioni sull'utilizzo di questa API in una delle lingue specifiche AWS SDKs, consulta quanto segue:

- AWS SDK per C++
- AWS SDK per Java V2
- AWS SDK per Ruby V3

LexiconDescription

Descrive i contenuti del lessico.

Indice

Attributes

Fornisce i metadati del lessico.

Tipo: oggetto LexiconAttributes

Campo obbligatorio: no

Name

Nome del lessico.

Tipo: stringa

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

Campo obbligatorio: no

Vedi anche

Per ulteriori informazioni sull'utilizzo di questa API in una delle lingue specifiche AWS SDKs, consulta quanto segue:

- AWS SDK per C++
- AWS SDK per Java V2
- AWS SDK per Ruby V3

SynthesisTask

SynthesisTask oggetto che fornisce informazioni su un'attività di sintesi vocale.

Indice

CreationTime

Indicatore orario per l'ora in cui è stato avviato il processo di sintesi.

Tipo: Timestamp

Campo obbligatorio: no

Engine

Speciifica il motore (`standard`,`neural`, `long-form` or`generative`) per Amazon Polly da utilizzare durante l'elaborazione del testo di input per la sintesi vocale. Se si utilizza una voce non supportata per il motore selezionato, si verificherà un errore.

Tipo: stringa

Valori validi: `standard | neural | long-form | generative`

Campo obbligatorio: no

LanguageCode

Codice lingua facoltativo per un processo di sintesi. Ciò è necessario solo se si utilizza una voce bilingue, come Aditi, che può essere utilizzata sia per l'inglese indiano (en-IN) che per l'hindi (hi-IN).

Se viene utilizzata una voce bilingue e non viene specificato alcun codice lingua, Amazon Polly utilizza la lingua predefinita della voce bilingue. La lingua predefinita per ogni voce è quella restituita dall'DescribeVoicesoperazione per il `LanguageCode` parametro. Ad esempio, se non viene specificato alcun codice di lingua, Aditi utilizzerà l'inglese indiano anziché l'hindi.

Tipo: stringa

Valori validi: `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`

Campo obbligatorio: no

LexiconNames

Elenco di uno o più nomi di lessico di pronuncia che si desidera applicare il servizio durante la sintesi. I lessici vengono applicati solo se la lingua del lessico corrisponde alla lingua della voce.

Tipo: matrice di stringhe

Membri della matrice: numero massimo di 5 elementi.

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

Campo obbligatorio: no

OutputFormat

Il formato in cui verrà codificato l'output restituito. Per il flusso audio, questo sarà mp3, ogg_vorbis, o pcm. Per i segni vocali, questo sarà json.

Tipo: stringa

Valori validi: `json | mp3 | ogg_vorbis | pcm`

Campo obbligatorio: no

OutputUri

Percorso per il file di sintesi vocale di output.

Tipo: string

Campo obbligatorio: no

RequestCharacters

Numero di caratteri fatturabili sintetizzati.

Tipo: integer

Campo obbligatorio: no

SampleRate

La frequenza audio specificata in Hz.

I valori validi per mp3 e ogg_vorbis sono "8000", "16000", "22050" e "24000". Il valore predefinito per le voci standard è "22050". Il valore predefinito per le voci neurali è "24000". Il valore predefinito per le voci lunghe è «24000". Il valore predefinito per le voci generative è «24000".

I valori validi per pcm sono "8000" e "16000". Il valore predefinito è "16000".

Tipo: string

Campo obbligatorio: no

SnsTopicArn

ARN per l'argomento SNS utilizzato facoltativamente per la notifica di stato per un'attività di sintesi vocale.

Tipo: stringa

Modello: `^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)$`

Campo obbligatorio: no

SpeechMarkTypes

Tipo di segni vocali restituiti per il testo di input.

Tipo: matrice di stringhe

Membri della matrice: numero massimo di 4 elementi.

Valori validi: `sentence | ssml | viseme | word`

Campo obbligatorio: no

TaskId

Il identificatore unico per l'attività di sintesi vocale generato da Amazon Polly.

Tipo: stringa

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

Campo obbligatorio: no

TaskStatus

Stato corrente dell'attività di sintesi vocale individuale.

Tipo: stringa

Valori validi: `scheduled | inProgress | completed | failed`

Campo obbligatorio: no

TaskStatusReason

Motivo dello stato corrente di un'attività di sintesi vocale specifica, inclusi gli errori se l'attività non è riuscita.

Tipo: string

Campo obbligatorio: no

TextType

Specifica se il testo di input è testo normale o SSML. Il valore di default è testo normale.

Tipo: stringa

Valori validi: `ssml | text`

Campo obbligatorio: no

VoiceId

ID vocale da utilizzare per la sintesi.

Tipo: stringa

Valori validi: `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
```

Campo obbligatorio: no

Vedi anche

Per ulteriori informazioni sull'utilizzo di questa API in una delle lingue AWS SDKs specifiche, consulta quanto segue:

- AWS SDK per C++
- AWS SDK per Java V2
- AWS SDK per Ruby V3

Voice

Descrizione del voce.

Indice

AdditionalLanguageCodes

Codici aggiuntivi per le lingue disponibili per la voce specificata oltre alla lingua predefinita.

Ad esempio, la lingua predefinita per Aditi è l'inglese indiano (en-IN) perché è stato utilizzato per la prima volta per quella lingua. Poiché Aditi è bilingue e fluente in inglese indiano e hindi, questo parametro mostrerebbe il codice `hi-IN`.

Tipo: matrice di stringhe

Valori validi: `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`

Campo obbligatorio: no

Gender

Sesso della voce.

Tipo: stringa

Valori validi: `Female | Male`

Campo obbligatorio: no

Id

Amazon Polly ha assegnato l'ID della voce. Questo è l'ID che si specifica quando si chiama la operazione `SynthesizeSpeech`.

Tipo: stringa

Valori validi: `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`

Campo obbligatorio: no

LanguageCode

Codice della lingua della voce.

Tipo: stringa

Valori validi: `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`

Campo obbligatorio: no

LanguageName

Nome leggibile della lingua in inglese.

Tipo: string

Campo obbligatorio: no

Name

Nome della voce (ad esempio, Salli, Kendra, ecc.). Questo fornisce un nome della voce leggibile dall'utente che potrebbe essere visualizzato nell'applicazione.

Tipo: string

Campo obbligatorio: no

SupportedEngines

Speciifica quali motori (`standard`,`neural`, `long-form` or`generative`) sono supportati da una determinata voce.

Tipo: matrice di stringhe

Valori validi: `standard | neural | long-form | generative`

Campo obbligatorio: no

Vedi anche

Per ulteriori informazioni sull'utilizzo di questa API in una delle lingue specifiche AWS SDKs, consulta quanto segue:

- AWS SDK per C++
- AWS SDK per Java V2
- AWS SDK per Ruby V3

Errori comuni

Questa sezione elenca gli errori comuni alle azioni API di tutti i AWS servizi. Per gli errori specifici di un'azione API per questo servizio, consulta l'argomento per quell'azione API.

AccessDeniedException

Non disponi dell'autorizzazione di accesso sufficiente per eseguire questa operazione.

Codice di stato HTTP: 400

IncompleteSignature

La firma della richiesta non è conforme agli AWS standard.

Codice di stato HTTP: 400

InternalFailure

L'elaborazione della richiesta non è riuscita a causa di un errore, un'eccezione o un guasto interno sconosciuto.

Codice di stato HTTP: 500

InvalidAction

L'azione o l'operazione richiesta non è valida. Verifica che l'operazione sia digitata correttamente.

Codice di stato HTTP: 400

InvalidClientTokenId

Il certificato X.509 o AWS l'ID della chiave di accesso fornito non esiste nei nostri archivi.

Codice di stato HTTP: 403

NotAuthorized

Non disponi delle autorizzazioni per eseguire questa azione.

Codice di stato HTTP: 400

OptInRequired

L'ID della chiave di AWS accesso richiede un abbonamento al servizio.

Codice di stato HTTP: 403

RequestExpired

La richiesta ha raggiunto il servizio più di 15 minuti dopo la data indicata sulla richiesta o più di 15 minuti dopo la data di scadenza della richiesta (ad esempio nel caso di pre-firmata URLs), oppure la data indicata sulla richiesta è tra più di 15 minuti.

Codice di stato HTTP: 400

ServiceUnavailable

La richiesta non è riuscita a causa di un errore temporaneo del server.

Codice di stato HTTP: 503

ThrottlingException

La richiesta è stata negata a causa del throttling della richiesta.

Codice di stato HTTP: 400

ValidationError

L'input non soddisfa i vincoli specificati da un servizio. AWS

Codice di stato HTTP: 400

Parametri comuni

L'elenco seguente contiene i parametri utilizzati da tutte le azioni per firmare le richieste di Signature Version 4 con una stringa di query. Qualsiasi parametro specifico di un'operazione è riportato nell'argomento relativo all'operazione. Per ulteriori informazioni sulla versione 4 di Signature, consulta Signing AWS API requests nella IAM User Guide.

Action

azione da eseguire.

Tipo: stringa

Campo obbligatorio: sì

Version

La versione dell'API per cui è scritta la richiesta, espressa nel formato YYYY-MM-DD.

Tipo: stringa

Campo obbligatorio: sì

X-Amz-Algorithm

Algoritmo hash utilizzato per creare la firma della richiesta.

Condition: specifica questo parametro quando includi le informazioni di autenticazione in una stringa di query anziché nell'intestazione di autorizzazione HTTP.

Tipo: stringa

Valori validi: `AWS4-HMAC-SHA256`

Obbligatorio: condizionale

X-Amz-Credential

Il valore dell'ambito delle credenziali, che è una stringa che include la chiave di accesso, la data, la regione di destinazione, il servizio richiesto e una stringa di terminazione ("aws4_request"). Il valore viene espresso nel seguente formato: chiave_accesso/AAAAMMGG/regione/servizio/aws4_request.

Per ulteriori informazioni, consulta Creare una richiesta AWS API firmata nella Guida per l'utente IAM.

Condition: specifica questo parametro quando includi le informazioni di autenticazione in una stringa di query anziché nell'intestazione di autorizzazione HTTP.

Tipo: stringa

Obbligatorio: condizionale

X-Amz-Date

La data utilizzata per creare la firma. Il formato deve essere il formato di base ISO 8601 (YYYYMMDD'T'HHMMSS'Z'). Per esempio, la data e l'ora seguenti è un X-Amz-Date valore valido:`20120325T120000Z`.

Condizione: X-Amz-Date è facoltativa per tutte le richieste; può essere utilizzata per sovrascrivere la data utilizzata per firmare le richieste. Se l'intestazione Date è specificata nel formato base ISO 8601, non X-Amz-Date è obbligatoria. Quando X-Amz-Date viene utilizzata, sovrascrive sempre il valore dell'intestazione Date. Per ulteriori informazioni, consulta Elementi di una firma di richiesta AWS API nella Guida per l'utente IAM.

Tipo: stringa

Obbligatorio: condizionale

X-Amz-Security-Token

Il token di sicurezza temporaneo ottenuto tramite una chiamata a AWS Security Token Service (AWS STS). Per un elenco di servizi che supportano le credenziali di sicurezza temporanee da AWS STS, consulta la pagina Servizi AWS che funzionano con IAM nella Guida per l'utente di IAM.

Condizione: se utilizzi credenziali di sicurezza temporanee di AWS STS, devi includere il token di sicurezza.

Tipo: stringa

Obbligatorio: condizionale

X-Amz-Signature

Specifica la firma con codifica esadecimale calcolata dalla stringa da firmare e dalla chiave di firma derivata.

Condition: specifica questo parametro quando includi le informazioni di autenticazione in una stringa di query anziché nell'intestazione di autorizzazione HTTP.

Tipo: stringa

Obbligatorio: condizionale

X-Amz-SignedHeaders

Specifica tutte le intestazioni HTTP incluse come parte della richiesta canonica. Per ulteriori informazioni sulla specificazione delle intestazioni firmate, consulta Creare una richiesta AWS API firmata nella Guida per l'utente IAM.

Condition: specifica questo parametro quando includi le informazioni di autenticazione in una stringa di query anziché nell'intestazione di autorizzazione HTTP.

Tipo: stringa

Obbligatorio: condizionale

Cronologia dei documenti per Amazon Polly

La tabella che segue descrive le modifiche importanti apportate a ogni release della Amazon Polly Developer Guide (Guida per sviluppatori di Amazon Polly). Per ricevere notifiche sugli aggiornamenti di questa documentazione, è possibile sottoscrivere un feed RSS.

- Ultimo aggiornamento della documentazione: 18 febbraio 2025

Modifica	Descrizione	Data
Nuova regione aggiunta per le voci neurali e standard	Amazon Polly è ora disponibile nella regione Asia Pacifico (Malesia). AWS Questa regione supporta il TTS neurale (NTTS) e le voci standard. Per ulteriori informazioni, consulta Voci neurali e Voci standard.	27 marzo 2025
Nuova voce aggiunta per neural text-to-speech	Amazon Polly ora offre una voce coreana aggiuntiva: Jihye. Consulta Voci neurali per un elenco di voci NTTS.	26 marzo 2025
Aggiunta una nuova regione per le voci neurali e standard	Amazon Polly è ora disponibile nella regione Europa (Spagna). AWS Questa regione supporta il TTS neurale (NTTS) e le voci standard. Per ulteriori informazioni, consulta Voci neurali e Voci standard.	18 febbraio 2025
Nuova voce aggiunta per neural text-to-speech	Amazon Polly ora fornisce una voce aggiuntiva in inglese (singaporiano): Jasmine.	11 febbraio 2025

	Consulta Voci neurali per un elenco di voci NTTS.	
Nuove voci aggiunte per la modalità generativa text-to-speech	Amazon Polly offre ora sette nuove voci generative: Pedro, Andrés, Sergio, Daniel, Kajal, Rémi, Bianca. Vedi Generative voices per un elenco di voci TTS generative.	21 novembre 2024
Sono state aggiunte nuove voci di lunga durata	Amazon Polly ora offre voci più lunghe. Sono state aggiunte un'altra voce in inglese (Stati Uniti) e due nuove voci di lunga durata in spagnolo (Spagna): Patrick, Alba e Raúl. Vedi Voci in formato lungo per un elenco di tutte le voci in formato lungo.	14 novembre 2024
Nuove voci aggiunte per scopi generativi text-to-speech	Amazon Polly ora offre sei nuove voci generative: Ayanda, Léa, Lucia, Mía, Lupe e Vicki. Consulta Generative voices per un elenco di voci TTS generative.	6 novembre 2024
Nuove voci aggiunte per la modalità generativa text-to-speech	Amazon Polly ora offre quattro nuove voci generative: Olivia, Danielle, Joanna e Stephen. Consulta Generative voices per un elenco di voci TTS generative.	10 ottobre 2024

Nuove lingue aggiunte per text-to-speech	Amazon Polly ora offre due nuove lingue TTS: ceco (CS-CZ) e tedesco (svizzero) (de-CH). Vedi Lingue supportate per un elenco di lingue.	26 settembre 2024
Nuove voci aggiunte per NTTS	Amazon Polly ora offre due nuove voci NTTS: Jitka e Sabrina. Consulta Voci neurali per un elenco di voci NTTS.	27 agosto 2024
Aggiunto un nuovo motore vocale generativo	Amazon Polly ora offre un motore vocale generativo progettato per contenuti più lunghi, con tre voci inglesi in una variante generativa: Amy, Matthew e Ruth. Per ulteriori informazioni, consulta Generative voices.	28 marzo 2024
Aggiunta nuova voce per NTTS	Amazon Polly ora fornisce la voce turca NTTS Burcu. Consulta Voci neurali per un elenco di voci NTTS.	14 febbraio 2024
Aggiunto un nuovo motore vocale di lunga durata	Amazon Polly ora offre un motore vocale di lunga durata progettato per contenuti più lunghi, con tre voci en-US: Danielle, Gregory e Ruth. Per ulteriori informazioni, consulta Voci in formato lungo.	16 novembre 2023

Nuove voci aggiunte per NTTS	Amazon Polly fornisce ora due nuove voci NTTS in inglese americano: Danielle e Gregory. Consulta Voci neurali per un elenco di voci NTTS.	5 ottobre 2023
Amazon Polly per Windows	Il plugin Amazon Polly Windows Speech Application Programming Interface (SAPI) non sarà più supportato.	26 settembre 2023
Aggiornata la guida delle quote per Amazon Polly	Guida aggiornata alle quote di Amazon Polly. Sono stati aggiunti esempi e chiarimen ti sui termini. Fare riferimento a Quote in Amazon Polly per gli aggiornamenti.	17 agosto 2023
Aggiunta nuova voce per NTTS	Amazon Polly ora fornisce la voce Zayd in arabo del Golfo come NTTS. Consulta Voci neurali per un elenco di voci NTTS.	16 agosto 2023
Aggiunta nuova voce per NTTS	Amazon Polly ora fornisce la voce francese belga Isabelle come voce NTTS. Consulta Voci neurali per un elenco di voci NTTS.	1° agosto 2023
Aggiunta nuova voce per NTTS	Amazon Polly ora fornisce la voce NTTS belga olandese (fiamminga) Lisa. Consulta Voci neurali per un elenco di voci NTTS.	7 giugno 2023

Nuove voci aggiunte per NTTS	Amazon Polly fornisce ora due nuove voci NTTS: inglese irlandese (Niamh) e danese (Sofie). Consulta Voci neurali per un elenco di voci NTTS.	30 maggio 2023
Aggiornata la guida IAM per Amazon Polly	Guida aggiornata per l'allinea mento alle best practice IAM. Per ulteriori informazioni, consulta Best practice per la sicurezza in IAM.	19 aprile 2023
WordPress update	Il WordPress plug-in Amazon Polly non sarà più supportato.	6 aprile 2023
Nuova regione aggiunta	Amazon Polly è ora disponibi le nella regione AWS Asia Pacifico (Osaka). Questa regione supporta il TTS neurale (NTTS). Per maggiori informazioni e per un elenco di regioni che supportano NTTS, consulta Caratteristiche e compatibilità delle regioni.	5 aprile 2023
Nuove voci aggiunte per NTTS	Amazon Polly fornisce ora due nuove voci NTTS in giappones e: Kazuha e Tomoko. Consulta Voci neurali per un elenco di voci NTTS.	7 febbraio 2023
Nuove voci aggiunte per NTTS	Amazon Polly fornisce ora due nuove voci NTTS in inglese americano: Stephen e Ruth. Consulta Voci neurali per un elenco di voci NTTS.	31 gennaio 2023

Nuove voci aggiunte per NTTS	Amazon Polly fornisce ora nuove voci NTTS per: portoghese brasiliano (Thiago), spagnolo castiglia no (Sergio), francese (Rémi), italiano (Adriano) e spagnolo messicano (Andrés). Consulta Voci neurali per un elenco di voci NTTS.	24 gennaio 2023
Nuove voci aggiunte per NTTS	Amazon Polly fornisce ora voci NTTS per arabo (Hala) e polacco (Ola). Consulta Voci neurali per un elenco di voci NTTS.	17 novembre 2022
Supporto per le versioni AWS PrivateLink	Amazon Polly ora fornisce AWS PrivateLink supporto. Per ulteriori informazioni, consulta Utilizzo di Amazon Polly con endpoint VPC.	9 novembre 2022
Nuove voci e lingue aggiunte per NTTS	Amazon Polly fornisce ora voci NTTS per finlandese (Suvi), norvegese (Ida) e svedese (Elin). Consulta Voci neurali per un elenco di voci NTTS.	8 novembre 2022
Aggiunta nuova voce per NTTS	Amazon Polly fornisce ora la voce NTTS olandese Laura. Consulta Voci neurali per un elenco di voci NTTS.	2 novembre 2022

Nuova regione aggiunta	Amazon Polly è ora disponibile nella regione AWS Europa (Parigi). Questa regione supporta il TTS neurale (NTTS). Per maggiori informazioni e per un elenco di regioni che supportano NTTS, consulta Caratteristiche e compatibilità delle regioni.	22 settembre 2022
Nuove voci e lingue aggiunte per NTTS	Amazon Polly fornisce ora la voce NTTS cantonese Hiujin. Consulta Voci neurali per un elenco di voci NTTS.	20 settembre 2022
Nuova regione aggiunta	Amazon Polly è ora disponibile nella regione AWS Asia Pacifico (Mumbai). Questa regione supporta il TTS neurale (NTTS). Per maggiori informazioni e per un elenco di regioni che supportano NTTS, consulta Caratteristiche e compatibilità delle regioni.	1 settembre 2022
Aggiunta nuova voce per NTTS	Amazon Polly fornisce ora la voce mandarino Zhiyu come una voce NTTS. Consulta Voci neurali per un elenco di voci NTTS.	23 agosto 2022
Aggiunta nuova voce per NTTS	Amazon Polly fornisce ora la voce NTTS Hindi Kajal. Consulta Voci neurali per un elenco di voci NTTS.	27 luglio 2022

Nuove voci aggiunte per NTTS	Amazon Polly fornisce ora voci NTTS per spagnolo americano (Pedro), tedesco (Daniel), francese canadese (Liam) e inglese britannico (Arthur). Consulta Voci neurali per un elenco di voci NTTS.	28 giugno 2022
Aggiunta nuova voce per NTTS	Amazon Polly fornisce ora la voce portoghese (brasiliano) Vitória come una voce NTTS. Consulta Voci neurali per un elenco di voci NTTS.	27 aprile 2022
Aggiunta nuova voce per NTTS	Amazon Polly fornisce ora la voce portoghese (europeo) Inês come una voce NTTS. Consulta Voci neurali per un elenco di voci NTTS.	26 aprile 2022
Nuove voci e lingue aggiunte per NTTS	Amazon Polly ora fornisce la lingua tedesco (austriaco) e la voce NTTS Hannah. Consulta Voci neurali per un elenco di voci NTTS.	19 aprile 2022
Nuove voci e lingue aggiunte per NTTS	Amazon Polly ora fornisce la voce spagnola (messicano) Mia come voce NTTS. È stata aggiunta una nuova lingua, il catalano, insieme alla voce NTTS Arlet. Consulta Voci neurali per un elenco di voci NTTS.	22 marzo 2022

Aggiunta nuova voce per NTTS	Amazon Polly ora fornisce la voce giapponese Takumi come voce NTTS. Consulta Voci neurali per un elenco di voci NTTS.	6 dicembre 2021
Aggiunta nuova voce per NTTS	Amazon Polly ora fornisce la voce francese Léa come voce NTTS. Consulta Voci neurali per un elenco di voci NTTS.	18 novembre 2021
Nuove voci aggiunte per NTTS	Amazon Polly ora fornisce la voce italiana Bianca e la voce spagnola europea Lucia come voci NTTS. Consulta Voci neurali per un elenco di voci NTTS.	8 novembre 2021
Aggiunta nuova voce per NTTS	Amazon Polly ora fornisce una nuova voce inglese sudafricana, Ayanda. Questa voce è disponibile solo come voce NTTS. Consulta Voci neurali per un elenco di voci NTTS.	1 settembre 2021
Nuova regione aggiunta	Amazon Polly è ora disponibile nella regione AWS Africa (Città del Capo). Questa regione supporta il TTS neurale (NTTS). Per maggiori informazioni e per un elenco di regioni che supportano NTTS, consulta Caratteristiche e compatibilità delle regioni.	1 settembre 2021

Aggiunte nuove lingua e voce	Amazon Polly ora supporta l'inglese neozelandese (en-NZ). Una nuova voce NTTS, Aria, parla inglese neozeland ese e supporta una selezione di parole maori.	24 agosto 2021
Nuova caratteristica	Amazon Polly rende lo stile di conversazione la versione predefinita per le voci neurali Matthew e Joanna. Abbiamo rimosso i riferimenti allo stile di conversazione.	28 giugno 2021
Aggiunta nuova voce per NTTS	Amazon Polly ora fornisce la voce tedesca Vicki come voce NTTS.	15 giugno 2021
Nuova voce aggiunta	Una nuova voce femminile, Gabrielle, è stata aggiunta alle impostazioni locali francese (canadese) (fr-CA). Questa nuova voce è di alta qualità ed è disponibile solo come voce NTTS. Come tutte le voci neurali, è disponibile solo in determinate regioni. Per l'elenco delle regioni, consulta Feature and region compatibi lity (Caratteristiche e compatibi lità delle regioni).	1 giugno 2021
Aggiunta nuova voce per NTTS	Amazon Polly ora fornisce la voce coreana Seoyeon come voce NTTS.	11 maggio 2021

Aggiunta nuova Regione per NTTS	Amazon Polly ora supporta il TTS neurale (NTTS) nella regione Canada (Centrale). AWS Per ulteriori informazioni, consulta Feature and Region Compatibility (Caratteristiche e compatibilità delle regioni) per NTTS.	17 marzo 2021
Nuova voce disponibile per lo stile newscaster	Oltre alle voci di Matthew, Joanna e Lupe per lo stile Newscaster,Amazon Polly ora fornisce un'opzione aggiuntiv a per questo stile di pronuncia . Utilizzando il motore neurale, è possibile utilizzare la voce Amy in inglese britanico per lo stile Newscaster. Per ulteriori informazioni, consultare la sezione relativa agli stili di pronuncia NTTS.	10 novembre 2020
Aggiunte nuove Regioni per NTTS	Oltre alle regioni esistenti per NTTS (us-east-1, us-west-2, eu-west-1 e ap-southeast-2), le voci neurali sono ora supportate in quattro regioni aggiuntive: ap-northeast-1 (Tokyo), ap-southeast-1 (Singapore), eu-central-1 (Francoforte) e eu-west-2 (Londra). Per ulteriori informazioni, consulta Feature and Region Compatibility (Caratteristiche e compatibilità delle regioni) per NTTS.	3 settembre 2020

Nuova voce aggiunta	Oltre alle voci di bambino di Ivy e Justin, la nuova voce di un bambino, Kevin, è stata aggiunta all'inglese americano (en-US). Questa nuova voce è di altissima qualità ed è disponibile solo come voce NTTS. Come tutte le voci neurali, è supportata solo in quattro regioni: us-east-1 (Virginia settentrionale), us-west-2 (Oregon), eu-west-1 (Irlanda) e ap-southeast-2 (Sydney). Per ulteriori informazioni, consulta Voci NTTS.	16 giugno 2020
Nuova voce disponibile per lo stile newscaster	Oltre alle voci di Matthew e Joanna per lo stile Newscaster, Amazon Polly ora fornisce un'opzione aggiuntiva per questo stile di pronuncia. Utilizzando il motore neurale, è possibile utilizzare la voce Lupe in spagnolo (americano) per lo stile Newscaster. Per ulteriori informazioni, consultare la sezione relativa agli stili di pronuncia NTTS.	16 aprile 2020

Nuova caratteristica	Oltre allo stile di pronuncia Newscaster, Amazon Polly ora fornisce un secondo stile di pronuncia NTTS per aiutarti a sintetizzare ancora meglio i passaggi di sintesi vocale. Lo stile Conversational utilizza il sistema neurale per generare un discorso in uno stile di conversazione più amichevole ed espressivo che può essere utilizzato in molti casi d'uso. Per ulteriori informazioni, consultare la sezione relativa agli stili di pronuncia NTTS.	25 novembre 2019
Nuove voci aggiunte	Aggiunte due nuove voci: Camila (femminile, Portoghes e-Brasile) e Lupe (femmina, Spagnolo-Stati Uniti).	23 ottobre 2019
Nuova caratteristica aggiunta	Aggiunta di Amazon Polly for Windows plugin (Amazon Polly per plug-in per Windows) per incorporare la gamma completa di voci Amazon Polly nelle applicazioni compatibili con Windows SAPI.	26 settembre 2019

Nuova caratteristica principale	Oltre alle voci standard text-to-speech (TTS) supportat e da Amazon Polly sin dal suo lancio, Amazon Polly ora offre un sistema TTS neurale (NTTS) migliorato in grado di fornire voci di qualità ancora superiore, fornendoti così le voci più naturali e simili a quelle umane possibili. text-to-speech Per ulteriori informazi oni, consulta Neural. Text-to-S peech	30 luglio 2019
Nuove voci aggiunte	Nuove voci aggiunte: Lucia (femmina, spagnolo) e Bianca (femmina, italiano).	2 agosto 2018
Nuova lingua aggiunta	Nuova lingua aggiunta: spagnolo, Messico (es-MX). Questa lingua utilizza la voce femminile di Mia.	2 agosto 2018
Nuova lingua aggiunta	Nuova lingua aggiunta: hindi (hi-IN). Questa voce utilizza la voce femminile di Aditi, che viene utilizzata anche per l'inglese indiano, pertanto Aditi è la prima voce bilingue di Amazon Polly.	2 agosto 2018
Nuova caratteristica aggiunta	Aggiunta della sintesi vocale di lunghi passaggi di testo (fino a 100.000 caratteri fatturati).	17 luglio 2018
Nuova caratteristica SSML aggiunta	Aggiunta della durata massima per la sintesi vocale.	17 luglio 2018

Nuova voce aggiunta	Nuova voce aggiunta: Léa (femminile, francese).	5 giugno 2018
Espansione regionale	Ampliamento del servizio Amazon Polly a tutte le regioni commerciali.	4 giugno 2018
Nuova lingua aggiunta	Nuova lingua aggiunta: coreano (ko-KR).	4 giugno 2018
Caratteristica ampliata	La funzionalità Amazon Polly WordPress Plugin, inclusa l'aggiunta delle funzionalità di Amazon Translate.	4 giugno 2018
Nuove voci aggiunte	Due nuove voci aggiunte: Aditi (donna, anglo-indiana) e Seoyeon (donna, coreana).	15 novembre 2017
Nuova caratteristica	Aggiunta della nuova caratteristica Contrassegni vocali, nonché un'espansione delle funzionalità SSML.	19 aprile 2017
Nuova guida	La prima versione della Amazon Polly Developer Guide (Guida per gli sviluppatori di Amazon Polly).	30 novembre 2016

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.