aws

Guía para desarrolladores

Amazon Polly

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

Amazon Polly: Guía para desarrolladores

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

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon, de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

¿Qué es Amazon Polly? 1
Funcionamiento 1
Ventajas 2
¿Es la primera vez que usa ? 3
Trabajando con AWS SDKs 3
Introducción 5
Registrarse en AWS 5
Inscríbase en un Cuenta de AWS 5
Creación de un usuario con acceso administrativo 6
Configuración del AWS CLI 8
Reconfigurar el AWS CLI 9
Ejemplo de síntesis de voz 11
Voces en Amazon Polly 15
Voces disponibles 15
Voces de marca 23
Voces bilingües 23
Voces bilingües con acento 23
Voces totalmente bilingües 24
Aplicación de la voz de presentador 25
Cómo escuchar las voces 28
Velocidad de las voces 28
Cambio de la velocidad de una voz 29
Idiomas en Amazon Polly 31
Árabe (arb) 34
Árabe del Golfo (ar-AE) 39
Catalán (ca-ES) 46
Chino (cantonés) (yue-CN) 50
Chino, mandarín (cmn-CN) 54
Checo (cs-CZ) 58
Danés (da-DK) 62
Neerlandés (belga) (NL-BE) 67
Neerlandés (nl-NL) 71
Inglés (EE. UU.) (en-US) 75
Inglés (Australia) (en-AU) 79

Inglés (Reino Unido) (en-GB) 84
Inglés (India) (en-IN) 88
Inglés (Irlanda) (en-IE) 93
Inglés (Nueva Zelanda) (en-NZ) 96
inglés (singapurense) (en-SG) 103
Inglés (sudafricano) (en-ZA) 107
Inglés (Gales) (en-GB-WLS) 112
Finés (fi-FI) 116
Francés (fr-FR) 122
Francés (belga) (fr-BE) 125
Francés (Canadá) (fr-CA) 129
Alemán (de-DE) 133
Alemán (austriaco) (de-AT) 138
Alemán (estándar suizo) (de-CH) 143
Hindi (hi-IN) 148
Islandés (is-IS) 152
Italiano (it-IT) 156
Japonés (ja-JP) 160
Coreano (ko-KR) 164
Noruego (nb-NO) 167
Polaco (pl-PL) 171
Portugués (pt-PT) 175
Portugués (Brasil) (pt-BR) 179
Rumano (ro-RO) 182
Ruso (ru-RU) 185
Español (es-ES) 189
Español (México) (es-MX) 193
Español (EE.UU.) (es-US) 196
Sueco (sv-SE) 199
Turco (tr-TR) 203
Galés (cy-GB) 208
Motores de voz 213
Motor generativo 213
Voces generativas disponibles 214
Compatibilidad con características y regiones 216
Motor de formato largo 217

Voces de formato largo disponibles 218
Compatibilidad con características y regiones 218
Motor neuronal 219
Voces neuronales disponibles 220
Compatibilidad con características y regiones 223
Motor estándar 225
Voces estándar disponibles 225
Compatibilidad con características y regiones 228
Elección de un motor de voz 230
Marcas de voz 231
Tipos de marcas de voz 231
Visemas y Amazon Polly 232
Salida de marcas de voz 233
Solicitud de marcas de voz 234
Ejemplo de marcas de voz sin SSML 236
Ejemplo de marcas de voz con SSML 237
Uso de SSML 239
Caracteres reservados 240
Uso de SSML en la consola 243
Uso de SSML con el comando de síntesis de voz 244
Síntesis de un documento mejorado con SSML 245
Etiquetas SSML admitidas 246
Identificación de texto mejorado con SSML 249
Agregación de una pausa 249
Énfasis de palabras 251
Especificación de otro idioma para palabras específicas 251
Colocación de una etiqueta personalizada en el texto 253
Agregación de una pausa entre párrafos 253
Uso de la pronunciación fonética 254
Control del volumen, velocidad de habla y tono 256
Establecimiento de una duración máxima para voz sintetizada 258
Agregación de una pausa entre frases 262
Control de cómo se leen los tipos especiales de palabras 263
Pronunciación de acrónimos y abreviaturas 266
Mejora de la pronunciación especificando partes del discurso 267
Adición de sonido de respiración 268

Estilo de habla Newscaster ... 273
Adición de compresión de rango dinámico ... 273
Habla de forma suave ... 276
Control del timbre ... 276
Susurros ... 278
Administración de lexicones ... 280
Uso de varios lexicones ... 281
Carga de un lexicón ... 282
Aplicación de lexicones (síntesis del habla) ... 288
Filtrado de la lista de lexicones en la consola ... 292
Descarga de lexicones en la consola ... 293
Eliminación de un lexicón ... 295
Archivo de audio largos ... 297
Configuración de la política de IAM para síntesis asíncrona ... 298
Creación de archivos de audio largos ... 299
Cuotas ... 304
Regiones de admitidas ... 305
Cuotas y tasas de limitación ... 305
Solicitudes simultáneas ... 306
Mejores prácticas para mitigar la limitación ... 306
Lexicones de pronunciación ... 307
SynthesizeSpeech Operaciones de API ... 307
SpeechSynthesisTask Operaciones de API ... 308
Speech Synthesis Markup Language (SSML, idioma de marcado de síntesis de voz) ... 308
Ejemplos de código y aplicaciones ... 310
Muestras de Java ... 310
DeleteLexicon ... 311
DescribeVoices ... 313
GetLexicon ... 314
ListLexicons ... 316
PutLexicon ... 317
StartSpeechSynthesisTask ... 320
Marcas de voz ... 324
SynthesizeSpeech ... 327
Muestras de Python ... 328
DeleteLexicon ... 328

GetLexicon ... 329
ListLexicon ... 331
PutLexicon ... 332
StartSpeechSynthesisTask ... 333
SynthesizeSpeech ... 334
Ejemplo de Java ... 334
Ejemplo de Python ... 339
Ejemplo de Python: index.html ... 341
Ejemplo de Python: server.py ... 346
Ejemplo de iOS ... 352
Ejemplo de Android ... 354
Ejemplos de código ... 357
Conceptos básicos ... 358
Acciones ... 358
Escenarios ... 401
Convierta texto en voz y de nuevo a texto ... 401
Crear una aplicación de sincronización labial ... 402
Creación de una aplicación para analizar los comentarios de los clientes ... 403
Seguridad ... 410
Protección de los datos ... 411
Cifrado en reposo ... 411
Cifrado en tránsito ... 412
Privacidad del tráfico entre redes ... 412
Identity and Access Management ... 412
Público ... 412
Autenticación con identidades ... 413
Administración de acceso mediante políticas ... 417
Cómo funciona Amazon Polly con IAM ... 420
Ejemplos de políticas basadas en identidades ... 428
Referencia sobre los permisos de la API de Amazon Polly ... 435
Solución de problemas ... 437
Registro y supervisión ... 439
Validación de la conformidad ... 439
Resiliencia ... 440
Seguridad de infraestructuras ... 441
Prácticas recomendadas de seguridad ... 441

Uso de los puntos de enlace de la VPC de tipo interfaz 442
Disponibilidad 442
Creación de una política de puntos de conexión de VPC para Amazon Polly 442
Comprobación de la conexión entre la VPC y Amazon Polly 443
Control del acceso al punto de conexión de VPC de Amazon Polly 443
Compatibilidad con las claves de contexto de la VPC 444
Registrar llamadas a la API Amazon Polly con AWS CloudTrail 445
Información sobre Amazon Polly en CloudTrail 445
Ejemplo: Entradas del archivo de registro de Amazon Polly 446
CloudWatch integración 449
Obtención de CloudWatch métricas (consola) 449
Obtener CloudWatch las métricas en AWS CLI 449
Métricas de Amazon Polly 450
Dimensiones para las métricas de Amazon Polly 452
referencia de la API 453
Acciones 453
DeleteLexicon 454
DescribeVoices 456
GetLexicon 460
GetSpeechSynthesisTask 463
ListLexicons 466
ListSpeechSynthesisTasks 469
PutLexicon 472
StartSpeechSynthesisTask 475
SynthesizeSpeech 483
Data Types 489
Lexicon 491
LexiconAttributes 492
LexiconDescription 494
SynthesisTask 495
Voice 500
Errores comunes 502
Parámetros comunes 504
Historial de documentos 507
..... dxxiii

¿Qué es Amazon Polly?

Amazon Polly es un servicio en la nube que convierte el texto en un segmento hablado muy realista. Puede utilizar Amazon Polly para desarrollar aplicaciones que aumenten la participación y mejoren la accesibilidad. Amazon Polly está disponible en varios idiomas e incluye una variedad de voces realistas. Con Amazon Polly puede crear aplicaciones que pueden usarse en distintas ubicaciones y emplear la voz más adecuada para los clientes. Además, solo paga por el texto que se sintetiza. También puede almacenar en caché el habla generada con Amazon Polly y reproducirla sin ningún costo adicional.

Amazon Polly ofrece muchas opciones de voz, incluidas las opciones generativas, de formato largo, neuronales y estándar text-to-speech (TTS). Estas voces ofrecen mejoras revolucionarias en la calidad de la voz gracias a la nueva tecnología de aprendizaje automático para ofrecer las voces más naturales y parecidas a las humanas posibles. text-to-speech La tecnología TTS neuronal también admite el estilo de habla Presentador adaptado a casos de uso de locución de noticias.

Entre los casos de uso comunes de Amazon Polly se incluyen, entre otras, las aplicaciones móviles, como lectores de noticias, juegos, plataformas de recursos de aprendizaje electrónico, las aplicaciones de accesibilidad para personas con discapacidad y el segmento de Internet de las cosas (IoT), en rápido crecimiento.

Amazon Polly está certificado para su uso con cargas de trabajo reguladas por la HIPAA (Ley de Portabilidad y Responsabilidad de Seguros Médicos de 1996) y el Estándar de Seguridad de Datos del Sector de las Tarjetas de Pago (PCI DSS).

Cómo funciona Amazon Polly

Amazon Polly convierte el texto de entrada en fragmentos de voz en lenguaje natural. Para usar una voz de Amazon Polly, elija un motor de voz, llame a un método de síntesis de voz, proporcione el texto que desee sintetizar y, a continuación, especifique un formato de salida de audio. A continuación, Amazon Polly sintetiza el texto proporcionado en una secuencia de audio hablada de gran calidad.

- Texto de entrada: introduzca el texto que desea sintetizar y Amazon Polly devolverá una secuencia de audio. Puede proporcionar la entrada como texto sin formato o con el formato SSML (lenguaje de marcado de síntesis de voz). Si utiliza SSML, podrá controlar diferentes aspectos del fragmento hablado, como la pronunciación, el volumen, el tono y la velocidad de habla. Para obtener más información, consulte Generación de fragmentos hablados desde documentos SSML.

- Voces disponibles: Amazon Polly proporciona una serie de idiomas y diversas voces, incluida una voz bilingüe (tanto para inglés como para hindi). En la mayoría de los idiomas, puede elegir entre varias voces, tanto masculinas como femeninas. Especifique el nombre del ID de la voz al lanzar la tarea de síntesis de voz y Amazon Polly la utilizará para convertir el texto en voz. Amazon Polly no es un servicio de traducción: el fragmento sintetizado está en el mismo idioma que el texto. Los números escritos con dígitos (por ejemplo, 53, en lugar de cincuenta y tres) se sintetizan en el idioma de la voz y no del texto. Para obtener más información, consulte Voces en Amazon Polly.
- Formato de salida: Amazon Polly puede proporcionar el resultado de la síntesis de voz en varios formatos. Puede seleccionar el formato de audio que mejor se adapte a sus necesidades. Por ejemplo, puede solicitar el discurso en formato Ogg Vorbis MP3 o en formato Ogg Vorbis para su uso en aplicaciones web y móviles. O bien, puede solicitar el formato de salida PCM para que lo consuman los AWS IoT dispositivos y las soluciones de telefonía.

> **Note**
>
> Para escuchar voces de ejemplo de Amazon Polly en el navegador, consulte la descripción general del producto Amazon Polly.

Ventajas

Alguno de los beneficios de usar Amazon Polly son:

- Alta calidad: Amazon Polly ofrece voces generativas, de formato largo, neuronales y de alta calidad (TTS) de alto rendimiento. text-to-speech Estas tecnologías sintetizan el lenguaje natural con una pronunciación muy precisa (incluida la expansión de abreviaturas y acrónimos, la interpretación de fechas y horas y la desambiguación homográfica).
- Baja latencia: Amazon Polly obtiene tiempos de respuesta rápidos, lo que la convierte en una opción viable para casos de uso de baja latencia, como los sistemas de diálogo.
- Compatibilidad con un amplio conjunto de idiomas y voces: Amazon Polly admite docenas de voces e idiomas, la mayoría de los cuales disponen de voces masculinas y femeninas. Este número seguirá aumentando a medida que pongamos en red más voces neuronales. Las voces de Matthew y Joanna en inglés de Estados Unidos también pueden utilizar el estilo de habla neuronal Newscaster, similar al de un locutor de noticias profesional.

- Rentable: el pay-per-use modelo de Amazon Polly significa que no hay costes de instalación. Comience con pocos recursos y vaya escalándolos verticalmente a medida que crece la aplicación.
- Solución basada en la nube: las soluciones de conversión de texto a voz instaladas en los dispositivos necesitan importantes recursos informáticos y una gran potencia de CPU, así como gran cantidad de RAM y de espacio en disco. Esto puede generar elevados costos de desarrollo y un mayor consumo de potencia en dispositivos como tablets, smartphones, etc. Por el contrario, la conversión a TTS realizada en el reduce Nube de AWS drásticamente los requisitos de recursos locales. Esto permite usar todos los idiomas y voces disponibles con una calidad excepcional. Además, las mejoras del fragmento hablado están disponibles de forma instantánea para todos los usuarios finales y no requieren nuevas actualizaciones de los dispositivos.

> **Note**
>
> Para escuchar voces de ejemplo de Amazon Polly en el navegador, consulte la descripción general del producto Amazon Polly.

¿Es la primera vez que usa ?

Si es la primera vez que utiliza Amazon Polly, le recomendamos que lea las siguientes secciones en el orden en el que aparecen:

1. Cómo funciona Amazon Polly: en esta sección, se presentan las diferentes entradas y opciones de Amazon Polly con las que puede trabajar para crear una experiencia sencilla.
2. Introducción a Amazon Polly: en esta sección, configurará la cuenta y probará la síntesis de voz de Amazon Polly.
3. Ejemplos de código y aplicaciones para Amazon Polly: en esta sección, encontrará otros ejemplos que puede usar para conocer Amazon Polly en profundidad.

Uso de Amazon Polly con un SDK AWS

AWS Los kits de desarrollo de software (SDKs) están disponibles para muchos lenguajes de programación populares. Cada SDK proporciona una API, ejemplos de código y documentación que facilitan a los desarrolladores la creación de aplicaciones en su lenguaje preferido.

Documentación de SDK	Ejemplos de código
AWS SDK para C++	AWS SDK para C++ ejemplos de código
AWS CLI	AWS CLI ejemplos de código
AWS SDK para Go	AWS SDK para Go ejemplos de código
AWS SDK para Java	AWS SDK para Java ejemplos de código
AWS SDK para JavaScript	AWS SDK para JavaScript ejemplos de código
AWS SDK para Kotlin	AWS SDK para Kotlin ejemplos de código
AWS SDK para .NET	AWS SDK para .NET ejemplos de código
AWS SDK para PHP	AWS SDK para PHP ejemplos de código
Herramientas de AWS para PowerShell	Herramientas para ejemplos PowerShell de código
AWS SDK para Python (Boto3)	AWS SDK para Python (Boto3) ejemplos de código
AWS SDK para Ruby	AWS SDK para Ruby ejemplos de código
AWS SDK para Rust	AWS SDK para Rust ejemplos de código
AWS SDK para SAP ABAP	AWS SDK para SAP ABAP ejemplos de código
AWS SDK para Swift	AWS SDK para Swift ejemplos de código

Ejemplo de disponibilidad

¿No encuentra lo que necesita? Solicite un ejemplo de código a través del enlace de Enviar comentarios que se encuentra al final de esta página.

Introducción a Amazon Polly

Amazon Polly cuenta con varias operaciones de la API que se pueden integrar de un modo sencillo con las aplicaciones existentes. Para ver una lista de las operaciones admitidas, consulte Acciones.

Puede realizar prácticamente las mismas operaciones en la consola de Amazon Polly y en la AWS CLI. Sin embargo, no puede escuchar un discurso sintetizado en el AWS CLI. Para trabajar con audio en la AWS CLI, guarde el texto en un archivo. Después, abra el archivo en la aplicación de audio que prefiera.

Puede utilizar una de las siguientes opciones:

- AWS SDKs — Cuando utilices el SDKs, tus solicitudes a Amazon Polly se firman y autentican automáticamente con las credenciales que proporciones. Esta es la opción recomendada para compilar las aplicaciones.
- AWS CLI — Puedes AWS CLI usar Amazon Polly sin escribir ningún código.

Antes de utilizar Amazon Polly por primera vez, debe registrarse en AWS. Cuando te registras en Amazon Web Services (AWS), tu AWS cuenta se registra automáticamente en todos los servicios de Amazon Polly AWS, incluido Amazon Polly. Solo se le cobrará por los servicios y recursos que utilice. Si eres un AWS cliente nuevo, puedes empezar a utilizar Amazon Polly sin coste alguno. Para obtener más información, consulte Nivel de uso gratuito de AWS.

En las secciones siguientes se describe cómo comenzar a utilizar Amazon Polly.

Temas

- Registrarse en AWS
- Configuración del AWS CLI
- Reconfigurar el AWS CLI

Registrarse en AWS

Antes de poder utilizar cualquier AWS servicio, incluido Amazon Polly, debes registrarte en. AWS

Inscríbase en un Cuenta de AWS

Si no tiene uno Cuenta de AWS, complete los siguientes pasos para crearlo.

Para suscribirte a una Cuenta de AWS

1. Abrir https://portal.aws.amazon.com/billing/registro.
2. Siga las instrucciones que se le indiquen.

   Parte del procedimiento de registro implica recibir una llamada telefónica o un mensaje de texto e introducir un código de verificación en el teclado del teléfono.

   Cuando te registras en un Cuenta de AWS, Usuario raíz de la cuenta de AWSse crea un. El usuario raíz tendrá acceso a todos los Servicios de AWS y recursos de esa cuenta. Como práctica recomendada de seguridad, asigne acceso administrativo a un usuario y utilice únicamente el usuario raíz para realizar tareas que requieren acceso de usuario raíz.

AWS te envía un correo electrónico de confirmación una vez finalizado el proceso de registro. En cualquier momento, puede ver la actividad de su cuenta actual y administrarla accediendo a https://aws.amazon.com/y seleccionando Mi cuenta.

Creación de un usuario con acceso administrativo

Después de crear un usuario administrativo Cuenta de AWS, asegúrelo Usuario raíz de la cuenta de AWS AWS IAM Identity Center, habilite y cree un usuario administrativo para no usar el usuario root en las tareas diarias.

Proteja su Usuario raíz de la cuenta de AWS

1. Inicie sesión AWS Management Consolecomo propietario de la cuenta seleccionando el usuario root e introduciendo su dirección de Cuenta de AWS correo electrónico. En la siguiente página, escriba su contraseña.

   Para obtener ayuda para iniciar sesión con el usuario raíz, consulte Iniciar sesión como usuario raíz en la Guía del usuario de AWS Sign-In .
2. Active la autenticación multifactor (MFA) para el usuario raíz.

   Para obtener instrucciones, consulte Habilitar un dispositivo MFA virtual para el usuario Cuenta de AWS raíz (consola) en la Guía del usuario de IAM.

Creación de un usuario con acceso administrativo

1. Activar IAM Identity Center.

Consulte las instrucciones en Activar AWS IAM Identity Center en la Guía del usuario de AWS IAM Identity Center .

2. En IAM Identity Center, conceda acceso administrativo a un usuario.

   Para ver un tutorial sobre su uso Directorio de IAM Identity Center como fuente de identidad, consulte Configurar el acceso de los usuarios con la configuración predeterminada Directorio de IAM Identity Center en la Guía del AWS IAM Identity Center usuario.

Inicio de sesión como usuario con acceso de administrador

- Para iniciar sesión con el usuario de IAM Identity Center, use la URL de inicio de sesión que se envió a la dirección de correo electrónico cuando creó el usuario de IAM Identity Center.

  Para obtener ayuda para iniciar sesión con un usuario del Centro de identidades de IAM, consulte Iniciar sesión en el portal de AWS acceso en la Guía del AWS Sign-In usuario.

Concesión de acceso a usuarios adicionales

1. En IAM Identity Center, cree un conjunto de permisos que siga la práctica recomendada de aplicar permisos de privilegios mínimos.

   Para conocer las instrucciones, consulte Create a permission set en la Guía del usuario de AWS IAM Identity Center .

2. Asigne usuarios a un grupo y, a continuación, asigne el acceso de inicio de sesión único al grupo.

   Para conocer las instrucciones, consulte Add groups en la Guía del usuario de AWS IAM Identity Center .

Para obtener más información sobre IAM, consulte lo siguiente:

- AWS Identity and Access Management (IAM)
- Introducción
- Guía del usuario de IAM

> **Note**
>
> Anote el ID AWS de su cuenta. Lo necesitará en los siguientes pasos.

Configuración del AWS CLI

Sigue estos pasos para descargar y configurar el AWS CLI para que funcione con Amazon Polly.

Para configurar el AWS Command Line Interface

1. Descargue y configure la AWS CLI. Para obtener instrucciones, consulte los siguientes temas en la Guía del usuario de AWS Command Line Interface :
   - Cómo configurarse con el AWS Command Line Interface
   - Configurando el AWS Command Line Interface
2. Agregue un perfil con nombre para el usuario administrador en el archivo AWS CLI AWS Config. Puede usar este perfil al ejecutar los AWS CLI comandos. Para obtener más información sobre los perfiles con nombre, consulte Perfiles con nombre en la Guía del usuario de la 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
   ```

   Para obtener una lista de AWS las regiones disponibles y de las que admite Amazon Polly, consulte Regiones y puntos de enlace en. Referencia general de Amazon Web Services

   > **Note**
   >
   > Si utiliza una región compatible con Amazon Polly que especificó al configurar la AWS CLI, omita la siguiente línea de los ejemplos de código. AWS CLI
   >
   > ```
   > --region aws-region
   > ```

3. Compruebe la configuración; para ello, escriba el siguiente comando de ayuda en el símbolo del sistema.

```
aws help
```

Debería aparecer una lista de AWS comandos válidos en la AWS CLI ventana.

Reconfigurar el AWS CLI

Si has descargado y configurado previamente el AWS CLI, es posible que Amazon Polly no esté disponible a menos que vuelvas a configurar el. AWS CLI El procedimiento siguiente comprueba si es necesario.

Para reactivar Amazon Polly desde AWS CLI

1. Comprueba la disponibilidad de Amazon Polly escribiendo el siguiente comando de ayuda en la línea de AWS CLI comandos.

   ```
   aws polly help
   ```

   Si ves una descripción de Amazon Polly y aparece una lista de comandos válidos en la AWS CLI ventana, puedes usar Amazon Polly inmediatamente. AWS CLI En este caso, puede omitir el resto del procedimiento. Si no aparece, continúe en el paso 2.
2. Active Amazon Polly con una de las dos opciones siguientes:

   a. Desinstale y vuelva a instalar el. AWS CLI

      Para las instrucciones, consulte Instalación de la AWS Command Line Interface en la AWS Command Line Interface Guía del usuario.

      o

   b. Descargue el archivo service-2.json.

      En el símbolo del sistema, ejecute el siguiente comando.

      ```
      aws configure add-model --service-model file://service-2.json --service-name
       polly
      ```
3. Vuelva a comprobar la disponibilidad de Amazon Polly.

```
aws polly help
```

La descripción de Amazon Polly debe estar visible.

Ejemplo de síntesis de voz con Amazon Polly

Esta página presenta un breve ejemplo de síntesis de voz realizada en la consola AWS CLI, y con Python. En este ejemplo se realiza la síntesis de voz a partir de texto sin formato, no de SSML.

Console

Síntesis de la voz en la consola

1. Inicia sesión en la consola Amazon Polly AWS Management Console y ábrela en. https://console.aws.amazon.com/polly/
2. Elija la pestaña Text-to-Speech. El campo de texto se cargará con texto de ejemplo para que pueda probar Amazon Polly rápidamente.
3. Desactive SSML.
4. Escriba o pegue el texto en el cuadro de entrada.

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

5. En Motor, elija Generativo, Forma larga, Neural o Estándar.
6. Elija un idioma y una AWS región y, a continuación, elija una voz. (Si selecciona Neural en Motor, solo estarán disponibles los idiomas y las voces compatibles con NTTS. Todas las voces de formato largo y estándar están desactivadas).
7. Si desea escuchar el fragmento hablado inmediatamente, elija Escuchar.
8. Para guardar el fragmento hablado en un archivo, realice una de las operaciones siguientes:
   a. Elija Descargar.
   b. Para cambiar a un formato de archivo diferente, expanda Configuración adicional, active la Configuración del formato del archivo de locución, elija el formato de archivo que desee y, a continuación, elija Descargar.

AWS CLI

En este ejercicio, va a llamar a la operación `SynthesizeSpeech` para pasar el texto de entrada. Puede guardar el audio resultante como un archivo y comprobar su contenido.

1. Ejecute el `synthesize-speech` AWS CLI comando para sintetizar un texto de muestra en un archivo de audio (`hello.mp3`).

   El siguiente AWS CLI ejemplo está formateado para Unix, Linux y macOS. En Windows, sustituya la barra invertida (\) del carácter de continuación de Unix al final de cada línea por un signo de intercalación (^) y utilice comillas completas (") alrededor del texto introducido con comillas simples (') para las etiquetas interiores.

   ```
   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
   ```

   En la llamada a `synthesize-speech`, debe proporcionar un texto de muestra para que lo sintetice la voz que elija. Debe proporcionar un identificador de voz (que se explica en el paso siguiente) y un formato de salida. El comando guarda el audio resultante en el archivo `hello.mp3`. Además del MP3 archivo, la operación envía el siguiente resultado a la consola.

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

2. Reproduzca el archivo `hello.mp3` resultante para comprobar la síntesis de voz.

Python

Para probar el código de ejemplo de Python, necesita AWS SDK for Python (Boto). Para obtener instrucciones, consulte AWS SDK para Python (Boto3).

En este ejemplo de código Python se realizan las siguientes acciones:

- Invoca el AWS SDK for Python (Boto) para enviar una `SynthesizeSpeech` solicitud a Amazon Polly (proporcionando algún texto como entrada).
- Obtiene acceso a la secuencia de audio resultante de la respuesta y la guarda en un archivo (`speech.mp3`) en el disco local.
- Reproduce el archivo de audio con el reproductor predeterminado del sistema local.

Guarde el código en un archivo (example.py) y ejecútelo.

```
"""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])
```

Para obtener ejemplos más detallados, consulte los siguientes temas:

- Uso de SSML en la consola
- Aplicación de lexicones (síntesis del habla)
- Ejemplos de código y aplicaciones para Amazon Polly

Voces en Amazon Polly

Amazon Polly ofrece docenas de voces realistas en diversos idiomas y soporte para una variedad de idiomas. Cada voz se crea con hablantes nativos del idioma, por lo que existen variaciones entre las voces, incluso dentro del mismo idioma. También puedes usar el AWS Management Console para probar cada voz con el texto que elijas. En la mayoría de los casos, habrá al menos una voz masculina y una voz femenina, a menudo más de una de cada. Son pocos los idiomas que solo disponen de una voz.

El inventario de voces y el número de idiomas incluidos se actualiza de manera continua para añadir opciones adicionales. Si desea proponer un nuevo idioma o una nueva voz, no dude en dejar sus comentarios en esta página. Lamentablemente, no podemos hacer ningún comentario sobre los planes para lanzar nuevos idiomas específicos.

> **Note**
>
> Para escuchar voces de ejemplo de Amazon Polly en el navegador, consulte la descripción general del producto Amazon Polly.

Temas

- Voces disponibles
- Voces bilingües
- Aplicación de la voz de presentador
- Cómo escuchar las voces
- Velocidad de las voces
- Cambio de la velocidad de una voz

Voces disponibles

Amazon Polly proporciona voces realistas en varios idiomas para usarlas al sintetizar el habla a partir de texto. En la siguiente tabla se muestran todas las voces que ofrece Amazon Polly.

	Idioma y variantes lingüísticas	Código de idioma	Nombre/ID	Gender	Voz generativa	Voz de formato largo	Voz neuronal	Voz estándar
1	Árabe	arb	Zeina	Mujer	No	No	No	Sí
2	Árabe (Golfo)	ar-AE	Hala*	Mujer	No	No	Sí	No
			Zayd*	Hombre	No	No	Sí	No
3	Neerlandés (belga)	nl-BE	Lisa	Mujer	No	No	Sí	No
4	Catalán	ca-ES	Arlet	Mujer	No	No	Sí	No
5	Checo	cs-CZ	Jitka	Mujer	No	No	Sí	No
6	Chino (cantonés)	yue-CN	Hiujin	Mujer	No	No	Sí	No
7	Chino (mandarín)	cmn-CN	Zhiyu	Mujer	No	No	Sí	Sí
8	Danés	da-DK	Naja	Mujer	No	No	No	Sí
			Mads	Hombre	No	No	No	Sí
			Sofie	Mujer	No	No	Sí	No
9	Neerlandés	nl-NL	Laura	Mujer	No	No	Sí	No
			Lotte	Mujer	No	No	No	Sí
			Ruben	Hombre	No	No	No	Sí

	Idioma y variantes lingüísticas	Código de idioma	Nombre/ ID	Gender	Voz generativa	Voz de formato largo	Voz neuronal	Voz estándar
10	Inglés (Australia)	en-AU	Nicole	Mujer	No	No	No	Sí
			Olivia	Mujer	Sí	No	Sí	No
			Russell	Hombre	No	No	No	Sí
11	Inglés (Reino Unido)	en-GB	Amy**	Mujer	Sí	No	Sí	Sí
			Emma	Mujer	No	No	Sí	Sí
			Brian	Hombre	No	No	Sí	Sí
			Arthur	Hombre	No	No	Sí	No
12	Inglés (India)	en-IN	Aditi*	Mujer	No	No	No	Sí
			Raveena	Mujer	No	No	No	Sí
			Kajal*	Mujer	Sí	No	Sí	No
13	Inglés (Irlanda)	en-IE	Niamh	Mujer	No	No	Sí	No
14	Inglés (Nueva Zelanda)	en-NZ	Aria	Mujer	No	No	Sí	No
15	Inglés (singapurense)	En-SG	Jazmín	Mujer	No	No	Sí	No
16	Inglés (sudafricano)	en-ZA	Ayanda	Mujer	Sí	No	Sí	No

	Idioma y variantes lingüísticas	Código de idioma	Nombre/ID	Gender	Voz generativa	Voz de formato largo	Voz neuronal	Voz estándar
17	English (EE. UU.)	en-US	Danielle	Mujer	Sí	Sí	Sí	No
			Gregory	Hombre	No	Sí	Sí	No
			Ivy	Mujer (niña)	No	No	Sí	Sí
			Joanna**		Sí	No	Sí	Sí
			Kendra	Mujer	No	No	Sí	Sí
			Kimberly	Mujer	No	No	Sí	Sí
			Salli	Mujer	No	No	Sí	Sí
			Joey	Mujer	No	No	Sí	Sí
			Justin	Hombre	No	No	Sí	No
			Kevin	Hombre (niño)	No	No	Sí	Sí
			Matthew*,	Hombre (niño)	Sí	No	Sí	No
			Ruth		Sí	Sí	Sí	No
			Stephen	Hombre	Sí	No	Sí	No
			Patricio	Mujer	No	Sí	No	No
				Hombre				
				Hombre				
18	Inglés (Gales)	en-GB-WLS	Geraint	Hombre	No	No	No	Sí
19	Finés	fi-FI	Suvi	Mujer	No	No	Sí	No

	Idioma y variantes lingüísticas	Código de idioma	Nombre/ID	Gender	Voz generativa	Voz de formato largo	Voz neuronal	Voz estándar
20	Francés	fr-FR	Céline/Celine Léa Mathieu Rémi	Mujer Mujer Hombre Hombre	No Sí No Sí	No No No No	No Sí No Sí	Sí Sí Sí No
21	Francés (belga)	fr-BE	Isabelle	Mujer	No	No	Sí	No
22	Francés (Canadá)	fr-CA	Chantal Gabrielle Liam	Mujer Mujer Hombre	No No No	No No No	No Sí Sí	Sí No No
23	Alemán	de-DE	Marlene Vicki Hans Daniel	Mujer Mujer Hombre Hombre	No Sí No Sí	No No No No	No Sí No Sí	Sí Sí Sí No
24	Alemán (austriaco)	de-AT	Hannah	Mujer	No	No	Sí	No
25	Alemán (suizo)	de-CH	Sabrina	Mujer	No	No	Sí	No

	Idioma y variantes lingüísticas	Código de idioma	Nombre/ID	Gender	Voz generativa	Voz de formato largo	Voz neuronal	Voz estándar
26	Hindi	hi-IN	Aditi*	Mujer	No	No	No	Sí
			Kajal*	Mujer	No	No	Sí	No
27	Islandés	is-IS	Dóra/Dora	Mujer	No	No	No	Sí
			Karl	Hombre	No	No	No	Sí
28	Italiano	it-IT	Carla	Mujer	No	No	No	Sí
			Bianca	Mujer	Sí	No	Sí	Sí
			Giorgio	Hombre	No	No	No	Sí
			Adriano	Hombre	No	No	Sí	No
29	Japonés	ja-JP	Mizuki	Mujer	No	No	No	Sí
			Takumi	Hombre	No	No	Sí	Sí
			Kazuha	Mujer	No	No	Sí	No
			Tomoko	Mujer	No	No	Sí	No
30	Coreano	ko-KR	Seoyeon	Mujer	No	No	Sí	Sí
			Jihye	Mujer	No	No	Sí	No
31	Noruego	nb-NO	Liv	Mujer	No	No	No	Sí
			Ida	Mujer	No	No	Sí	No

	Idioma y variantes lingüísticas	Código de idioma	Nombre/ID	Gender	Voz generativa	Voz de formato largo	Voz neuronal	Voz estándar
32	Polaco	pl-PL	Ewa	Mujer	No	No	No	Sí
			Maja	Mujer	No	No	No	Sí
			Jacek	Hombre	No	No	No	Sí
			Ene	Hombre	No	No	No	Sí
			Ola	Mujer	No	No	Sí	No
33	Portugués (Brasil)	pt-BR	Camila	Mujer	No	No	Sí	Sí
			Vitória/Vitoria	Mujer	No	No	Sí	Sí
			Ricardo	Hombre	No	No	No	Sí
			Thiago	Hombre	No	No	Sí	No
34	Portugués (Europa)	pt-PT	Inês/Ines	Mujer	No	No	Sí	Sí
			Cristiano	Hombre	No	No	No	Sí
35	Rumano	ro-RO	Carmen	Mujer	No	No	No	Sí
36	Ruso	ru-RU	Tatyana	Mujer	No	No	No	Sí
			Maxim	Hombre	No	No	No	Sí

	Idioma y variantes lingüísticas	Código de idioma	Nombre/ID	Gender	Voz generativa	Voz de formato largo	Voz neuronal	Voz estándar
37	Español (España)	es-ES	Conchita	Mujer	No	No	No	Sí
			Lucía	Mujer	Sí	No	Sí	Sí
			Alba	Mujer	No	Sí	No	No
			Enrique	Hombre	No	No	No	Sí
			Sergio	Hombre	Sí	No	Sí	No
			Raúl	Hombre	No	Sí	No	No
38	Español (México)	es-MX	Mia	Mujer	Sí	No	Sí	Sí
			Andrés	Hombre	Sí	No	Sí	No
39	Español (EE. UU.)	es-US	Lupe**	Mujer	Sí	No	Sí	Sí
			Penélope/Penelope	Mujer	No	No	No	Sí
			Miguel	Hombre	No	No	No	Sí
			Pedro	Hombre	Sí	No	Sí	No
40	Sueco	sv-SE	Astrid	Mujer	No	No	No	Sí
			Elin	Mujer	No	No	Sí	No
41	Turco	tr-TR	Filiz	Mujer	No	No	No	Sí
			Burcu	Mujer	No	No	Sí	No
42	Galés	cy-GB	Gwyneth	Mujer	No	No	No	Sí

* Esta voz es bilingüe. Para obtener más información, consulte Voces bilingües.

** Estas voces se pueden utilizar con el estilo de habla Newscaster cuando se usan con el formato neuronal. Para obtener más información, consulte Aplicación de la voz de presentador.

Cada motor de voz de Amazon Polly tiene características únicas. Obtenga más información sobre las características y la disponibilidad regional de los motores de voz que ofrece Amazon Polly:

- Voces generativas
- Voces de formato largo
- Voces neuronales
- Voces estándar

Voces de marca

Además de las voces disponibles que aparecen en la tabla anterior, puede usar Amazon Polly para crear una voz personalizada para el perfil de la marca. Con una voz de marca, puede ofrecer voces únicas y exclusivas a los clientes. Para obtener más información sobre las voces de marca de Amazon Polly, consulte Voz de marca.

Voces bilingües

Amazon Polly tiene dos maneras de producir voces bilingües:

- Voces bilingües con acento
- Voces totalmente bilingües

Voces bilingües con acento

Las voces bilingües con acento se pueden crear usando cualquier voz de Amazon Polly, pero solo cuando se usan etiquetas SSML.

Normalmente, todas las palabras del texto de entrada se pronuncian en el idioma predeterminado de la voz especificada que está utilizando.

Por ejemplo, si está utilizando la voz de Joanna (que habla inglés de Estados Unidos), Amazon Polly pronuncia lo siguiente en la voz de Joanna sin acento francés:

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

En este caso, las palabras Je ne parle pas français se pronuncian como si fueran inglesas.

Si utiliza la voz de Joanna con la etiqueta **<lang>**, Amazon Polly lee la frase en la voz de Joanna en francés con acento americano:

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

Como Joanna no es una hablante nativa del francés, la pronunciación tendrá como base su idioma nativo, inglés de Estados Unidos. Por ejemplo, aunque una pronunciación francesa perfecta pronunciaría la palabra français con una /R/ vibrante uvular, la voz de Joanna, cuyo idioma nativo es el inglés americano, pronuncia este fonema con el sonido /r/ correspondiente.

Si utiliza la voz de Giorgio, que habla italiano, con el siguiente texto, Amazon Polly pronuncia la frase en la voz de Giorgio con pronunciación italiana:

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

Voces totalmente bilingües

Una voz totalmente bilingüe, como Aditi (inglés de India e hindi) puede hablar dos idiomas de forma fluida. Esto le ofrece la capacidad de usar palabras y frases de ambos idiomas en un solo texto usando la misma voz.

Actualmente, Aditi, Kajal, Hala y Zayd son las únicas voces totalmente bilingües disponibles.

Uso de una voz bilingüe (Aditi)

Aditi habla inglés de India (en-IN) e hindi (hi-IN) con fluidez. Puede sintetizar el habla tanto en inglés como en hindi, y la voz puede cambiar entre los dos idiomas incluso dentro de la misma oración.

El idioma hindi se puede usar en dos formas diferentes:

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

Además, es posible mezclar el inglés e hindi de una o ambas formas en una sola oración:

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

Debido a que Aditi es una voz bilingüe, el texto en todos estos casos se leerá correctamente, ya que Amazon Polly puede diferenciar entre los idiomas y los guiones.

Amazon Polly también admite números, fechas, horas y expansión monetaria tanto en inglés (números arábigos) como en hindi (números Devanagari). De forma predeterminada, los números arábigos se leen en inglés de India. Para hacer que Amazon Polly los lea en hindi, debe utilizar el parámetro de código de idioma `hi-IN`.

Aplicación de la voz de presentador

Las personas utilizan diferentes estilos de habla, en función del contexto. Una conversación informal, por ejemplo, suena muy diferente a la de un noticiario de televisión o radio. Debido a la forma de hacer estas voces estándar, no pueden producir distintos estilos de habla. Sin embargo, las voces neuronales sí pueden hacerlo. A continuación, el sistema se entrena para un estilo de habla específico, con las variaciones y el énfasis en determinadas partes del habla inherentes a ese estilo.

Además de las voces neuronales predeterminadas, Amazon Polly ofrece un estilo de voz de presentador de noticias que utiliza el sistema neuronal para generar voz al estilo de un presentador de noticias de radio o televisión. El estilo Newscaster está disponible con las voces de Matthew y Joanna, en inglés estadounidense (en-US), y la voz de Lupe, en español estadounidense (es-US).

Para utilizar el estilo Newscaster, primero elija el motor neuronal y, a continuación, utilice la sintaxis descrita en los pasos siguientes en el texto de entrada.

> **Note**
>
> - Para utilizar cualquier estilo de habla neuronal, debe utilizar una de las regiones de AWS que admitan voces neuronales. Esta opción no está disponible en todas las regiones. Para obtener más información, consulte Compatibilidad con características y regiones.

Console

Aplicación del estilo Presentador

1. Abre la consola Amazon Polly en. https://console.aws.amazon.com/polly/
2. Asegúrese de utilizar una AWS región en la que se admitan las voces neuronales.
3. En la Text-to-Speech página, en Engine, selecciona Neural.
4. Elija el idioma y la voz que desea utilizar. Solo Matthew y Joanna para inglés estadounidense (en-US) y Lupe para español estadounidense (es-US) están disponibles para la voz de presentador.
5. Active SSML.
6. Añada texto de entrada a su text-to-speech solicitud utilizando la sintaxis SSML de estilo Newscaster.

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

Por ejemplo, puede usar la etiqueta presentador de la siguiente manera:

```
<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. Seleccione Escuchar.

AWS CLI

Aplicación del estilo Presentador

1. En la solicitud de API, incluya el parámetro del motor con el valor `neural`:

```
  --engine neural
```

2. Añada texto de entrada a su solicitud de API utilizando la sintaxis SSML de estilo Newscaster.

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

Por ejemplo, puede usar la etiqueta presentador de la siguiente manera:

```
<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>
```

Para obtener más información sobre SSML y , consulte Etiquetas SSML admitidas.

Cómo escuchar las voces

Después de configurar Amazon Polly, puede probar las voces con texto personalizado en la consola.

Audición de las voces de Amazon Polly en la consola

1. Inicia sesión en la consola Amazon Polly AWS Management Console y ábrela en. https://console.aws.amazon.com/polly/
2. Elija la pestaña Text-to-Speech.
3. En Motor, elija Generativo, Forma larga, Neural o Estándar.
4. Seleccione un idioma y una región. A continuación, elija una voz.
5. Introduzca el texto que desea que lea la voz o utilice la frase predeterminada y, a continuación, elija Escuchar.

Velocidad de las voces

Debido a la variación natural existente entre las voces, cada voz disponible habla a velocidades ligeramente diferentes. Por ejemplo, con las voces del idioma inglés de Estados Unidos, Ivy y Joanna son ligeramente más rápidas que Matthew y considerablemente más rápidas que Joey. Dado que existe tanta variación entre las voces, no hay una velocidad estándar (palabras por minuto) disponible para las voces de Amazon Polly. Sin embargo, puede saber el tiempo que tarda la voz en decir el texto seleccionado mediante las marcas de voz.

Medición de la duración de un pasaje de texto hablado

1. Abre el AWS CLI.
2. Ejecute el siguiente código, rellenándolo con datos según sea necesario.

```
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. Abra `LengthOfText.txt`.

Si el texto era "Mary had a little lamb", las últimas líneas devueltas por Amazon Polly serían:

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

El último visema, básicamente el sonido de las letras finales de "lamb" comienza 1082 milisegundos después del comienzo del fragmento de voz. Si bien no es exactamente la duración del audio, es un valor próximo y puede servir de base para la comparación entre voces.

Cambio de la velocidad de una voz

En determinadas aplicaciones, es posible que prefiera que la voz que le gusta sea más lenta o más rápida. Si la velocidad de la voz es un problema, Amazon Polly ofrece la posibilidad de modificarla mediante etiquetas SSML. Por ejemplo, si la organización estaba creando una aplicación que lee libros para un público inmigrante, puede que desee variar la velocidad de la voz. Puede que el público hable inglés, pero con una fluidez limitada. Amazon Polly lo ayuda a ralentizar la velocidad de la voz mediante la etiqueta SSML **<**prosody**>**.

Puede usar un porcentaje:

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

O una velocidad preestablecida:

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

Dispone de dos opciones de velocidad al utilizar SSML con Amazon Polly:

- Velocidades preestablecidas: `x-slow`, `slow`, `medium`, `fast` y `x-fast`. En estos casos, la velocidad de cada opción es aproximada, dependiendo de la voz elegida. La opción `medium` es la velocidad normal de la voz.

- n% de la velocidad de habla: se puede utilizar cualquier porcentaje de la velocidad de habla que esté comprendido entre 20 % y 200 %. En estos casos, puede elegir exactamente la velocidad que desea. Sin embargo, la velocidad real de la voz es aproximada, dependiendo de la voz que haya seleccionado. 100 % se considera la velocidad normal de la voz.

> **Note**
>
> Pruebe la voz que ha seleccionado a varias velocidades. La velocidad de cada opción es aproximada y depende de la voz que elija.

Para obtener más información sobre el uso de la etiqueta `prosody`, consulte Control del volumen, velocidad de habla y tono .

Idiomas en Amazon Polly

Los siguientes idiomas son compatibles con Amazon Polly y se pueden utilizar para sintetizar la voz. Cada idioma tiene un código único. Estos códigos de idioma son etiquetas de identificación de idioma del W3C (*ISO 639-3*para el nombre del idioma y *ISO 3166* para el código de país).

Seleccione un idioma de la siguiente tabla para obtener detalles sobre los fonemas y visemas que proporciona Amazon Polly.

Idioma	Código de idioma
Árabe	arb
Árabe (Golfo)	ar-AE
Catalán	ca-ES
Chino (cantonés)	yue-CN
Chino (mandarín)	cmn-CN
Checo	cs-CZ
Danés	da-DK
Neerlandés (belga)	nl-BE

Idioma	Código de idioma
Neerlandés	nl-NL
Inglés (Australia)	en-AU
Inglés (Reino Unido)	en-GB
Inglés (India)	en-IN
Inglés (Nueva Zeland	en-NZ
Inglés (singapense)	En-SG
Inglés (sudafricano)	en-ZA
English (EE. UU.)	en-US
Inglés (Gales)	en-GB-WLS
Finés	fi-FI

Idioma	Código de idioma
Francé	fr-FR
Francé (belga)	fr-BE
Francé (Canac	fr-CA
Hindi	hi-IN
Alemár	de-DE
Alemár (austria o)	de-AT
Alemár (estánc suizo)	de-CH
Islandé	is-IS
Italiano	it-IT
Japoné	ja-JP
Corear	ko-KR
Norueg	nb-NO
Polaco	pl-PL

Idioma	Código de idioma
Portugu (Brasil)	pt-BR
Portugu (Europ	pt-PT
Roman	ro-RO
Ruso	ru-RU
Españ (Españ	es-ES
Españ (Méxic	es-MX
Españ (EE. UU.)	es-US
Sueco	sv-SE
Turco	tr-TR
Galés	cy-GB

Árabe (arb)

En la tabla siguiente se muestra el conjunto de fonemas del Alfabeto Fonético Internacional (IPA, por sus siglas en inglés), los símbolos del Alfabeto Fonético Extendido SAM (X-SAMPA, por sus

siglas en inglés) y los correspondientes visemas para la voz del idioma árabe de Zeina admitida por Amazon Polly.

Tabla de fonemas/visemas

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Consonantes				
ʔ	?	oclusiva glotal	أنا	
ʕ	?\	fricativa faríngea sonora	عُمَر	k
b	b	oclusiva bilabial sonora	بَلَد	p
d	d	oclusiva alveolar sonora	داري	t
dˤ	d_?\	Oclusiva alveolar sonora enfática	ضَوء	t
ʤ	dZ	africada postalveolar sonora	جَميل	S
ð	D	fricativa dental sonora	ذلكَ	T
ðˤ	D_?\	fricativa dental sonora enfática	ظَالم	T
f	f	fricativa labiodental sorda	فَصل	f
g	g	oclusiva velar sonora	إنجلترا	k
ɣ	G	fricativa velar sonora	غَرب	k

IPA	X-SAMPA	Descripción	Ejemplo	Visema
h	h	fricativa glotal sorda	اذه	k
j	j	aproximante palatal	يشمَي	i
k	k	oclusiva velar sorda	بلَك	k
l	l	aproximante alveolar lateral	ىقال	t
lˠ	l_G	aproximante alveolar lateral enfática	هلدابع	t
m	m	nasal bilabial	اذام	p
n	n	nasal alveolar	رون	t
p	p	oclusiva bilabial sorda	سبَح	p
q	q	oclusiva uvular sorda	بيرَق	k
r	r	vibrante alveolar	لمَر	r
s	s	fricativa alveolar sorda	لاؤُس	s
sˤ	s_?\	fricativa alveolar sorda enfática	بحاص	s
ʃ	S	fricativa postalveolar sorda	رشُك	S

IPA	X-SAMPA	Descripción	Ejemplo	Visema
t	t	oclusiva alveolar sorda	تَمر	t
tˤ	t_?\	oclusiva alveolar sorda enfática	طالِب	t
θ	T	fricativa dental sorda	ثَلاث	T
v	v	fricativa labiodental sonora	فيتامين	f
w	w	aproximante labio-velar	وَلَد	u
x	x	fricativa velar sorda	خَوْف	k
ħ	X\	oclusiva faríngea sonora	حَوْلَ	k
z	z	fricativa alveolar sonora	زُهور	s
Vocales				
a	a	vocal abierta anterior no redondeada	بَرد	a
a:	a:	vocal larga abierta anterior no redondeada	دار	a
ɑˤ	A_?\	vocal abierta posterior no redondeada enfática	طَبل	a

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ɑˤː	A_?\:	vocal larga abierta posterior no redondeada enfática	ظالِم	a
u	u	vocal cerrada posterior redondeada	شُرب	u
uː	u:	vocal larga cerrada posterior redondeada	سور	u
uˤ	u_?\	vocal larga cerrada posterior redondeada enfática	بُدّ	u
uˤː	u_?\:	vocal larga cerrada posterior redondeada enfática	طول	u
i	i	vocal cerrada anterior no redondeada	بِنت	i
iː	i:	vocal larga cerrada anterior no redondeada	حَزين	i
iˤ	i_?\	vocal cerrada anterior no redondeada enfática	ضِدّ	i

IPA	X-SAMPA	Descripción	Ejemplo	Visema
iˤː	i_?\:	vocal larga cerrada anterior no redondeada enfática	يضام	i
e	e	vocal semicerrada anterior no redondeada	تكرام	e
eː	e:	vocal larga semicerrada anterior no redondeada	ليدوم	e
ɔ	O	vocal semiabierta posterior redondeada	يجولونكت	O
ɔː	O:	vocal larga semiabierta posterior redondeada	نويزفيلت	O

Árabe del Golfo (ar-AE)

En la tabla siguiente se muestra el conjunto de fonemas del Alfabeto Fonético Internacional (IPA, por sus siglas en inglés), los símbolos del Alfabeto Fonético Extendido SAM (X-SAMPA, por sus siglas en inglés) y los correspondientes visemas para la voz del idioma árabe de Zeina admitida por Amazon Polly.

Tabla de fonemas/visemas

IPA	X-SAMPA	Descripción	Ejemplo	Pronunciación	Visema
Consonantes					

IPA	X-SAMPA	Descripción	Ejemplo	Pronunciación	Visema
b	b	oclusiva bilabial sonora	دلب	/ " b a . l a d /	b
d	d	oclusiva alveolar sonora	درد	/ " r a d d /	d
dˤ	d_?\	oclusiva alveolar faríngea sonora	ضوء	/ " d_?\ a w ? /	D
f	f	fricativa labiodental sorda	نرف	/ " f l . r l n /	f
g	g	oclusiva velar sonora	قال	/ " g a: l /	k
j	j	aproximante palatal sonora	يشميي	/ " j l m . S i: /	i
k	k	oclusiva velar sorda	لماك	/ " k a: . m i l /	k
l	l	aproximante lateral alveolar sonora	ليل	/ " l e: l /	t
l	l_G	aproximante lateral alveolar faríngea sonora	هلدابع	/ ?\ a b . " d A_?\ l_G . l_G A_?\ /	t
m	m	oclusiva nasal bilabial	مئة	/ " m l j . j a /	p

IPA	X-SAMPA	Descripción	Ejemplo	Pronunciación	Visema
n	n	oclusiva nasal alveolar	نور	/ " n u: r /	t
p	p	oclusiva bilabial sorda	أبرا	/ " ? O . p e . r a: /	p
q	q	oclusiva uvular sorda	قرص	/ " q A_?\ s_?\ r /	k
r	r	vibrante alveolar	رمل	/ " r a . m l l /	r
s	s	fricativa alveolar sorda	سمسم	/ " s l m . s l m /	s
sˤ	s_?\	fricativa alveolar sorda enfática	صاحب	/ " s_?\ A_?: . X \ l b /	s
t	t	oclusiva alveolar sorda	تمر	/ "t a . m a r /	t
tˤ	t_?\	fricativa alveolar sorda enfática	طالب	/ " t_?\ A_?: . l l b /	t
v	v	fricativa labiodental sonora	فيتامين	/ v i: . t A . " m i: n /	f
w	w	aproximante labiovelar sonora	وايد	/ " w a: . j l d /	u
x	x	fricativa velar sorda	خروف	/ x a . " r u: f /	k

IPA	X-SAMPA	Descripción	Ejemplo	Pronunciación	Visema
z	z	fricativa velar sorda	زهور	/ " z h u: r /	s
ð	D	fricativa interdental sonora	ذلك	/ " D a: . l l k /	D
ðˤ	D_?\	fricativa interdental faríngea sonora	ظلام	/ D_?\ A_?\ . " l a: m /	D
ħ	X\	oclusiva faríngea sonora	الجين	/ ? a l . " X\ i: n /	k
ŋ	N	oclusiva velar nasal	هونغ كونغ	/ h O N . " k O N g /	k
ɣ	G	fricativa velar sonora	غريبة	/ G I . " r i: . b a /	k
ʃ	S	fricativa postalveolar sorda	شمس	/ " S a m s /	S
ʒ	Z	fricativa postalveolar sonora	جاكيت	/ Z a . " k e: t /	S
ʔ	?	oclusiva glotal	مؤسسة	/ m u . " ? a s . s a . s a /	
ʕ	?\	fricativa faríngea sonora	عام	/ " ?\ a: m m /	k

IPA	X-SAMPA	Descripción	Ejemplo	Pronunciación	Visema
ʤ	dZ	africada postalveolar sonora	جامعة	/ " dZ a: m . ?\ a /	S
θ	T	fricativa interdental sonora	ثالثة	/ T a . " l a: . T a /	T
ɦ	h	fricativa glotal sonora	هلال	/ " h l a: l /	k
Vocales					
æ	a	vocal larga semiabierta anterior no redondeada	فرس	/ " s a . f a r /	a
ɑˤ	A_?\	vocal corta abierta posterior no redondeada enfática	صلب	/ " s_?\ A_?\ l b /	a
æː	a:	vocal larga semiabierta anterior no redondeada	باب	/ " b a: b /	a
ɑˤː	A_?\:	vocal larga abierta posterior no redondeada enfática	ضان	/ " n A_?: . D_? \ i_?\ dZ /	a

IPA	X-SAMPA	Descripción	Ejemplo	Pronunciación	Visema
a	A	vocal abierta central no redondeada	wifi	/ " w A j . f A j /	a
i	i	vocal corta no redondeada anterior cerrada (MSA)	إسحاق	/ ? i s . " X\ A_? \: q /	i
ɪ	I	vocal cerrada anterior no redondeada	بنت	/ " b I n t /	i
iˤ	i_?\	vocal corta cerrada anterior no redondeada enfática	لطف	/ " t_?\ i_?\ f I l /	i
iː	i:	vocal cerrada anterior no redondeada	ليبس	/ s a . " b i: l /	i
iˤː	i_?:	vocal larga cerrada anterior no redondeada enfática	بيطر	/ r A_?\ . " t_?\ i_?: b /	i
u	u	vocal corta redondeada posterior (MSA)	عرتخم	/ " m u x . t a . r i ?\ /	u

IPA	X-SAMPA	Descripción	Ejemplo	Pronunciación	Visema
ʊ	U	vocal cerrada posterior redondeada	رسوم	/ r U . " s u: m /	u
uˤ	u_?\	vocal corta cerrada posterior redondeada enfática	عصفور	/ ?\ u_?\ s_?\ . " f u: r /	u
u:	u:	vocal larga cerrada posterior redondeada	توت	/ " t u: t /	u
uˤː	u_?\:	vocal larga cerrada posterior redondeada enfática	صور	/ " s_?\ u_?\: r /	u
e	e	vocal semianterior no redondeada	إِنْتَرْنِت	/ " s e n t /	e
e:	e:	vocal larga semianterior no redondeada	إيش	/ " ? e: S /	e
ɔ	O	vocal semiabierta posterior redondeada	دولار	/ d O . " l A r /	O

IPA	X-SAMPA	Descripción	Ejemplo	Pronunciación	Visema
ɔː	O:	vocal larga semiabierta posterior redondeada	لون	/ " l O: n /	O

Catalán (ca-ES)

En la tabla siguiente se muestra el conjunto de fonemas del Alfabeto Fonético Internacional (IPA, por sus siglas en inglés), los símbolos del Alfabeto Fonético Extendido SAM (X-SAMPA, por sus siglas en inglés) y los correspondientes visemas para la voz del idioma árabe de Zeina admitida por Amazon Polly.

Tabla de fonemas/visemas

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Consonantes				
p	p	oclusiva bilabial sorda	ploure	p
t	t	oclusiva alveolar sorda	Tarragona	t
k	k	oclusiva velar sorda	com	k
b	b	oclusiva bilabial sonora	bata	p
d	d	oclusiva alveolar sonora	endoll	t
g	g	oclusiva velar sonora	gros	k

IPA	X-SAMPA	Descripción	Ejemplo	Visema
m	m	nasal bilabial sonora	manera	p
n	n	nasal alveolar sonora	donar	t
ɲ	J	nasal palatina sonora	any	J
ŋ	N	nasal velar sonora	pingüí	k
ɫ	5	aproximante lateral alveolar velarizada (l oscura)	albercoc	l
ʎ	L	aproximante palatal lateral sonora	llop	J
r	r	vibrante alveolar sonora	parra	r
ɾ	4	Tap alveolar sonora	para	t
f	f	fricativa labiodental sorda	èmfasi	f
s	s	fricativa alveolar sorda	sac	s
z	z	fricativa alveolar sonora	calzes	s
ʃ	S	fricativa postalveo lar sorda	guix	S

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ʒ	Z	fricativa postalveolar sonora	col·legi	S
͡tʃ	tS	africada postalveolar sorda	cotxe	S
͡dʒ	dZ	africada postalveolar sonora	platja	S
β	B	aproximante bilabial sonora	obert	B
ð	D	aproximante dental sonora	bedoll	T
j	j	aproximante palatal sonora	noia	i
ɣ	G	aproximante velar sonora	pega	k
v	v	fricativa labiodental sonora	afgà	f
w	w	aproximante labiovelar sonora	aigua	u
x	x	fricativa velar sorda	Jiménez	k
ʝ	j\	fricativa palatal sonora	yeso	J
l	l	aproximante lateral alveolar sonora	alondra	t
θ	T	fricativa dental sorda	González	T

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Vocales				
a	a	Vocal posterior abierta	casa	a
e	e	vocal semicerrada anterior no redondeada	llenya	e
ɛ	E	vocal semiabierta anterior no redondeada	xec	E
i	i	vocal cerrada anterior no redondeada	visca	i
o	o	vocal semicerrada posterior redondeada	gos	o
ɔ	O	vocal semiabierta posterior redondeada	joc	O
u	u	vocal cerrada posterior redondeada	un	u
ə	@	vocal media central	casa	@
Símbolos adicionales				
ˈ	"	acento principal	Alabama	
ˌ	%	acento secundario	Alabama	
.	.	límite de sílaba	A.la.ba.ma	

Chino (cantonés) (yue-CN)

En la tabla siguiente se muestran los fonemas de jyutping y el Sistema Fonético Internacional (IPA, por sus siglas en inglés) para la voz de chino mandarín admitida por Amazon Polly. El jyutping es un sistema de romanización del cantonés que se usa comúnmente en el mundo académico y entre los hablantes de cantonés. IPA y X-SAMPA no se suelen utilizar, pero están disponibles para la compatibilidad con el inglés. Los símbolos IPA y X-SAMPA de la tabla son únicamente una referencia y no deben utilizarse para transcripción en chino. También se muestran ejemplos de jyutping y los visemas correspondientes.

Para hacer que Amazon Polly use la pronunciación fonética con jyutping, use la etiquetaphoneme `alphabet="x-amazon-jyutping"`.

Los siguientes ejemplos muestran esto con cada estándar.

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 solo acepta entradas en cantonés codificadas en UTF-8.

Tabla de fonemas/visemas

Jyutping	IPA	X-SAMPA	Descripción	Ejemplo de jyutping	Visema
Consonantes					
b	p	p	oclusiva bilabial sorda	巴, baa1	p
c	ts^h	ts_h	africada alveolar aspirada sorda	叉, caa1	s
d	t	t	oclusiva alveolar sorda	打, daa2	t
f	f	f	fricativa labiodental sorda	花, faa1	f
g	k	k	oclusiva velar sorda	家, gaa1	k
gw	k^w	k_w	oclusiva velar palatal sorda	瓜, gwaa1	u
h	h	h	fricativa glotal sorda	哈, haa1	k
k	k^h	k_h	oclusiva velar aspirada sorda	卡, kaa1	k
kw	k^{wh}	k_wh	labializada oclusiva velar aspirada sorda	誇, kwaa1	u
l	l	l	aproximante alveolar lateral	啦, laa1	t
m	m	m	nasal bilabial	媽, maa1	p
m	m	m=	nasal bilabial silábica	唔, m4	p
ng	ŋ	N	nasal velar	牙, ngaa4	k
ng	ŋ	N=	velar nasal silábica	吳, ng4	k

Jyutping	IPA	X-SAMPA	Descripción	Ejemplo de jyutping	Visema
n	n	n	nasal alveolar	拿, naa4	t
p	p^h	p_h	oclusiva bilabial aspirada sorda	趴, paa1	p
s	s	s	fricativa alveolar sorda	沙, saa1	s
t	t^h	t_h	oclusiva alveolar aspirada sorda	他, taa1	t
w	w	w	aproximante labio-velar	娃, waa1	u
y	j	j	aproximante palatal	也, jaa5	i
z	ts	ts	africada alveolar sorda	渣, zaa1	s
Vocales					
a	ɐ	6	vocal casi abierta central	吉, gat1	a
aa	ɑ	A	vocal abierta posterior no redondeada	家, gaa1	a
aai	ɑi	Ai	diptongo	街, gaai1	a
aau	ɑu	Au	diptongo	交, gaau1	a
ai	ɐi	6i	diptongo	雞, gai1	a
au	ɐu	6u	diptongo	溝, kau1	a
e	ɛ	E	vocal semiabierta anterior no redondeada	爹, de1	E
ei	ei	ei	diptongo	基, gei1	e
eo	ɵ	8	vocal semicerrada central redondeada	春, ceon1	o

Jyutping	IPA	X-SAMPA	Descripción	Ejemplo de jyutping	Visema
eoi	ey	8y	diptongo	居, geoi1	o
eu	ɛu	Eu	diptongo	掉 in 掉垃圾, deu6	E
i	i	i	vocal cerrada anterior no redondeada	斯, si1	i
i	I	I	vocal casi cerrada semianterior no redondeada	激, gik1	i
iu	iu	iu	diptongo	驕, giu1	i
o	ɔ	O	vocal semiabierta posterior redondeada	哥, go1	O
oe	œ	9	vocal semiabierta anterior redondeada	鋸, goe3	O
oi	ɔi	Oi	diptongo	該, goi1	O
ou	ou	ou	diptongo	高, gou1	o
u	u	u	vocal semicerrada posterior redondeada	姑, gu1	u
u	ʊ	U	vocal casi cerrada semiposterior redondeada	谷, guk5	u
ui	ui	ui	diptongo	劧, gui6	u
yu	y	y	vocal cerrada anterior redondeada	於, jyu1	u
Marcas tonales y símbolos adicionales					

Jyutping	IPA	X-SAMPA	Descripción	Ejemplo de jyutping	Visema
1			nivel alto	詩, si1	
2			media elevación	史, si2	
3			nivel medio	試, si3	
4			nivel muy bajo	時, si4	
5			baja elevación	市, si5	
6			nivel bajo	是, si6	
-	.	.	límite de sílaba	語音 jyu5-jam1	

Chino, mandarín (cmn-CN)

En la tabla siguiente se muestran los fonemas de pinyin y el Sistema Fonético Internacional (IPA, por sus siglas en inglés) para la voz de chino mandarín admitida por Amazon Polly. Pinyin es el estándar internacional para la latinización del chino estándar. IPA y X-SAMPA no se suelen utilizar, pero están disponibles para la compatibilidad con el inglés. Los símbolos IPA y X-SAMPA de la tabla son únicamente una referencia y no deben utilizarse para transcripción en chino. También se muestran ejemplos de pinyin y los visemas correspondientes.

Para que Amazon Polly utilice la pronunciación fonética de pinyin, utilice la etiquetaphoneme `alphabet="x-amazon-`*`phonetic standard used`*"`.`

Los siguientes ejemplos muestran esto con cada estándar.

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 solo acepta la entrada de chino mandarín codificada en UTF-8. Amazon Polly no admite actualmente el estándar de codificación GB 18030.

Tabla de fonemas/visemas

Pinyin	IPA	X-SAMPA	Descripción	Ejemplo de pinyin	Visema
Consonantes					
f	f	f	fricativa labiodental sorda	发, fa1	f
h	h	h	fricativa glotal sorda	和, he2	k
g	k	k	oclusiva velar sorda	古, gu3	k
k	kʰ	k_h	oclusiva velar aspirada sorda	苦, ku3	k
l	l	l	aproximante alveolar lateral	拉, la1	t
m	m	m	nasal bilabial	骂, ma4	p

Pinyin	IPA	X-SAMPA	Descripción	Ejemplo de pinyin	Visema
n	n	n	nasal alveolar	那, na4	t
ng	ŋ	N	nasal velar	正, zheng4	k
b	p	p	oclusiva bilabial sorda	爸, ba4	p
p	pʰ	p_h	oclusiva bilabial aspirada sorda	怕, pa4	p
s	s	s	fricativa alveolar sorda	四, si4	s
x	ɕ	s\	fricativa alveolo-palatal sorda	西, xi1	J
sh	ʂ	s`	fricativa retrofleja sorda	是, shi4	S
d	t	t	oclusiva alveolar sorda	打, da3	t
t	tʰ	t_h	oclusiva alveolar aspirada sorda	他, ta1	t
zh	͡tʂ	t`s`	africada retrofleja sorda	之, zhi1	S
ch	͡tʂʰ	t`s`_h	africada retrofleja aspirada sorda	吃, chi1	S
s	͡ts	ts	africada alveolar sorda	字, zi4	s
j	͡tɕ	ts\	africada alveolo-palatal sorda	鸡, ji1	J
q	͡tɕʰ	ts_h	africada alveolo-palatal aspirada sorda	七, qi1	J
c	͡tsʰ	ts_h	africada alveolar aspirada sorda	次, ci4	s
w	w	w	aproximante labio-velar	我, wo3	u

Pinyin	IPA	X-SAMPA	Descripción	Ejemplo de pinyin	Visema
r	ʐ	z`	fricativa retrofleja sonora	日, ri4	S
"sílabas enfatizadas er" y "r"					
er	ɚ	@`	vocal media central rótica	二, er4	@
-r			sílaba rótica	馅儿, xianr4	@
Vocales					
e	ɤ	7	vocal semicerrada posterior no redondeada	恶, e4	e
e	ə	@	vocal media central	恩, en1	@
a	a	a	vocal abierta anterior no redondeada	安, an1	a
ai	aɪ	aI	diptongo	爱, ai4	a
ao	aʊ	aU	diptongo	奥, ao4	a
ei	eɪ	e	diptongo	诶, ei4	e
e	ɛ	E	vocal semiabierta anterior no redondeada	姐, jie3	E
i	i	i	vocal cerrada anterior no redondeada	鸡, ji1	i
ou	oʊ	oU	diptongo	欧, ou1	o
o	ɔ	O	vocal semiabierta posterior redondeada	哦, o4	o
u	u	u	vocal semicerrada posterior redondeada	主, zhu3	u

Pinyin	IPA	X-SAMPA	Descripción	Ejemplo de pinyin	Visema
yu	y	y	vocal cerrada anterior redondeada	于, yu2	u
Marcas tonales y símbolos adicionales					
1			tono alto	淤, yu1	
2			tono ascendente	鱼, yu2	
3			tono bajo (descendente)	语, yu3	
4			tono descendente	育, yu4	
0			tono neutro	的, de0	
-	.	.	límite de sílaba	语音 yu3-yin1	

Checo (cs-CZ)

En la tabla siguiente se muestra el conjunto de fonemas del alfabeto fonético internacional (IPA, por sus siglas en inglés), los símbolos del alfabeto fonético extendido SAM (X-SAMPA, por sus siglas en inglés) y los correspondientes visemas para las voces del idioma polaco admitidas por Amazon Polly.

Tabla de fonemas/visemas

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Consonantes				
p	p	oclusiva bilabial sorda	pes	p
t	t	oclusiva alveolar sorda	tok	t
c	c	oclusiva palatal sorda	t’uk	J

IPA	X-SAMPA	Descripción	Ejemplo	Visema
k	k	oclusiva velar sorda	kos	k
b	b	oclusiva bilabial sonora	bez	p
d	d	oclusiva alveolar sonora	dok	t
ɟ	J\	oclusiva palatal sonora	d’as	J
g	g	oclusiva velar sonora	gum	k
f	f	fricativa labiodental sorda	film	f
v	v	fricativa labiodental sonora	ves	f
s	s	fricativa alveolar sorda	sen	s
z	z	fricativa alveolar sonora	zel	s
ʃ	S	fricativa postalveolar sorda	šel	S
ʒ	Z	fricativa postalveolar sonora	žen	S
x	x	fricativa velar sorda	chat	k
ɦ	h	fricativa glotal sonora	hus	k

IPA	X-SAMPA	Descripción	Ejemplo	Visema
t͡s	ts	africada alveolar sorda	co	s
t͡ʃ	tS	africada postalveolar sorda	čin	S
d͡z	dz	africada alveolar sonora	špicberský	s
d͡ʒ	dZ	africada alveolar sorda	džin	S
m	m	nasal bilabial	mor	p
n	n	nasal alveolar	nos	t
ɲ	J	nasal palatal	ňader	J
ŋ	N	nasal velar	banka	k
r	r	vibrante alveolar sonora	rys	r
r̝	r_r	vibrante fricativa alveolar elevada sonora	řez	r
r̝̊	r_0_r	vibrante fricativa alveolar elevada sorda	keř	r
l	l	aproximante alveolar lateral	les	t
j	j	aproximante palatal	jen	i

IPA	X-SAMPA	Descripción	Ejemplo	Visema
w	w	aproximante labiovelar	Watson	u
r̩	r_=	vibrante alveolar sonora silábica	krk	r
l̩	l_=	aproximante alveolar lateral silábica	vlna	t
Vocales				
a	a	vocal abierta anterior no redondeada	lan	a
aː	a:	vocal larga abierta anterior no redondeada	lán	a
ɛ	E	vocal semiabier ta anterior no redondeada	let	E
ɛː	E:	vocal larga semiabierta anterior no redondeada	lét	E
ɪ	I	vocal casi cerrada semianterior no redondeada	bit	i
iː	i:	vocal larga cerrada anterior no redondeada	bít	i

IPA	X-SAMPA	Descripción	Ejemplo	Visema
o	o	vocal semicerrada posterior redondeada	hol	o
oː	o:	vocal larga semicerrada posterior redondeada	gól	o
u	u	vocal semicerrada posterior redondeada	pul	u
uː	u:	vocal larga cerrada posterior redondeada	půl	u
‿au	au	diptongo	auto	a
‿ɛu	Eu	diptongo	euro	E
‿ou	ou	diptongo	mouk	o
Símbolos adicionales				
ˈ	“	acento principal		
.	.	límite de sílaba		

Danés (da-DK)

En la tabla siguiente se muestra el conjunto de fonemas del Alfabeto Fonético Internacional (IPA, por sus siglas en inglés), los símbolos del Alfabeto Fonético Extendido SAM (X-SAMPA, por sus siglas en inglés) y los correspondientes visemas para las voces del idioma danés admitidas por Amazon Polly.

Tabla de fonemas/visemas

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Consonantes				
b	b	oclusiva bilabial sonora	bat	p
d	d	oclusiva alveolar sonora	da	t
ð	D	fricativa dental sonora	mad, thriller	T
f	f	fricativa labiodental sorda	fat	f
g	g	oclusiva velar sonora	gat	k
h	h	fricativa glotal sorda	hat	k
j	j	aproximante palatal	jo	i
k	k	oclusiva velar sorda	kat	k
l	l	aproximante alveolar lateral	ladt	t
m	m	nasal bilabial	mat	p
n	n	nasal alveolar	nay	t
ŋ	N	nasal velar	lang	k
p	p	oclusiva bilabial sorda	pande	p

IPA	X-SAMPA	Descripción	Ejemplo	Visema
r	r	vibrante alveolar	thriller, story	r
ʁ	R	fricativa uvular sonora	rat	k
s	s	fricativa alveolar sorda	sat	s
t	t	oclusiva alveolar sorda	tal	t
v	v	fricativa labiodental sonora	vat	f
w	w	aproximante velo-labial	hav, weekend	u
Vocales				
ø	2	vocal semicerrada anterior redondeada	øst	o
ø:	2:	vocal larga semicerrada anterior redondeada	øse	o
ɐ	6	vocal casi abierta central	mor	a
œ	9	vocal semiabierta anterior redondeada	skøn, grønt	O

IPA	X-SAMPA	Descripción	Ejemplo	Visema
œ:	9:	vocal larga semiabierta anterior redondeada	høne, gøre	O
ə	@	vocal media central	ane	@
æ:	{:	vocal larga casi abierta anterior no redondeada	male	a
a	a	vocal abierta anterior no redondeada	man	a
æ	{	vocal casi abierta anterior no redondeada	adresse	a
ɑ	A	vocal abierta posterior no redondeada	lak, tak	a
ɑ:	A:	vocal larga abierta posterior no redondeada	rase	a
e	e	vocal semicerrada anterior no redondeada	midt	e
e:	e:	vocal larga semicerrada anterior no redondeada	mele	e

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ɛ	E	vocal semiabierta anterior no redondeada	mæt	E
ɛː	E:	vocal larga semiabierta anterior no redondeada	mæle	E
i	i	vocal cerrada anterior no redondeada	mit	i
iː	i:	vocal larga cerrada anterior no redondeada	mile	i
o	o	vocal semicerrada posterior redondeada	foto	o
oː	o:	vocal larga semicerrada posterior redondeada	mole	o
ɔ	O	vocal semiabierta posterior redondeada	mund	O
ɔː	O:	vocal larga semiabierta posterior redondeada	måle	O

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ɒː	Q:	vocal larga abierta posterior redondeada	morse	O
u	u	vocal semicerrada posterior redondeada	lusk	u
uː	u:	vocal larga cerrada posterior redondeada	mule	u
ʌ	V	semiabierta posterior no redondeada	kører	E
y	y	vocal cerrada anterior redondeada	yt	u
y:	y:	vocal larga cerrada anterior redondeada	hyle	u
Símbolos adicionales				
ˈ	"	acento principal	Alabama	
ˌ	%	acento secundario	Alabama	
.	.	límite de sílaba	A.la.ba.ma	

Neerlandés (belga) (NL-BE)

En la tabla siguiente se muestra el conjunto de fonemas del Alfabeto Fonético Internacional (IPA, por sus siglas en inglés), los símbolos del Alfabeto Fonético Extendido SAM (X-SAMPA, por sus siglas

en inglés) y los correspondientes visemas para las voces del idioma portugués de Brasil admitidas por Amazon Polly.

Tabla de fonemas/visemas

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Consonantes				
b	b	oclusiva bilabial sonora	bak	p
d	d	oclusiva alveolar sonora	dak	t
ʤ	dZ	africada postalveolar sonora	manager	S
f	f	fricativa labiodental sorda	fel	f
g	g	oclusiva velar sonora	goal	k
ɣ	G	fricativa velar sonora	hoed	k
ɦ	h\	fricativa glotal sonora	hand	k
j	j	aproximante palatal	ja	i
k	k	oclusiva velar sorda	kap	k
l	l	aproximante alveolar lateral	land	t
m	m	nasal bilabial	met	p

IPA	X-SAMPA	Descripción	Ejemplo	Visema
n	n	nasal alveolar	net	t
ŋ	N	nasal velar	bang	k
p	p	oclusiva bilabial sorda	pak	p
r	r	vibrante alveolar	rand	r
s	s	fricativa alveolar sorda	sein	s
ʃ	S	fricativa postalveolar sorda	show	S
t	t	oclusiva alveolar sorda	tak	t
v	v	fricativa labiodental sonora	vel	f
ʋ	v\	aproximante labiodental	wit	f
x	x	fricativa velar sorda	toch	k
z	z	fricativa alveolar sonora	ziin	s
ʒ	Z	fricativa postalveolar sonora	bagage	S
Vocales				
øː	2:	vocal larga semicerrada anterior redondeada	neus	o

IPA	X-SAMPA	Descripción	Ejemplo	Visema
œy	9y	diptongo	buit	O
ə	@	vocal media central	de	@
a:	a:	vocal larga abierta anterior no redondeada	baad	a
ɑ:	A	vocal abierta posterior no redondeada	bad	a
e:	e:	vocal larga semicerrada anterior no redondeada	beet	e
ɜː	3:	vocal larga semiabierta central no redondeada	barrière	E
ɛ	E	vocal semiabier ta anterior no redondeada	bed	E
ɛi	Ei	diptongo	beet	E
i	i	vocal cerrada anterior no redondeada	vier	i
ɪ	I	vocal casi cerrada semianterior no redondeada	pit	i

IPA	X-SAMPA	Descripción	Ejemplo	Visema
o:	o:	vocal larga semicerrada posterior redondeada	boot	o
ɔ	O	vocal semiabierta posterior redondeada	pot	O
u	u	vocal semicerrada posterior redondeada	hoed	u
ʌu	Vu	diptongo	fout	E
y:	y:	vocal larga cerrada anterior redondeada	fuut	u
ʏ	S	vocal casi cerrada semianterior redondeada	hut	u
Símbolos adicionales				
ˈ	"	acento principal	Alabama	
ˌ	%	acento secundario	Alabama	
.	.	límite de sílaba	A.la.ba.ma	

Neerlandés (nl-NL)

En la tabla siguiente se muestra el conjunto de fonemas del Alfabeto Fonético Internacional (IPA, por sus siglas en inglés), los símbolos del Alfabeto Fonético Extendido SAM (X-SAMPA, por sus siglas

en inglés) y los correspondientes visemas para las voces del idioma holandés admitidas por Amazon Polly.

Tabla de fonemas/visemas

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Consonantes				
b	b	oclusiva bilabial sonora	bak	p
d	d	oclusiva alveolar sonora	dak	t
ʤ	dZ	africada postalveolar sonora	manager	S
f	f	fricativa labiodental sorda	fel	f
g	g	oclusiva velar sonora	goal	k
ɣ	G	fricativa velar sonora	hoed	k
ɦ	h\	fricativa glotal sonora	hand	k
j	j	aproximante palatal	ja	i
k	k	oclusiva velar sorda	kap	k
l	l	aproximante alveolar lateral	land	t
m	m	nasal bilabial	met	p

IPA	X-SAMPA	Descripción	Ejemplo	Visema
n	n	nasal alveolar	net	t
ŋ	N	nasal velar	bang	k
p	p	oclusiva bilabial sorda	pak	p
r	r	vibrante alveolar	rand	r
s	s	fricativa alveolar sorda	sein	s
ʃ	S	fricativa postalveolar sorda	show	S
t	t	oclusiva alveolar sorda	tak	t
v	v	fricativa labiodental sonora	vel	f
ʋ	v\	aproximante labiodental	wit	f
x	x	fricativa velar sorda	toch	k
z	z	fricativa alveolar sonora	ziin	s
ʒ	Z	fricativa postalveolar sonora	bagage	S
Vocales				
øː	2:	vocal larga semicerrada anterior redondeada	neus	o

IPA	X-SAMPA	Descripción	Ejemplo	Visema
œy	9y	diptongo	buit	O
ə	@	vocal media central	de	@
a:	a:	vocal larga abierta anterior no redondeada	baad	a
ɑ:	A	vocal abierta posterior no redondeada	bad	a
e:	e:	vocal larga semicerrada anterior no redondeada	beet	e
ɜː	3:	vocal larga semiabierta central no redondeada	barrière	E
ɛ	E	vocal semiabier ta anterior no redondeada	bed	E
ɛi	Ei	diptongo	beet	E
i	i	vocal cerrada anterior no redondeada	vier	i
ɪ	I	vocal casi cerrada semianterior no redondeada	pit	i

IPA	X-SAMPA	Descripción	Ejemplo	Visema
oː	o:	vocal larga semicerrada posterior redondeada	boot	o
ɔ	O	vocal semiabierta posterior redondeada	pot	O
u	u	vocal semicerrada posterior redondeada	hoed	u
ʌu	Vu	diptongo	fout	E
yː	y:	vocal larga cerrada anterior redondeada	fuut	u
ʏ	S	vocal casi cerrada semianterior redondeada	hut	u
Símbolos adicionales				
ˈ	"	acento principal	Alabama	
ˌ	%	acento secundario	Alabama	
.	.	límite de sílaba	A.la.ba.ma	

Inglés (EE. UU.) (en-US)

En la tabla siguiente se muestra el conjunto de fonemas del Alfabeto Fonético Internacional (IPA, por sus siglas en inglés), los símbolos del Alfabeto Fonético Extendido SAM (X-SAMPA, por sus siglas

en inglés) y los correspondientes visemas para las voces del idioma inglés americano admitidas por Amazon Polly.

Tabla de fonemas/visemas

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Consonantes				
b	b	oclusiva bilabial sonora	bed	p
d	d	oclusiva alveolar sonora	dig	t
ʤ	dZ	africada postalveolar sonora	jump	S
ð	D	fricativa dental sonora	then	T
f	f	fricativa labiodental sorda	five	f
g	g	oclusiva velar sonora	game	k
h	h	fricativa glotal sorda	house	k
j	j	aproximante palatal	yes	i
k	k	oclusiva velar sorda	cat	k
l	l	aproximante alveolar lateral	lay	l
m	m	nasal bilabial	mouse	p

IPA	X-SAMPA	Descripción	Ejemplo	Visema
n	n	nasal alveolar	nap	t
ŋ	N	nasal velar	thing	k
p	p	oclusiva bilabial sorda	speak	p
ɹ	r\	aproximante alveolar	red	r
s	s	fricativa alveolar sorda	seem	s
ʃ	S	fricativa postalveolar sorda	ship	S
t	t	oclusiva alveolar sorda	trap	t
t͡ʃ	tS	africada postalveolar sorda	chart	S
θ	T	fricativa dental sorda	thin	T
v	v	fricativa labiodental sonora	vest	f
w	w	aproximante velo-labial	west	u
z	z	fricativa alveolar sonora	zero	s
ʒ	Z	fricativa postalveolar sonora	vision	S
Vocales				

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ə	@	vocal media central	arena	@
ɚ	@`	vocal media central rótica	reader	@
æ	{	vocal casi abierta anterior no redondeada	trap	a
aɪ	aI	diptongo	price	a
aʊ	aU	diptongo	mouth	a
ɑ	A	vocal larga abierta posterior no redondeada	father	a
eɪ	eI	diptongo	face	e
ɝ	3`	vocal media central rótica redondeada abierta	nurse	E
ɛ	E	vocal abierta semianterior no redondeada	dress	E
i	i	vocal larga cerrada anterior no redondeada	fleece	i
ɪ	I	vocal casi cerrada semianterior no redondeada	kit	i
oʊ	oU	diptongo	goat	o

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ɔ	O	vocal abierta posterior redondeada larga	thought	O
ɔɪ	OI	diptongo	choice	O
u	u	vocal larga cerrada posterior redondeada	goose	u
ʊ	U	vocal casi cerrada semiposterior redondeada	foot	u
ʌ	V	open-mid-back vocal no redondeada	strut	E
Símbolos adicionales				
ˈ	"	acento principal	Alabama	
ˌ	%	acento secundario	Alabama	
.	.	límite de sílaba	A.la.ba.ma	

Inglés (Australia) (en-AU)

En la tabla siguiente se muestra el conjunto de fonemas del Alfabeto Fonético Internacional (IPA, por sus siglas en inglés), los símbolos del Alfabeto Fonético Extendido SAM (X-SAMPA, por sus siglas en inglés) y los correspondientes visemas para las voces del idioma inglés de Australia admitidas por Amazon Polly.

Tabla de fonemas/visemas

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Consonantes				
b	b	oclusiva bilabial sonora	bed	p
d	d	oclusiva alveolar sonora	dig	t
͡dʒ	dZ	africada postalveolar sonora	jump	S
ð	D	fricativa dental sonora	then	T
f	f	fricativa labiodental sorda	five	f
g	g	oclusiva velar sonora	game	k
h	h	fricativa glotal sorda	house	k
j	j	aproximante palatal	yes	i
k	k	oclusiva velar sorda	cat	k
l	l	aproximante alveolar lateral	lay	t
l̩	l=	aproximante alveolar lateral silábica	battle	t

IPA	X-SAMPA	Descripción	Ejemplo	Visema
m	m	nasal bilabial	mouse	p
m̩	m=	nasal bilabial silábica	anthem	p
n	n	nasal alveolar	nap	t
n̩	n=	nasal alveolar silábica	botón	t
ŋ	N	nasal velar	thing	k
p	p	oclusiva bilabial sorda	pin	p
ɹ	r\	aproximante alveolar	red	r
s	s	fricativa alveolar sorda	seem	s
ʃ	S	fricativa postalveolar sorda	ship	S
t	t	oclusiva alveolar sorda	task	t
t͡ʃ	tS	africada postalveolar sorda	chart	S
Θ	T	fricativa dental sorda	thin	T
v	v	fricativa labiodental sonora	vest	f
w	w	aproximante velo-labial	west	u

IPA	X-SAMPA	Descripción	Ejemplo	Visema
z	z	fricativa alveolar sonora	zero	s
ʒ	Z	fricativa postalveolar sonora	vision	S
Vocales				
ə	@	vocal media central	arena	@
əʊ	@U	diptongo	goat	@
æ	{	vocal casi abierta anterior no redondeada	trap	a
aɪ	aI	diptongo	price	a
aʊ	aU	diptongo	mouth	a
ɑː	A:	vocal larga abierta posterior no redondeada	father	a
eɪ	eI	diptongo	face	e
ɜː	3:	vocal larga abierta media central no redondeada	nurse	E
ɛ	E	vocal abierta semianterior no redondeada	dress	E
ɛə	E@	diptongo	square	E

IPA	X-SAMPA	Descripción	Ejemplo	Visema
iː	i	vocal larga cerrada anterior no redondeada	fleece	i
ɪ	I	vocal casi cerrada semianterior no redondeada	kit	i
ɪə	I@	diptongo	near	i
ɔː	OI	vocal larga semiabier ta posterior redondeada	thought	O
ɔɪ	OI	Diptongo	choice	O
ɒ	Q	vocal abierta posterior redondeada	lot	O
uː	u:	vocal larga cerrada posterior redondeada	goose	u
ʊ	U	vocal casi cerrada semiposterior redondeada	foot	u
ʊə	U@	diptongo	cure	u
ʌ	V	Open-mid-back vocal no redondeada	strut	E
Símbolos adicionales				

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ˈ	"	acento principal	Alabama	
ˌ	%	acento secundario	Alabama	
.	.	límite de sílaba	A.la.ba.ma	

Inglés (Reino Unido) (en-GB)

En la tabla siguiente se muestra el conjunto de fonemas del Alfabeto Fonético Internacional (IPA, por sus siglas en inglés), los símbolos del Alfabeto Fonético Extendido SAM (X-SAMPA, por sus siglas en inglés) y los correspondientes visemas para las voces del idioma inglés británico admitidas por Amazon Polly.

Tabla de fonemas/visemas

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Consonantes				
b	b	oclusiva bilabial sonora	bed	p
d	d	oclusiva alveolar sonora	dig	t
ʤ	dZ	africada postalveolar sonora	jump	S
ð	D	fricativa dental sonora	then	T
f	f	fricativa labiodental sorda	five	f
g	g	oclusiva velar sonora	game	k

IPA	X-SAMPA	Descripción	Ejemplo	Visema
h	h	fricativa glotal sorda	house	k
j	j	aproximante palatal	yes	i
k	k	oclusiva velar sorda	cat	k
l	l	aproximante alveolar lateral	lay	t
l̩	l=	aproximante alveolar lateral silábica	battle	t
m	m	nasal bilabial	mouse	p
m̩	m=	nasal bilabial silábica	anthem	p
n	n	nasal alveolar	nap	t
n̩	n=	nasal alveolar silábica	botón	t
ŋ	N	nasal velar	thing	k
p	p	oclusiva bilabial sorda	pin	p
ɹ	r\	aproximante alveolar	red	r
s	s	fricativa alveolar sorda	seem	s

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ʃ	S	fricativa postalveolar sorda	ship	S
t	t	oclusiva alveolar sorda	task	t
‿tʃ	tS	africada postalveolar sorda	chart	S
Θ	T	fricativa dental sorda	thin	T
v	v	fricativa labiodental sonora	vest	f
w	w	aproximante velo-labial	west	u
z	z	fricativa alveolar sonora	zero	s
ʒ	Z	fricativa postalveolar sonora	vision	S
Vocales				
ə	@	vocal media central	arena	@
əʊ	@U	diptongo	goat	@
æ	{	vocal casi abierta anterior no redondeada	trap	a
aɪ	aI	diptongo	price	a
aʊ	aU	diptongo	mouth	a

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ɑː	A:	vocal larga abierta posterior no redondeada	father	a
eɪ	eI	diptongo	face	e
ɜː	3:	vocal larga abierta media central no redondeada	nurse	E
ɛ	E	vocal abierta semianterior no redondeada	dress	E
ɛə	E@	diptongo	square	E
iː	i	vocal larga cerrada anterior no redondeada	fleece	i
ɪ	I	vocal casi cerrada semianterior no redondeada	kit	i
ɪə	I@	diptongo	near	i
ɔː	O:	vocal larga semiabier ta posterior redondeada	thought	O
ɔɪ	OI	Diptongo	choice	O
ɒ	Q	vocal abierta posterior redondeada	lot	O

IPA	X-SAMPA	Descripción	Ejemplo	Visema
u:	u:	vocal larga cerrada posterior redondeada	goose	u
ʊ	U	vocal casi cerrada semiposterior redondeada	foot	u
ʊə	U@	diptongo	cure	u
ʌ	V	Open-mid-back vocal no redondeada	strut	E
Símbolos adicionales				
ˈ	"	acento principal	Alabama	
ˌ	%	acento secundario	Alabama	
.	.	límite de sílaba	A.la.ba.ma	

Inglés (India) (en-IN)

En la tabla siguiente se muestra el conjunto de fonemas del Alfabeto Fonético Internacional (IPA, por sus siglas en inglés), los símbolos del Alfabeto Fonético Extendido SAM (X-SAMPA, por sus siglas en inglés) y los correspondientes visemas para las voces del idioma inglés de India admitidas por Amazon Polly.

Para determinar los fonemas que se usan conjuntamente con el inglés de India, consulte Hindi (hi-IN).

Tabla de fonemas/visemas

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Consonantes				

IPA	X-SAMPA	Descripción	Ejemplo	Visema
b	b	oclusiva bilabial sonora	bed	p
d	d	oclusiva alveolar sonora	dig	t
͡dʒ	dZ	africada postalveolar sonora	jump	S
ð	D	fricativa dental sonora	then	T
f	f	fricativa labiodental sorda	five	f
g	g	oclusiva velar sonora	game	k
h	h	fricativa glotal sorda	house	k
j	j	aproximante palatal	yes	i
k	k	oclusiva velar sorda	cat	k
l	l	aproximante alveolar lateral	lay	t
l̩	l=	aproximante alveolar lateral silábica	battle	t
m	m	nasal bilabial	mouse	p
m̩	m=	nasal bilabial silábica	anthem	p

IPA	X-SAMPA	Descripción	Ejemplo	Visema
n	n	nasal alveolar	nap	t
n̩	n=	nasal alveolar silábica	nap	t
ŋ	N	nasal velar	thing	k
p	p	oclusiva bilabial sorda	pin	p
ɹ	r\	aproximante alveolar	red	r
s	s	fricativa alveolar sorda	seem	s
ʃ	S	fricativa postalveolar sorda	ship	S
t	t	oclusiva alveolar sorda	task	t
t͡ʃ	tS	africada postalveolar sorda	chart	S
Θ	T	fricativa dental sorda	thin	T
v	v	fricativa labiodental sonora	vest	f
w	w	aproximante velo-labial	west	u
z	z	fricativa alveolar sonora	zero	s

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ʒ	Z	fricativa postalveolar sonora	vision	S
Vocales				
ə	@	vocal media central	arena	@
əʊ	@U	diptongo	goat	@
æ	{	vocal casi abierta anterior no redondeada	trap	a
aɪ	aI	diptongo	price	a
aʊ	aU	diptongo	mouth	a
ɑː	A:	vocal larga abierta posterior no redondeada	father	a
eɪ	eI	diptongo	face	e
ɜː	3:	vocal larga abierta media central no redondeada	nurse	E
ɛ	E	vocal abierta semianterior no redondeada	dress	E
ɛə	E@	diptongo	square	E
iː	i	vocal larga cerrada anterior no redondeada	fleece	i

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ɪ	I	vocal casi cerrada semianterior no redondeada	kit	i
ɪə	I@	diptongo	near	i
ɔː	OI	vocal larga semiabier ta posterior redondeada	thought	O
ɔɪ	OI	Diptongo	choice	O
ɒ	Q	vocal abierta posterior redondeada	lot	O
uː	u:	vocal larga cerrada posterior redondeada	goose	u
ʊ	U	vocal casi cerrada semiposterior redondeada	foot	u
ʊə	U@	diptongo	cure	u
ʌ	V	Open-mid-back vocal no redondeada	strut	E
Símbolos adicionales				
ˈ	"	acento principal	Alabama	
ˌ	%	acento secundario	Alabama	
.	.	límite de sílaba	A.la.ba.ma	

Inglés (Irlanda) (en-IE)

En la tabla siguiente se muestra el conjunto de fonemas del Alfabeto Fonético Internacional (IPA, por sus siglas en inglés), los símbolos del Alfabeto Fonético Extendido SAM (X-SAMPA, por sus siglas en inglés) y los correspondientes visemas para las voces del idioma inglés americano admitidas por Amazon Polly.

Tabla de fonemas/visemas

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Consonantes				
b	b	oclusiva bilabial sonora	bed	p
d	d	oclusiva alveolar sonora	dig	t
͡dʒ	dZ	africada postalveolar sonora	jump	S
ð	D	fricativa dental sonora	then	T
f	f	fricativa labiodental sorda	five	f
g	g	oclusiva velar sonora	game	k
h	h	fricativa glotal sorda	house	k
j	j	aproximante palatal	yes	i
k	k	oclusiva velar sorda	cat	k

IPA	X-SAMPA	Descripción	Ejemplo	Visema
l	l	aproximante alveolar lateral	lay	t
m	m	nasal bilabial	mouse	p
n	n	nasal alveolar	nap	t
ŋ	N	nasal velar	thing	k
p	p	oclusiva bilabial sorda	speak	p
ɹ	r\	aproximante alveolar	red	r
s	s	fricativa alveolar sorda	seem	s
ʃ	S	fricativa postalveolar sorda	ship	S
t	t	oclusiva alveolar sorda	trap	t
t͡ʃ	tS	africada postalveolar sorda	chart	S
θ	T	fricativa dental sorda	thin	T
v	v	fricativa labiodental sonora	vest	f
w	w	aproximante velo-labial	west	u
z	z	fricativa alveolar sonora	zero	s

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ʒ	Z	fricativa postalveolar sonora	vision	S
Vocales				
ə	@	vocal media central	arena	@
ɚ	@`	vocal media central rótica	reader	@
æ	{	vocal casi abierta anterior no redondeada	trap	a
aɪ	aI	diptongo	price	a
aʊ	aU	diptongo	mouth	a
ɑ	A	vocal larga abierta posterior no redondeada	father	a
eɪ	eI	diptongo	face	e
ɝ	3`	vocal media central rótica redondeada abierta	nurse	E
ɛ	E	vocal abierta semianterior no redondeada	dress	E
i	i	vocal larga cerrada anterior no redondeada	fleece	i

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ɪ	I	vocal casi cerrada semianterior no redondeada	kit	i
oʊ	oU	diptongo	goat	o
ɔ	O	vocal abierta posterior redondeada larga	thought	O
ɔɪ	OI	diptongo	choice	O
u	u	vocal larga cerrada posterior redondeada	goose	u
ʊ	U	vocal casi cerrada semiposterior redondeada	foot	u
ʌ	V	open-mid-back vocal no redondeada	strut	E
Símbolos adicionales				
ˈ	"	acento principal	Alabama	
ˌ	%	acento secundario	Alabama	
.	.	límite de sílaba	A.la.ba.ma	

Inglés (Nueva Zelanda) (en-NZ)

En la tabla siguiente se muestra el conjunto de fonemas del Alfabeto Fonético Internacional (IPA, por sus siglas en inglés), los símbolos del Alfabeto Fonético Extendido SAM (X-SAMPA, por sus siglas

en inglés) y los correspondientes visemas para las voces del idioma inglés americano admitidas por Amazon Polly.

Tabla de fonemas/visemas

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Consonantes				
b	b	oclusiva bilabial sonora	bed	p
d	d	oclusiva alveolar sonora	dig	t
ʤ	dZ	africada postalveolar sonora	jump	S
ð	D	fricativa dental sonora	then	T
f	f	fricativa labiodental sorda	five	f
g	g	oclusiva velar sonora	game	k
h	h	fricativa glotal sorda	house	k
j	j	aproximante palatal	yes	i
k	k	oclusiva velar sorda	cat	k
l	l	aproximante alveolar lateral	lay	t

IPA	X-SAMPA	Descripción	Ejemplo	Visema
l̩	l=	aproximante alveolar lateral silábica	battle	t
m	m	nasal bilabial	mouse	p
m̩	m=	nasal bilabial silábica	anthem	p
n	n	nasal alveolar	nap	t
n̩	n=	nasal alveolar silábica	botón	t
ŋ	N	nasal velar	thing	k
p	p	oclusiva bilabial sorda	pin	p
ɹ	r\	aproximante alveolar	red	r
s	s	fricativa alveolar sorda	seem	s
ʃ	S	fricativa postalveolar sorda	ship	S
t	t	oclusiva alveolar sorda	task	t
t͡ʃ	tS	africada postalveolar sorda	chart	S
Θ	T	fricativa dental sorda	thin	T

IPA	X-SAMPA	Descripción	Ejemplo	Visema
v	v	fricativa labiodental sonora	vest	f
w	w	aproximante velo-labial	west	u
z	z	fricativa alveolar sonora	zero	s
ʒ	Z	fricativa postalveolar sonora	vision	S
Vocales				
ə	@	vocal media central	arena	@
əʊ	@U	diptongo	goat	@
æ	{	vocal casi abierta anterior no redondeada	trap	a
aɪ	aI	diptongo	price	a
aʊ	aU	diptongo	mouth	a
ɑː	A:	vocal larga abierta posterior no redondeada	father	a
eɪ	eI	diptongo	face	e
ɜː	3:	vocal larga abierta media central no redondeada	nurse	E

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ɛ	E	vocal abierta semianterior no redondeada	dress	E
ɛə	E@	diptongo	square	E
iː	i	vocal larga cerrada anterior no redondeada	fleece	i
ɪ	I	vocal casi cerrada semianterior no redondeada	kit	i
ɪə	I@	diptongo	near	i
ɔː	O:	vocal larga semiabier ta posterior redondeada	thought	O
ɔɪ	OI	Diptongo	choice	O
ɒ	Q	vocal abierta posterior redondeada	lot	O
uː	u:	vocal larga cerrada posterior redondeada	goose	u
ʊ	U	vocal casi cerrada semiposterior redondeada	foot	u
ʊə	U@	diptongo	cure	u

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ʌ	V	Open-mid-back vocal no redondeada	strut	E
Símbolos adicionales				
ˈ	"	acento principal	Alabama	
ˌ	%	acento secundario	Alabama	
.	.	límite de sílaba	A.la.ba.ma	

La voz de Aria habla inglés neozelandés y ofrece un apoyo limitado al maorí. Puede pronunciar las siguientes palabras y frases maoríes. Las frases maoríes distinguen entre mayúsculas y minúsculas.

Inglés	Maorí
Hola/Saludos	Kia ora
Le damos la bienvenida (to)	Nau mai (ki)
Hola (una persona)/gracias	Tēnā koe
Hola (tres o más personas)/gracias	Tēnā koutou
Buenos días	Ata mārie
Buenos días	Mōrena
Gracias	Ngā mihi
Cuídate	Ngā manaakitanga
Nos vemos	Ka kite
Nos vemos más tarde	Mā te wā
Que tengas un buen día	Kia pai tō rā

Inglés	Maorí
Feliz Navidad	Meri Kirihimete
Maorí	Māori
Lengua maorí	te reo Māori
semana de la lengua maorí	Te wiki o te reo Māori
Nueva Zelanda	Aotearoa
Año nuevo maorí	Mātariki
Ciudad de Nueva Zelanda/El Día de Waitangi es el día nacional de Nueva Zelanda	Waitangi
One	tahi
Dos	rua
Tres	toru
Cuatro	whā
Cinco	rima
Seis	ono
Siete	whitu
Ocho	waru
Nueve	iwa
Diez	tekau
Veinte	rua tekau
Treinta	Toru tekau

inglés (singapurense) (en-SG)

En la siguiente tabla se enumeran los fonemas del Alfabeto Fonético Internacional (IPA), los símbolos del Alfabeto Fonético de los Métodos de Evaluación de la Voz Extendida (X-SAMPA) y los visemas correspondientes a las voces en inglés de Singapur compatibles con Amazon Polly.

Tabla de fonemas/visemas

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Consonantes				
b	b	oclusiva bilabial sonora	bed	p
d	d	oclusiva alveolar sonora	dig	t
ʤ	dZ	africada postalveolar sonora	jump	S
ð	D	fricativa dental sonora	then	T
f	f	fricativa labiodental sorda	five	f
g	g	oclusiva velar sonora	game	k
h	h	fricativa glotal sorda	house	k
j	j	aproximante palatal	yes	i
k	k	oclusiva velar sorda	cat	k

IPA	X-SAMPA	Descripción	Ejemplo	Visema
l	l	aproximante alveolar lateral	lay	t
l̩	l=	aproximante alveolar lateral silábica	battle	t
m	m	nasal bilabial	mouse	p
m̩	m=	nasal bilabial silábica	anthem	p
n	n	nasal alveolar	nap	t
n̩	n=	nasal alveolar silábica	botón	t
ŋ	N	nasal velar	thing	k
p	p	oclusiva bilabial sorda	pin	p
ɹ	r\	aproximante alveolar	red	r
s	s	fricativa alveolar sorda	seem	s
ʃ	S	fricativa postalveolar sorda	ship	S
t	t	oclusiva alveolar sorda	task	t
t͡ʃ	tS	africada postalveolar sorda	chart	S

IPA	X-SAMPA	Descripción	Ejemplo	Visema
θ	T	fricativa dental sorda	thin	T
v	v	fricativa labiodental sonora	vest	f
w	w	aproximante velo-labial	west	u
z	z	fricativa alveolar sonora	zero	s
ʒ	Z	fricativa postalveolar sonora	vision	S
Vocales				
ə	@	vocal media central	arena	@
əʊ	@U	diptongo	goat	@
æ	{	vocal casi abierta anterior no redondeada	trap	a
aɪ	aI	diptongo	price	a
aʊ	aU	diptongo	mouth	a
ɑː	A:	vocal larga abierta posterior no redondeada	father	a
eɪ	eI	diptongo	face	e
ɜː	3:	vocal larga abierta media central no redondeada	nurse	E

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ɛ	E	vocal abierta semianterior no redondeada	dress	E
ɛə	E@	diptongo	square	E
iː	i	vocal larga cerrada anterior no redondeada	fleece	i
ɪ	I	vocal casi cerrada semianterior no redondeada	kit	i
ɪə	I@	diptongo	near	i
ɔː	O:	vocal larga semiabier ta posterior redondeada	thought	O
ɔɪ	OI	Diptongo	choice	O
ɒ	Q	vocal abierta posterior redondeada	lot	O
uː	u:	vocal larga cerrada posterior redondeada	goose	u
ʊ	U	vocal casi cerrada semiposterior redondeada	foot	u
ʊə	U@	diptongo	cure	u

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ʌ	V	Open-mid-back vocal no redondeada	strut	E
Símbolos adicionales				
ˈ	"	acento principal	Alabama	
ˌ	%	acento secundario	Alabama	
.	.	límite de sílaba	A.la.ba.ma	

Inglés (sudafricano) (en-ZA)

En la tabla siguiente se muestra el conjunto de fonemas del Alfabeto Fonético Internacional (IPA, por sus siglas en inglés), los símbolos del Alfabeto Fonético Extendido SAM (X-SAMPA, por sus siglas en inglés) y los correspondientes visemas para las voces del idioma inglés de Australia admitidas por Amazon Polly.

Tabla de fonemas/visemas

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Consonantes				
b	b	oclusiva bilabial sonora	bed	p
d	d	oclusiva alveolar sonora	dig	t
ʤ	dZ	africada postalveolar sonora	jump	S
ð	D	fricativa dental sonora	then	T

IPA	X-SAMPA	Descripción	Ejemplo	Visema
f	f	fricativa labiodental sorda	five	f
g	g	oclusiva velar sonora	game	k
h	h	fricativa glotal sorda	house	k
j	j	aproximante palatal	yes	i
k	k	oclusiva velar sorda	cat	k
l	l	aproximante alveolar lateral	lay	t
l̩	l=	aproximante alveolar lateral silábica	battle	t
ɬ̩	K	fricativa lateral sorda	umhlanga	t
m	m	nasal bilabial	mouse	p
m̩	m=	nasal bilabial silábica	anthem	p
n	n	nasal alveolar	nap	t
n̩	n=	nasal alveolar silábica	botón	t
ŋ	N	nasal velar	thing	k

IPA	X-SAMPA	Descripción	Ejemplo	Visema
p	p	oclusiva bilabial sorda	pin	p
ɹ	r\	aproximante alveolar	red	r
r	r	vibrante alveolar	pareis	r
s	s	fricativa alveolar sorda	seem	s
ʃ	S	fricativa postalveolar sorda	ship	S
t	t	oclusiva alveolar sorda	task	t
͡tʃ	tS	africada postalveolar sorda	chart	S
Θ	T	fricativa dental sorda	thin	T
v	v	fricativa labiodental sonora	vest	f
w	w	aproximante velo-labial	west	u
x	x	fricativa velar sorda	gauteng	k
z	z	fricativa alveolar sonora	zero	s
!	!\	clic postalveolar	gqeberha	k
|	|\	clic dental	ncube	t

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ǁ	||\	clic lateral	xhosa	t
Vocales				
ə	@	vocal media central	arena	@
əi	@i	diptongo	nelspruit	i
əʊ	@U	diptongo	goat	@
æ	{	vocal casi abierta anterior no redondeada	trap	a
aɪ	aI	diptongo	price	a
aʊ	aU	diptongo	mouth	a
ɑː	A:	vocal larga abierta posterior no redondeada	father	a
eɪ	eI	diptongo	face	e
ɜː	3:	vocal larga abierta media central no redondeada	nurse	E
ɛ	E	vocal abierta semianterior no redondeada	dress	E
ɛə	E@	diptongo	square	E
iː	i	vocal larga cerrada anterior no redondeada	fleece	i

IPA	X-SAMPA	Descripción	Ejemplo	Visema
iə	I@	diptongo	du preez	i
ɪ	I	vocal casi cerrada semianterior no redondeada	kit	i
ɪə	I@	diptongo	near	i
ɔː	O:	vocal larga semiabier ta posterior redondeada	thought	O
ɔɪ	OI	Diptongo	choice	O
ɒ	Q	vocal abierta posterior redondeada	lot	O
uː	u:	vocal larga cerrada posterior redondeada	goose	u
ʊ	U	vocal casi cerrada semiposterior redondeada	foot	u
ʊə	U@	diptongo	cure	u
ʌ	V	Open-mid-back vocal no redondeada	strut	E
y	y	vocal cerrada anterior redondead a	van vuuren	u

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Símbolos adicionales				
ˈ	"	acento principal	Alabama	
ˌ	%	acento secundario	Alabama	
.	.	límite de sílaba	A.la.ba.ma	

Inglés (Gales) (en-GB-WLS)

En la tabla siguiente se muestra el conjunto de fonemas del Alfabeto Fonético Internacional (IPA, por sus siglas en inglés), los símbolos del Alfabeto Fonético Extendido SAM (X-SAMPA, por sus siglas en inglés) y los correspondientes visemas para las voces del idioma inglés de Gales admitidas por Amazon Polly.

Tabla de fonemas/visemas

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Consonantes				
b	b	oclusiva bilabial sonora	bed	p
d	d	oclusiva alveolar sonora	dig	t
ʤ	dZ	africada postalveolar sonora	jump	S
ð	D	fricativa dental sonora	then	T
f	f	fricativa labiodental sorda	five	f

IPA	X-SAMPA	Descripción	Ejemplo	Visema
g	g	oclusiva velar sonora	game	k
h	h	fricativa glotal sorda	house	k
j	j	aproximante palatal	yes	i
k	k	oclusiva velar sorda	cat	k
l	l	aproximante alveolar lateral	lay	t
l̩	l=	aproximante alveolar lateral silábica	battle	t
m	m	nasal bilabial	mouse	p
m̩	m=	nasal bilabial silábica	anthem	p
n	n	nasal alveolar	nap	t
n̩	n=	nasal alveolar silábica	nap	t
ŋ	N	nasal velar	thing	k
p	p	oclusiva bilabial sorda	pin	p
ɹ	r\	aproximante alveolar	red	r

IPA	X-SAMPA	Descripción	Ejemplo	Visema
s	s	fricativa alveolar sorda	seem	s
ʃ	S	fricativa postalveolar sorda	ship	S
t	t	oclusiva alveolar sorda	task	t
͡tʃ	tS	africada postalveolar sorda	chart	S
Θ	T	fricativa dental sorda	thin	T
v	v	fricativa labiodental sonora	vest	f
w	w	aproximante velo-labial	west	u
z	z	fricativa alveolar sonora	zero	s
ʒ	Z	fricativa postalveolar sonora	vision	S
Vocales				
ə	@	vocal media central	arena	@
əʊ	@U	diptongo	goat	@
æ	{	vocal casi abierta anterior no redondeada	trap	a
aɪ	aI	diptongo	price	a

IPA	X-SAMPA	Descripción	Ejemplo	Visema
aʊ	aU	diptongo	mouth	a
ɑː	A:	vocal larga abierta posterior no redondeada	father	a
eɪ	eI	diptongo	face	e
ɜː	3:	vocal larga abierta media central no redondeada	nurse	E
ɛ	E	vocal abierta semianterior no redondeada	dress	E
ɛə	E@	diptongo	square	E
iː	i	vocal larga cerrada anterior no redondeada	fleece	i
ɪ	I	vocal casi cerrada semianterior no redondeada	kit	i
ɪə	I@	diptongo	near	i
ɔː	OI	vocal larga semiabier ta posterior redondeada	thought	O
ɔɪ	OI	Diptongo	choice	O

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ɒ	Q	vocal abierta posterior redondeada	lot	O
uː	u:	vocal larga cerrada posterior redondeada	goose	u
ʊ	U	vocal casi cerrada semiposterior redondeada	foot	u
ʊə	U@	diptongo	cure	u
ʌ	V	Open-mid-back vocal no redondeada	strut	E
Símbolos adicionales				
ˈ	"	acento principal	Alabama	
ˌ	%	acento secundario	Alabama	
.	.	límite de sílaba	A.la.ba.ma	

Finés (fi-FI)

En la tabla siguiente se muestra el conjunto de fonemas del Alfabeto Fonético Internacional (IPA, por sus siglas en inglés), los símbolos del Alfabeto Fonético Extendido SAM (X-SAMPA, por sus siglas en inglés) y los correspondientes visemas para las voces del idioma polaco admitidas por Amazon Polly.

Tabla de fonemas/visemas

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Consonantes finlandesas				
p	p	oclusiva bilabial sorda	[p]ankki	p
t	t	oclusiva alveolar sorda	[t]alo	t
k	k	oclusiva velar sorda	[k]aali	k
d	d	oclusiva alveolar sonora	[d]ata	t
s	s	fricativa alveolar sorda	[s]ali	s
h	h	fricativa glotal sorda	[h]attu	k
ʋ	v\	aproximante labiodental sonora	[v]aiva´	v
j	j	aproximante palatal	[j]oki	i
l	l	aproximante alveolar lateral	[l]oma	t
r	r	vibrante alveolar sonora	[r]iita	r
m	m	nasal bilabial	[m]ato	p
n	n	nasal alveolar	[n]enää	t
ŋ	N	nasal velar	he[n]ki	k

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Las consonantes se encuentran en los préstamos				
b	b	oclusiva bilabial sonora	[b]ussi	p
f	f	fricativa labiodental sorda	[f]irma	v
w	w	aproximante velo-labial	[w]iki	u
z	z	fricativa alveolar sonora	[z]ulu	s
g	g	oclusiva velar sonora	[g]aala	k
ʃ	S	fricativa postalveolar sorda	[sh]akki	S
ʒ	Z	fricativa postalveolar sonora	[g]enre	S
θ	T	fricativa dental sorda	ear[th]	T
ð	D	fricativa dental sonora	ei[th]er	T
Vocales cortas				
i	i	vocal cerrada anterior no redondeada	k[i]lo	i
ɛ	E	vocal abierta semianterior no redondeada	k[e]sä	E

IPA	X-SAMPA	Descripción	Ejemplo	Visema
æ	{	vocal casi abierta anterior no redondeada	k[ä]ly	A
y	y	vocal cerrada anterior redondeada	k[y]lä	u
ø	2	vocal cerrada semianterior redondeada	p[ö]ly	O
u	u	vocal semicerrada posterior redondeada	k[u]lo	u
ɔ	O	vocal semiabierta posterior redondeada	k[o]lo	O
ɑ	A	vocal abierta posterior no redondeada	k[a]la	A
Vocales largas				
iː	iː	vocal larga cerrada anterior no redondeada	s[ii]li	i
ɛː	E:	vocal larga semiabierta anterior no redondeada	[ee]tu	E

IPA	X-SAMPA	Descripción	Ejemplo	Visema
æː	{:	vocal larga casi abierta anterior no redondeada	t[ää]llä	A
yː	y:	vocal larga cerrada anterior no redondeada	t[yy]li	u
øː	2:	vocal larga semicerrada anterior redondeada	t[öö]lö	O
uː	u:	vocal larga cerrada posterior redondeada	t[uu]li	u
ɔː	O:	vocal abierta posterior redondeada larga	r[oo]li	O
ɑː	A:	vocal larga abierta posterior no redondeada	k[aa]su	A
Diptongos				
ɛi	Ei	diptongo	l[ei]pä	E
æi	{i	diptongo	[äi]ti	A
ui	ui	diptongo	k[ui]n	u
ɑi	Ai	diptongo	k[ai]kki	A
ɔi	Oi	diptongo	p[oi]ka	O
øi	2i	diptongo	s[öi]n	O

IPA	X-SAMPA	Descripción	Ejemplo	Visema
yi	yi	diptongo	l[yi]jy	u
ɑu	Au	diptongo	s[au]na	A
ɔu	Ou	diptongo	k[ou]lu	O
ɛu	Eu	diptongo	r[eu]na	E
iu	iu	diptongo	v[iu]lu	i
æy	{y	diptongo	t[äy]nnä	A
øy	2y	diptongo	k[öy]hä	O
ɛy	Ey	diptongo	pes[ey]tyä	E
iy	iy	diptongo	käär[iy]tyä	i
iɛ	iE	diptongo	t[ie]	i
yø	y2	diptongo	[yö]	u
uɔ	uO	diptongo	t[uo]	u
Vocales que se encuentran en préstamos del inglés				
ɪ	I	vocal casi cerrada semianterior no redondeada	[bit]	i
ʊ	U	vocal casi cerrada semiposterior redondeada	[libro]	u
ə	@	vocal media central	[a]bout	@
ʌ	V	open-mid-back vocal no redondeada	c[u]t	E

Francés (fr-FR)

En la tabla siguiente se muestra el conjunto de fonemas del Alfabeto Fonético Internacional (IPA, por sus siglas en inglés), los símbolos del Alfabeto Fonético Extendido SAM (X-SAMPA, por sus siglas en inglés) y los correspondientes visemas para las voces del idioma francés admitidas por Amazon Polly.

Tabla de fonemas/visemas

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Consonantes				
b	b	oclusiva bilabial sonora	boire	p
d	d	oclusiva alveolar sonora	madame	t
f	f	fricativa labiodental sorda	femme	f
g	g	oclusiva velar sonora	grand	k
ɥ	H	aproximante labial palatal	bruit	u
j	j	aproximante palatal	meilleur	i
k	k	oclusiva velar sorda	quatre	k
l	l	aproximante alveolar lateral	malade	t
m	m	nasal bilabial	maison	p
n	n	nasal alveolar	astronome	t

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ɲ	J	nasal palatal	baigner	J
ŋ	N	nasal velar	parking	k
p	p	oclusiva bilabial sorda	pomme	p
ʁ	R	fricativa uvular sonora	amoureux	k
s	s	fricativa alveolar sorda	santé	s
ʃ	S	fricativa postalveolar sorda	chat	S
t	t	oclusiva alveolar sorda	téléphone	t
v	v	fricativa labiodental sonora	vrai	f
w	w	aproximante velo-labial	soir	u
z	z	fricativa alveolar sonora	raison	s
ʒ	Z	fricativa postalveolar sonora	aubergine	S
Vocales				
ø	2	vocal semicerrada anterior redondeada	deux	o

IPA	X-SAMPA	Descripción	Ejemplo	Visema
œ	9	vocal semiabierta anterior redondeada	neuf	O
œ̃	9~	vocal nasal semiabierta anterior redondeada	brun	O
ə	@	vocal media central	je	@
a	a	vocal abierta anterior no redondeada	table	a
ɑ̃	A~	vocal nasal posterior abierta no redondeada	camembert	a
e	e	vocal semicerrada anterior no redondeada	marché	e
ɛ	E	vocal semiabierta anterior no redondeada	neige	E
ɛ̃	E~	vocal nasal semiabierta anterior no redondeada	sapin	E
i	i	vocal cerrada anterior no redondeada	mille	i

IPA	X-SAMPA	Descripción	Ejemplo	Visema
o	o	vocal semicerrada posterior redondeada	hôpital	o
ɔ	O	vocal semiabierta posterior redondeada	homme	O
ɔ̃	O~	vocal nasal semiabierta posterior redondeada	bon	O
u	u	vocal semicerrada posterior redondeada	sous	u
y	y	vocal cerrada anterior redondeada	dur	u
Símbolos adicionales				
ˈ	"	acento principal	Alabama	
ˌ	%	acento secundario	Alabama	
.	.	límite de sílaba	A.la.ba.ma	

Francés (belga) (fr-BE)

En la tabla siguiente se muestra el conjunto de fonemas del Alfabeto Fonético Internacional (IPA, por sus siglas en inglés), los símbolos del Alfabeto Fonético Extendido SAM (X-SAMPA, por sus siglas en inglés) y los correspondientes visemas para las voces del idioma francés belga admitidas por Amazon Polly.

Tabla de fonemas/visemas

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Consonantes				
b	b	oclusiva bilabial sonora	boire	p
d	d	oclusiva alveolar sonora	madame	t
f	f	fricativa labiodental sorda	femme	f
g	g	oclusiva velar sonora	grand	k
ɥ	H	aproximante labial palatal	bruit	u
j	j	aproximante palatal	meilleur	i
k	k	oclusiva velar sorda	quatre	k
l	l	aproximante alveolar lateral	malade	t
m	m	nasal bilabial	maison	p
n	n	nasal alveolar	astronome	t
ɲ	J	nasal palatal	baigner	J
ŋ	N	nasal velar	parking	k
p	p	oclusiva bilabial sorda	pomme	p

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ʁ	R	fricativa uvular sonora	amoureux	k
s	s	fricativa alveolar sorda	santé	s
ʃ	S	fricativa postalveolar sorda	chat	S
t	t	oclusiva alveolar sorda	téléphone	t
v	v	fricativa labiodental sonora	vrai	f
w	w	aproximante velo-labial	soir	u
z	z	fricativa alveolar sonora	raison	s
ʒ	Z	fricativa postalveolar sonora	aubergine	S
Vocales				
ø	2	vocal semicerrada anterior redondeada	deux	o
œ	9	vocal semiabierta anterior redondeada	neuf	O

IPA	X-SAMPA	Descripción	Ejemplo	Visema
œ̃	9~	vocal nasal semiabierta anterior redondeada	brun	O
ə	@	vocal media central	je	@
a	a	vocal abierta anterior no redondeada	table	a
ɑ̃	A~	vocal nasal posterior abierta no redondeada	camembert	a
e	e	vocal semicerrada anterior no redondeada	marché	e
ɛ	E	vocal semiabierta anterior no redondeada	neige	E
ɛ̃	E~	vocal nasal semiabierta anterior no redondeada	sapin	E
i	i	vocal cerrada anterior no redondeada	mille	i
o	o	vocal semicerrada posterior redondeada	hôpital	o

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ɔ	O	vocal semiabierta posterior redondeada	homme	O
ɔ̃	O~	vocal nasal semiabierta posterior redondeada	bon	O
u	u	vocal semicerrada posterior redondeada	sous	u
y	y	vocal cerrada anterior redondeada	dur	u
Símbolos adicionales				
ˈ	"	acento principal	Alabama	
ˌ	%	acento secundario	Alabama	
.	.	límite de sílaba	A.la.ba.ma	

Francés (Canadá) (fr-CA)

En la tabla siguiente se muestra el conjunto de fonemas del Alfabeto Fonético Internacional (IPA, por sus siglas en inglés), los símbolos del Alfabeto Fonético Extendido SAM (X-SAMPA, por sus siglas en inglés) y los correspondientes visemas para las voces del idioma francés de Canadá por Amazon Polly.

Tabla de fonemas/visemas

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Consonantes				

IPA	X-SAMPA	Descripción	Ejemplo	Visema
b	b	oclusiva bilabial sonora	boire	p
d	d	oclusiva alveolar sonora	madame	t
f	f	fricativa labiodental sorda	femme	f
g	g	oclusiva velar sonora	grand	k
ɥ	H	aproximante labial palatal	bruit	u
j	j	aproximante palatal	meilleur	i
k	k	oclusiva velar sorda	quatre	k
l	l	aproximante alveolar lateral	malade	t
m	m	nasal bilabial	maison	p
n	n	nasal alveolar	astronome	t
ɲ	J	nasal palatal	baigner	J
ŋ	N	nasal velar	parking	k
p	p	oclusiva bilabial sorda	pomme	p
ʁ	R	fricativa uvular sonora	amoureux	k

IPA	X-SAMPA	Descripción	Ejemplo	Visema
s	s	fricativa alveolar sorda	santé	s
ʃ	S	fricativa postalveolar sorda	chat	S
t	t	oclusiva alveolar sorda	téléphone	t
v	v	fricativa labiodental sonora	vrai	f
w	w	aproximante velo-labial	soir	u
z	z	fricativa alveolar sonora	raison	s
ʒ	Z	fricativa postalveolar sonora	aubergine	S
Vocales				
ø	2	vocal semicerrada anterior redondeada	deux	o
œ	9	vocal semiabierta anterior redondeada	neuf	O
œ̃	9~	vocal nasal semiabierta anterior redondeada	brun	O
ə	@	vocal media central	je	@

IPA	X-SAMPA	Descripción	Ejemplo	Visema
a	a	vocal abierta anterior no redondeada	table	a
ɑ̃	A~	vocal nasal posterior abierta no redondeada	camembert	a
e	e	vocal semicerrada anterior no redondeada	marché	e
ɛ	E	vocal semiabierta anterior no redondeada	neige	E
ɛ̃	E~	vocal nasal semiabierta anterior no redondeada	sapin	E
i	i	vocal cerrada anterior no redondeada	mille	i
o	o	vocal semicerrada posterior redondeada	hôpital	o
ɔ	O	vocal semiabierta posterior redondeada	homme	O
ɔ̃	O~	vocal nasal semiabierta posterior redondeada	bon	O

IPA	X-SAMPA	Descripción	Ejemplo	Visema
u	u	vocal semicerrada posterior redondeada	sous	u
y	y	vocal cerrada anterior redondeada	dur	u
Símbolos adicionales				
ˈ	"	acento principal	Alabama	
ˌ	%	acento secundario	Alabama	
.	.	límite de sílaba	A.la.ba.ma	

Alemán (de-DE)

En la tabla siguiente se muestra el conjunto de fonemas del Alfabeto Fonético Internacional (IPA, por sus siglas en inglés), los símbolos del Alfabeto Fonético Extendido SAM (X-SAMPA, por sus siglas en inglés) y los correspondientes visemas para las voces del idioma alemán admitidas por Amazon Polly.

Tabla de fonemas/visemas

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Consonantes				
ʔ	?	oclusiva glotal		
b	b	oclusiva bilabial sonora	Bier	p
d	d	oclusiva alveolar sonora	Dach	t

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ç	C	fricativa palatal sorda	ich	k
͡dʒ	dZ	africada postalveolar sonora	Dschungel	S
f	f	Fricativa labiodental sorda	Vogel	f
g	g	Oclusiva velar sonora	Gabel	k
h	h	Fricativa glotal sorda	Haus	k
j	j	Fricativa glotal sorda	jemand	i
k	k	Oclusiva velar sorda	Kleid	k
l	l	Aproximante alveolar lateral	Loch	t
m	m	Nasal bilabial	Milch	p
n	n	Nasal alveolar	Natur	t
ŋ	N	Nasal velar	klingen	k
p	p	Oclusiva bilabial sorda	Park	p
͡pf	pf	Africada labiodental sorda	Apfel	
ʀ	R	Vibrante uvular	Regen	

IPA	X-SAMPA	Descripción	Ejemplo	Visema
s	s	fricativa alveolar sorda	Messer	s
ʃ	S	Fricativa postalveolar sorda	Fischer	S
t	t	Oclusiva alveolar sorda	Topf	T
͡ts	Ts	Africada alveolar sorda	Zahl	
͡tʃ	tS	Africada postalveolar sorda	deutsch	S
v	v	Fricativa labiodental sonora	Wasser	f
x	x	Fricativa velar sorda	kochen	k
z	z	Fricativa alveolar sonora	See	s
ʒ	Z	Fricativa postalveolar sonora	Orange	S
Vocales				
øː	2:	vocal larga semicerrada anterior redondeada	böse	o
ɐ	6	vocal casi abierta central	besser	a

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ɐ̯	6_^	vocal no silábica semiabierta central	Klar	a
œ	9	vocal semiabierta anterior redondeada	können	O
ə	@	vocal media central	Rede	@
a	a	vocal abierta anterior no redondeada	Salz	a
a:	a:	vocal larga abierta anterior no redondeada	Sahne	a
aɪ	aI	diptongo	nein	a
aʊ	aU	diptongo	Augen	a
ɑ̃	A~	vocal nasal posterior abierta no redondeada	Restaurant	a
e:	e:	vocal larga semicerrada anterior no redondeada	Rede	e
ɛ	E	vocal semiabierta anterior no redondeada	Keller	E

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ɛ̃	E~	vocal nasal semiabierta anterior no redondeada	Terrain	E
iː	i:	vocal larga cerrada anterior no redondeada	Lied	i
ɪ	I	vocal casi cerrada semianterior no redondeada	bitte	i
oː	o:	vocal larga semicerra da posterior redondeada	Kohl	o
ɔ	O	vocal semiabier ta posterior redondeada	Koffer	O
ɔ̃	O~	vocal nasal semiabier ta posterior redondeada	Annonce	O
ɔʏ	OY	diptongo	neu	O
uː	u:	vocal larga cerrada posterior redondeada	Bruder	u
ʊ	U	vocal casi cerrada semiposterior redondeada	Wunder	u

IPA	X-SAMPA	Descripción	Ejemplo	Visema
y:	y:	vocal larga cerrada anterior redondeada	kühl	u
ʏ	S	vocal casi cerrada semianterior redondeada	Küche	u
Símbolos adicionales				
ˈ	"	acento principal	Alabama	
ˌ	%	acento secundario	Alabama	
.	.	límite de sílaba	A.la.ba.ma	

Alemán (austriaco) (de-AT)

En la tabla siguiente se muestra el conjunto de fonemas del Alfabeto Fonético Internacional (IPA, por sus siglas en inglés), los símbolos del Alfabeto Fonético Extendido SAM (X-SAMPA, por sus siglas en inglés) y los correspondientes visemas para las voces del idioma inglés de Australia admitidas por Amazon Polly.

Tabla de fonemas/visemas

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Consonantes				
ʔ	?	oclusiva glotal		
b	b	oclusiva bilabial sonora	Bier	p
d	d	oclusiva alveolar sonora	Dach	t

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ç	C	fricativa palatal sorda	ich	k
͡dʒ	dZ	africada postalveolar sonora	Dschungel	S
f	f	Fricativa labiodental sorda	Vogel	f
g	g	Oclusiva velar sonora	Gabel	k
h	h	Fricativa glotal sorda	Haus	k
j	j	Fricativa glotal sorda	jemand	i
k	k	Oclusiva velar sorda	Kleid	k
l	l	Aproximante alveolar lateral	Loch	t
m	m	Nasal bilabial	Milch	p
n	n	Nasal alveolar	Natur	t
ŋ	N	Nasal velar	klingen	k
p	p	Oclusiva bilabial sorda	Park	p
͡pf	pf	Africada labiodental sorda	Apfel	
ʀ	R	Vibrante uvular	Regen	

IPA	X-SAMPA	Descripción	Ejemplo	Visema
s	s	fricativa alveolar sorda	Messer	s
ʃ	S	Fricativa postalveolar sorda	Fischer	S
t	t	Oclusiva alveolar sorda	Topf	T
͡ts	Ts	Africada alveolar sorda	Zahl	
͡tʃ	tS	Africada postalveolar sorda	deutsch	S
v	v	Fricativa labiodental sonora	Wasser	f
x	x	Fricativa velar sorda	kochen	k
z	z	Fricativa alveolar sonora	See	s
ʒ	Z	Fricativa postalveolar sonora	Orange	S
Vocales				
øː	2:	vocal larga semicerrada anterior redondeada	böse	o
ɐ	6	vocal casi abierta central	besser	a

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ɐ̯	6_^	vocal no silábica semiabierta central	Klar	a
œ	9	vocal semiabierta anterior redondeada	können	O
ə	@	vocal media central	Rede	@
a	a	vocal abierta anterior no redondeada	Salz	a
a:	a:	vocal larga abierta anterior no redondeada	Sahne	a
aɪ	aI	diptongo	nein	a
aʊ	aU	diptongo	Augen	a
ɑ̃	A~	vocal nasal posterior abierta no redondeada	Restaurant	a
e:	e:	vocal larga semicerrada anterior no redondeada	Rede	e
ɛ	E	vocal semiabierta anterior no redondeada	Keller	E

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ɛ̃	E~	vocal nasal semiabierta anterior no redondeada	Terrain	E
iː	i:	vocal larga cerrada anterior no redondeada	Lied	i
ɪ	I	vocal casi cerrada semianterior no redondeada	bitte	i
oː	o:	vocal larga semicerra da posterior redondeada	Kohl	o
ɔ	O	vocal semiabier ta posterior redondeada	Koffer	O
ɔ̃	O~	vocal nasal semiabier ta posterior redondeada	Annonce	O
ɔʏ	OY	diptongo	neu	O
uː	u:	vocal larga cerrada posterior redondeada	Bruder	u
ʊ	U	vocal casi cerrada semiposterior redondeada	Wunder	u

IPA	X-SAMPA	Descripción	Ejemplo	Visema
y:	y:	vocal larga cerrada anterior redondeada	kühl	u
ʏ	S	vocal casi cerrada semianterior redondeada	Küche	u
Símbolos adicionales				
ˈ	"	acento principal	Alabama	
ˌ	%	acento secundario	Alabama	
.	.	límite de sílaba	A.la.ba.ma	

Alemán (estándar suizo) (de-CH)

En la tabla siguiente se muestra el conjunto de fonemas del alfabeto fonético internacional (IPA, por sus siglas en inglés), los símbolos del alfabeto fonético extendido SAM (X-SAMPA, por sus siglas en inglés) y los correspondientes visemas para las voces del idioma alemán (estándar suizo) admitidas por Amazon Polly.

Tabla de fonemas/visemas

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Consonantes				
ʔ	?	oclusiva glotal		
b	b	oclusiva bilabial sonora	Bier	p
d	d	oclusiva alveolar sonora	Dach	t

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ç	C	fricativa palatal sorda	ich	k
͡dʒ	dZ	africada postalveolar sonora	Dschungel	S
f	f	Fricativa labiodental sorda	Vogel	f
g	g	Oclusiva velar sonora	Gabel	k
h	h	Fricativa glotal sorda	Haus	k
j	j	Fricativa glotal sorda	jemand	i
k	k	Oclusiva velar sorda	Kleid	k
l	l	Aproximante alveolar lateral	Loch	t
m	m	Nasal bilabial	Milch	p
n	n	Nasal alveolar	Natur	t
ŋ	N	Nasal velar	klingen	k
p	p	Oclusiva bilabial sorda	Park	p
͡pf	pf	Africada labiodental sorda	Apfel	
ʀ	R	Vibrante uvular	Regen	

IPA	X-SAMPA	Descripción	Ejemplo	Visema
s	s	fricativa alveolar sorda	Messer	s
ʃ	S	Fricativa postalveolar sorda	Fischer	S
t	t	Oclusiva alveolar sorda	Topf	T
͡ts	Ts	Africada alveolar sorda	Zahl	
͡tʃ	tS	Africada postalveolar sorda	deutsch	S
v	v	Fricativa labiodental sonora	Wasser	f
x	x	Fricativa velar sorda	kochen	k
z	z	Fricativa alveolar sonora	See	s
ʒ	Z	Fricativa postalveolar sonora	Orange	S
Vocales				
øː	2:	vocal larga semicerrada anterior redondeada	böse	o
ɐ	6	vocal casi abierta central	besser	a

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ɐ̯	6_^	vocal no silábica semiabierta central	Klar	a
œ	9	vocal semiabierta anterior redondeada	können	O
ə	@	vocal media central	Rede	@
a	a	vocal abierta anterior no redondeada	Salz	a
a:	a:	vocal larga abierta anterior no redondeada	Sahne	a
aɪ	aI	diptongo	nein	a
aʊ	aU	diptongo	Augen	a
ɑ̃	A~	vocal nasal posterior abierta no redondeada	Restaurant	a
e:	e:	vocal larga semicerrada anterior no redondeada	Rede	e
ɛ	E	vocal semiabierta anterior no redondeada	Keller	E

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ɛ̃	E~	vocal nasal semiabierta anterior no redondeada	Terrain	E
iː	i:	vocal larga cerrada anterior no redondeada	Lied	i
ɪ	I	vocal casi cerrada semianterior no redondeada	bitte	i
oː	o:	vocal larga semicerra da posterior redondeada	Kohl	o
ɔ	O	vocal semiabier ta posterior redondeada	Koffer	O
ɔ̃	O~	vocal nasal semiabier ta posterior redondeada	Annonce	O
ɔʏ	OY	diptongo	neu	O
uː	u:	vocal larga cerrada posterior redondeada	Bruder	u
ʊ	U	vocal casi cerrada semiposterior redondeada	Wunder	u

IPA	X-SAMPA	Descripción	Ejemplo	Visema
y:	y:	vocal larga cerrada anterior redondeada	kühl	u
ʏ	S	vocal casi cerrada semianterior redondeada	Küche	u
Símbolos adicionales				
ˈ	“	acento principal	Alabama	
ˌ	%	acento secundario	Alabama	
.	.	límite de sílaba	A.la.ba.ma	

Hindi (hi-IN)

En la tabla siguiente se muestra el conjunto de fonemas del Alfabeto Fonético Internacional (IPA, por sus siglas en inglés), los símbolos del Alfabeto Fonético Extendido SAM (X-SAMPA, por sus siglas en inglés) y el tipo de sonido del fonema para las voces del idioma hindi admitidas por Amazon Polly.

Para determinar los fonemas que se usan conjuntamente con el idioma hindi, consulte Inglés (India) (en-IN).

Tabla de fonemas/visemas

IPA	X-SAMPA	Descripción	Ejemplo
Consonantes			
p^h	p_h	oclusiva bilabial aspirada sorda	फूल (phool)
b^h	b_h	oclusiva bilabial aspirada sonora	भारी (bhaari)

IPA	X-SAMPA	Descripción	Ejemplo
t̪	t_d	oclusiva dental sorda	तापमान (taapmaan)
t̪ʰ	t_d_h	oclusiva dental aspirada sorda	थोड़ा (thoda)
d̪	d_d	oclusiva dental sonora	दिल्ली (dilli)
d̪ʰ	d_d_h	oclusiva dental aspirada sonora	धोबी (dhobi)
ʈ	t`	oclusiva retrofleja sorda	कटोरा (katora)
ʈʰ	t`_h	oclusiva retrofleja aspirada sorda	ठंड (thand)
ɖ	d`	oclusiva retrofleja sonora	डर (darr)
ɖʰ	d`_h	oclusiva retrofleja aspirada sonora	ढाल (dhal)
ʧʰ	tS_h	africada palatal aspirada sorda	छाल (chaal)
ʤʰ	dZ_h	africada palatal aspirada sonora	झाल (jhaal)
kʰ	k_h	oclusiva velar aspirada sorda	खान (khan)
gʰ	g_h	oclusiva velar aspirada sonora	घान (ghaan)
ɳ	n`	nasal retrofleja	क्षण (kshan)
ɾ	4	vibrante simple alveolar	राम (ram)
ɽ	r`	vibrante simple retrofleja normal	बड़ा (bada)

IPA	X-SAMPA	Descripción	Ejemplo
ɽʱ	r`_h	vibrante simple retrofleja aspirada sonora	बढ़ी (barhi)
ʊ	v\	aproximante bilabial	वसूल (wasool)
Vocales			
ə	@_o	vocal media central	अच्छा (achhaa)
ə̃	@~	vocal media central nasalizada	हँसना (hansnaa)
a	A_o	vocal abierta anterior no redondeada	आग (aag)
ã	A~	vocal nasalizada abierta anterior no redondeada	घ़डयिँा (ghariyaan)
ɪ	I_o	vocal casi cerrada semianterior no redondeada	इक्कीस (ikkees)
ɪ̃	I~	vocal nasalizada casi cerrada semianterior no redondeada	संचिाई (sinchai)
i	i_o	vocal cerrada anterior no redondeada	बल्लिी (billee)
ĩ	i~	vocal nasalizada cerrada anterior no redondeada	नही (nahin)
ʊ	U_o	vocal casi cerrada semiposterior redondeada	उलूल (ullu)

IPA	X-SAMPA	Descripción	Ejemplo
ʊ̃	U~	vocal nasalizada casi cerrada semiposterior redondeada	मुँह (munh)
u	u_o	vocal semicerrada posterior redondeada	फूल (phool)
ũ	u~	vocal nasalizada cerrada posterior redondeada	ऊँट (oont)
ɔ	O_o	vocal semiabierta posterior redondeada	कौन (kaun)
ɔ̃	O~	vocal nasalizada semiabierta posterior redondeada	भौ (bhaun)
o	o	vocal semicerrada posterior redondeada	सोना (sona)
õ	o~	vocal nasalizada semicerrada posterior redondeada	क्यों (kyon)
ɛ	E_o	vocal semiabierta anterior no redondeada	पैसा (paisa)
ɛ̃	E~	vocal nasalizada semiabierta anterior no redondeada	मैं (main)
e	e	vocal semicerrada anterior no redondeada	एक (ek)
ẽ	e~	vocal nasalizada semicerrada anterior no redondeada	किताबें (kitabein)

Islandés (is-IS)

En la tabla siguiente se muestra el conjunto de fonemas del Alfabeto Fonético Internacional (IPA, por sus siglas en inglés), los símbolos del Alfabeto Fonético Extendido SAM (X-SAMPA, por sus siglas en inglés) y los correspondientes visemas para las voces del idioma islandés admitidas por Amazon Polly.

Tabla de fonemas/visemas

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Consonantes				
b	b	oclusiva bilabial sonora	grasbakkanum	0
c	c	oclusiva palatal sorda	pakkin	k
c^h	c_h	oclusiva aspirada palatal sorda	anarkistai	k
ç	C	fricativa palatal sorda	héðan	k
d	d	oclusiva alveolar sonora	bóndi	t
ð	D	fricativa dental sonora	borð	T
f	f	fricativa labiodental sorda	duft	f
g	g	oclusiva velar sonora	holgóma	k
ɣ	G	fricativa velar sonora	hugur	k

IPA	X-SAMPA	Descripción	Ejemplo	Visema
h	h	fricativa glotal sorda	heili	k
j	j	aproximante palatal	jökull	i
kʰ	k_h	oclusiva velar aspirada sorda	ósköpunum	k
l	l	aproximante alveolar lateral	gólf	t
l̥	l_0	aproximante lateral alveolar sorda	fólk	t
m	m	nasal bilabial	september	p
m̥	m_0	nasal bilabial sorda	kompa	p
n	n	nasal alveolar	númer	t
n̥	n_0	nasal alveolar sorda	pöntun	t
ɲ	J	nasal palatal	pælingar	J
ŋ	N	nasal velar	söngvarann	k
ŋ̊	N_0	nasal velar sorda	frænka	k
pʰ	p_h	oclusiva bilabial aspirada sorda	afplánun	p
r	r	vibrante alveolar	afskrifta	r
r̥	r_0	vibrante alveolar sorda	andvörpum	r

IPA	X-SAMPA	Descripción	Ejemplo	Visema
s	s	fricativa alveolar sorda	baðhús	s
t^h	t_h	oclusiva alveolar aspirada sorda	tanki	t
θ	T	fricativa dental sorda	þeldökki	T
v	v	fricativa labiodental sonora	silfur	f
w	w	aproximante velo-labial		u
x	x	fricativa velar sorda	samfélags	k
Vocales				
œ	9	vocal semiabierta anterior redondeada	þröskuldinum	O
œ:	9:	vocal larga semiabierta anterior redondeada	tvö	O
a	a	vocal abierta anterior no redondeada	nefna	a
a:	a:	vocal larga abierta anterior no redondeada	fara	a
au	au	diptongo	átta	a

IPA	X-SAMPA	Descripción	Ejemplo	Visema
au:	au:	diptongo	átján	a
ɛ	E	vocal semiabierta anterior no redondeada	kennari	E
ɛ:	E:	vocal larga semiabierta anterior no redondeada	dreka	E
i	i	vocal cerrada anterior no redondeada	Gúlíver	i
i:	i:	vocal larga cerrada anterior no redondeada	þrír	i
ɪ	I	vocal casi cerrada semianterior no redondeada	samspil	i
ɪ:	I:	vocal larga casi cerrada semianterior no redondeada	stig	i
ɔ	O	vocal semiabierta posterior redondeada	regndropar	O
ɔ:	O:	vocal larga semiabierta posterior redondeada	ullarbolur	O
ɔu	Ou	diptongo	tólf	O

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ɔu:	Ou:	diptongo	fjórir	O
u	u	vocal semicerrada posterior redondeada	stúlkan	u
u:	u:	vocal larga cerrada posterior redondeada	frú	u
ʏ	S	vocal casi cerrada semianterior redondeada	tíu	u
ʏ:	S	vocal larga casi cerrada semianterior redondeada	gruninn	u
Símbolos adicionales				
ˈ	"	acento principal	Alabama	
ˌ	%	acento secundario	Alabama	
.	.	límite de sílaba	A.la.ba.ma	

Italiano (it-IT)

En la tabla siguiente se muestra el conjunto de fonemas del Alfabeto Fonético Internacional (IPA, por sus siglas en inglés), los símbolos del Alfabeto Fonético Extendido SAM (X-SAMPA, por sus siglas en inglés) y los correspondientes visemas para las voces del idioma italiano admitidas por Amazon Polly.

Tabla de fonemas/visemas

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Consonantes				
b	b	oclusiva bilabial sonora	bacca	p
d	d	oclusiva alveolar sonora	dama	t
͡dz	dz	africada alveolar sonora	zero	s
͡dʒ	dZ	africada postalveolar sonora	giro	S
f	f	fricativa labiodental sorda	famiglia	f
g	g	oclusiva velar sonora	gatto	k
h	h	fricativa glotal sorda	horror	k
j	j	aproximante palatal	dieci	i
k	k	oclusiva velar sorda	campo	k
l	l	aproximante alveolar lateral	lido	t
ʎ	L	aproximante palatal lateral	aglio	J
m	m	nasal bilabial	mille	p

IPA	X-SAMPA	Descripción	Ejemplo	Visema
n	n	nasal alveolar	nove	t
ɲ	J	nasal palatal	lasagne	J
p	p	oclusiva bilabial sorda	pizza	p
r	r	vibrante alveolar	risata	r
s	s	fricativa alveolar sorda	sei	s
ʃ	S	fricativa postalveolar sorda	scienza	S
t	t	oclusiva alveolar sorda	tavola	t
͡ts	ts	africada alveolar sorda	forza	s
͡tʃ	tS	africada postalveolar sorda	cielo	S
v	v	fricativa labiodental sonora	venti	f
w	w	aproximante velo-labial	quattro	u
z	z	fricativa alveolar sonora	bisogno	s
ʒ	Z	fricativa postalveolar sonora	bijou	S
Vocales				

IPA	X-SAMPA	Descripción	Ejemplo	Visema
a	a	vocal abierta anterior no redondeada	arco	a
e	e	vocal semicerrada anterior no redondeada	tre	e
ɛ	E	vocal semiabierta anterior no redondeada	ettaro	E
i	i	vocal cerrada anterior no redondeada	impero	i
o	o	vocal semicerrada posterior redondeada	cento	o
ɔ	O	vocal semiabierta posterior redondeada	otto	O
u	u	vocal semicerrada posterior redondeada	uno	u
Símbolos adicionales				
ˈ	"	acento principal	Alabama	
ˌ	%	acento secundario	Alabama	
.	.	límite de sílaba	A.la.ba.ma	

Japonés (ja-JP)

Amazon Polly admite los alfabetos de pronunciación kana y yomigana en japonés. Para que Amazon Polly utilice la pronunciación fonética con estos alfabetos, utilice el atributo del fonema `alphabet="x-amazon-`*`phonetic standard used`*`"`.

- `x-amazon-pron-kana`: indica que se utiliza la pronunciación Kana. La pronunciación kana son caracteres especiales del katakana que se utilizan para la transcripción fonética y pueden codificar el acento tonal.
- `x-amazon-yomigana`: indica que se utiliza el yomigana. Los yomigana pueden ser alfabetos convencionales en katakana, hiragana y latín, interpretados como romanización Hepburn.

En los siguientes ejemplos se muestra cómo utilizan:

Pronunciación: 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>
```

En la tabla siguiente se muestra el conjunto de fonemas del Alfabeto Fonético Internacional (IPA, por sus siglas en inglés), los símbolos del Alfabeto Fonético Extendido SAM (X-SAMPA, por sus siglas en inglés) y los correspondientes visemas para las voces del idioma japonés admitidas por Amazon Polly.

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Consonantes				

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ɾ	4	vibrante simple alveolar	練習, renshuu	t
ʔ	?	oclusiva glotal	あつっ, atsu'	
b	b	oclusiva bilabial sonora	舞踊, buyou	p
β	B	fricativa bilabial sonora	ヴィンテージ, vinteeji	B
c	c	oclusiva palatal sorda	ききょう, kikyou	k
ç	C	fricativa palatal sorda	人, hito	k
d	d	oclusiva alveolar sonora	濁点, dakuten	t
͡dʑ	dz\	africada alveolo-palatal sonora	純, jun	J
g	g	oclusiva velar sonora	ご飯, gohan	k
h	h	fricativa glotal sorda	本, hon	k
j	j	aproximante palatal	屋根, yane	i
ɟ	J\	oclusiva palatal sonora	行儀, gyougi	J
k	k	oclusiva velar sorda	漢字, kanji	k

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ɺ	l\	vibrante lateral alveolar	釣り, tsuri	r
ɺj	l\j	vibrante lateral alveolar, aproximante palatal	流行, ryuukou	r
m	m	nasal bilabial	飯, meshi	p
n	n	nasal alveolar	猫, neko	t
ɲ	J	nasal palatal	日本, nippon	J
ɴ	N\	nasal uvular	缶, kan	k
p	p	oclusiva bilabial sorda	パン, pan	p
ɸ	p\	fricativa bilabial sorda	福, huku	f
s	s	fricativa alveolar sorda	層, sou	s
ɕ	s\	fricativa alveolo-palatal sorda	書簡, shokan	J
t	t	oclusiva alveolar sorda	手紙, tegami	t
t͡s	ts	africada alveolar sorda	釣り, tsuri	s
t͡ɕ	ts\	africada alveolo-palatal sorda	吉, kichi	J

IPA	X-SAMPA	Descripción	Ejemplo	Visema
w	w	aproximante velo-labial	電話, denwa	u
z	z	fricativa alveolar sonora	座敷, zashiki	s
Vocales				
äː	a:_"	vocal larga abierta central no redondeada	羽蟻, haari	a
ä	a_"	vocal abierta central no redondeada	仮名, kana	a
eː	e:_o	vocal larga semianterior no redondeada	学生, gakusei	@
e	e_o	vocal semianterior no redondeada	歴, reki	@
i	i	vocal cerrada anterior no redondeada	気, ki	i
iː	i:	vocal larga cerrada anterior no redondeada	詩歌, shiika	i
ɯ	M	vocal cerrada posterior no redondeada	運, un	i

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ɯː	M:	vocal larga cerrada posterior no redondeada	宗教, shuukyou	i
oː	o:_o	vocal larga semiposterior redondeada	購読, koodoku	o
o	o_o	vocal semiposterior redondeada	読者, dokusha	o

Coreano (ko-KR)

En la tabla siguiente se muestra el conjunto de fonemas del Alfabeto Fonético Internacional (IPA, por sus siglas en inglés), los símbolos del Alfabeto Fonético Extendido SAM (X-SAMPA, por sus siglas en inglés) y los correspondientes visemas para las voces del idioma coreano admitidas por Amazon Polly.

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Consonantes				
k	k	oclusiva velar sorda	강, [g]ang	k
k#	k_t	oclusiva velar fuerte sorda	깨, [kk]e	k
n	n	nasal alveolar	남, [n]am	t
t	t	oclusiva alveolar sorda	도, [d]o	t
t#	t_t	oclusiva alveolar fuerte sorda	때, [tt]e	t

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ɾ	4	vibrante simple alveolar	사랑, sa[r]ang	t
l	l	aproximante alveolar lateral	돌, do[l]	t
m	m	nasal bilabial	무, [m]u	p
p	p	oclusiva bilabial sorda	봄, [b]om	p
p#	p_t	oclusiva bilabial fuerte sorda	뻘, [pp]eol	p
s	s	fricativa alveolar sorda	새, [s]e	s
s#	s_t	fricativa alveolar fuerte sorda	씨, [ss]i	s
ŋ	N	nasal velar	방, ba[ng]	k
t͡ɕ	ts\	africada alveolo-palatal sorda	조, [j]o	J
t#͡ɕ	ts_t	africada alveolo-palatal fuerte sorda	찌, [jj]i	J
t͡ɕʰ	ts_h	africada alveolo-palatal aspirada sorda	차, [ch]a	J
kʰ	k_h	oclusiva velar aspirada sorda	코, [k]o	k
tʰ	t_h	oclusiva alveolar aspirada sorda	통, [t]ong	t

IPA	X-SAMPA	Descripción	Ejemplo	Visema
p^h	p_h	oclusiva bilabial aspirada sorda	패, [p]e	p
h	h	fricativa glotal sorda	힘, [h]im	k
j	j	aproximante palatal	양, [y]ang	i
w	w	aproximante velo-labial	왕, [w]ang	u
ɯ	M\	aproximante velar>	의, [wj]i	i
Vocales				
a	a	vocal abierta anterior no redondeada	밥, b[a]b	a
ʌ	V	vocal semiabierta posterior no redondeada	정, j[eo]ng	E
ɛ	E	vocal semiabierta anterior no redondeada	배, b[e]	E
o	o	vocal semicerrada posterior redondeada	노, n[o]	o
u	u	vocal semicerrada posterior redondeada	둘, d[u]l	u

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ɯ	M	vocal cerrada posterior no redondeada	은, [eu]n	i
i	i	vocal cerrada anterior no redondeada	김, k[i]m	i

Noruego (nb-NO)

En la tabla siguiente se muestra el conjunto completo de fonemas del Alfabeto Fonético Internacional (IPA, por sus siglas en inglés) y símbolos del Alfabeto Fonético Extendido SAM (X-SAMPA, por sus siglas en inglés), así como sus correspondientes visemas, tal y como se utilizan en las voces de Amazon Polly para el idioma noruego.

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Consonantes				
ɾ	4	vibrante simple alveolar	prøv	t
b	b	oclusiva bilabial sonora	labb	p
ç	C	fricativa palatal sorda	kino	k
d	d	oclusiva alveolar sonora	ladd	t
ɖ	d`	oclusiva retrofleja sonora	verdi	t
f	f	fricativa labiodental sorda	fot	f

IPA	X-SAMPA	Descripción	Ejemplo	Visema
g	g	oclusiva velar sonora	tagg	k
h	h	fricativa glotal sorda	ha	k
j	j	aproximante palatal	gi	i
k	k	oclusiva velar sorda	takk	k
l	l	aproximante alveolar lateral	fall, ball	t
ɭ	l`	aproximante lateral retrofleja	ærlig	t
m	m	nasal bilabial	lam	p
n	n	nasal alveolar	vann	t
ɳ	n`	nasal retrofleja	garn	t
ŋ	N	nasal velar	sang	k
p	p	oclusiva bilabial sorda	hopp	p
s	s	fricativa alveolar sorda	lass	s
ʂ	s`	fricativa retrofleja sorda	års	S
ʃ	S	fricativa postalveo lar sorda	skyt	S

IPA	X-SAMPA	Descripción	Ejemplo	Visema
t	t	oclusiva alveolar sorda	lat	t
ʈ	t`	oclusiva retrofleja sorda	hardt	t
ʋ	v\	aproximante labiodental	vin	f
w	w	aproximante velo-labial	will	x
Vocales				
øː	2:	vocal larga semicerrada anterior redondeada	søt	o
œ	9	vocal semiabierta anterior redondeada	søtt	O
ə	@	vocal media central	ape	@
æː	{:	vocal larga casi abierta anterior no redondeada	vær	a
ʉ	}	vocal cerrada central redondeada	lund	u
ʉː	}:	vocal larga cerrada central redondeada	lun	u

IPA	X-SAMPA	Descripción	Ejemplo	Visema
æ	{	vocal casi abierta anterior no redondeada	vært	a
ɑ	A	vocal abierta posterior no redondeada	hatt	a
ɑː	A:	vocal larga abierta posterior no redondeada	hat	a
eː	e:	vocal larga semicerrada anterior no redondeada	sen	e
ɛ	E	vocal semiabierta anterior no redondeada	send	E
iː	i:	vocal larga cerrada anterior no redondeada	vin	i
ɪ	I	vocal casi cerrada semianterior no redondeada	vind	i
oː	o:	vocal larga semicerrada posterior redondeada	våt	o
ɔ	O	vocal semiabierta posterior redondeada	vått	O

IPA	X-SAMPA	Descripción	Ejemplo	Visema
u:	u:	vocal larga cerrada posterior redondeada	bok	u
ʊ	U	vocal casi cerrada semiposterior redondeada	bukk	u
y:	y:	vocal larga cerrada anterior redondeada	lyn	u
ʏ	S	vocal casi cerrada semianterior redondeada	lynne	u
Símbolos adicionales				
ˈ	"	acento principal	Alabama	
ˌ	%	acento secundario	Alabama	
.	.	límite de sílaba	A.la.ba.ma	

Polaco (pl-PL)

En la tabla siguiente se muestra el conjunto de fonemas del Alfabeto Fonético Internacional (IPA, por sus siglas en inglés), los símbolos del Alfabeto Fonético Extendido SAM (X-SAMPA, por sus siglas en inglés) y los correspondientes visemas para las voces del idioma polaco admitidas por Amazon Polly.

Tabla de fonemas/visemas

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Consonantes				

IPA	X-SAMPA	Descripción	Ejemplo	Visema
b	b	oclusiva bilabial sonora	bobas, belka	p
d	d	oclusiva alveolar sonora	dar, do	t
͡dz	dz	africada alveolar sonora	dzwon, widzowie	s
͡dʑ	dz\	africada alveolo-palatal sonora	dźwięk	J
͡dʐ	dz`	africada retrofleja sonora	dżem, dżungla	S
f	f	fricativa labiodental sorda	furtka, film	f
g	g	oclusiva velar sonora	gazeta, waga	k
h	h	fricativa glotal sorda	chleb, handel	k
j	j	aproximante palatal	jak, maja	i
k	k	oclusiva velar sorda	kura, marek	k
l	l	aproximante alveolar lateral	lipa, alicja	t
m	m	nasal bilabial	matka, molo	p
n	n	nasal alveolar	norka	t
ɲ	J	nasal palatal	koń, toruń	J

IPA	X-SAMPA	Descripción	Ejemplo	Visema
p	p	oclusiva bilabial sorda	pora, stop	p
r	r	vibrante alveolar	rok, park	r
s	s	fricativa alveolar sorda	sum, pas	s
ɕ	s\	fricativa alveolo-palatal sorda	śruba, śnieg	J
ʂ	s`	fricativa retrofleja sorda	szum, masz	S
t	t	oclusiva alveolar sorda	tok, stół	t
͡ts	ts	africada alveolar sorda	car, co	s
͡tɕ	ts\	africada alveolo-palatal sorda	ćma, mieć	J
͡tʂ	ts`	africada retrofleja sorda	czas, raczej	S
v	v	fricativa labiodental sonora	worek, mewa	f
w	w	aproximante velo-labial	łaska, mało	u
z	z	fricativa alveolar sonora	zero	s
ʑ	z\	fricativa alveolo-palatal sonora	źrebię, bieliźnie	J

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ʐ̩	z`	fricativa retrofleja sonora	żar, żona	S
Vocales				
a	a	vocal abierta anterior no redondeada	ja	a
ɛ	E	vocal semiabierta anterior no redondeada	echo	E
ɛ̃	E~	vocal nasal semiabierta anterior no redondeada	węże	E
i	i	vocal cerrada anterior no redondeada	ile	i
ɔ	O	vocal semiabierta posterior redondeada	oczy	O
ɔ̃	O~	vocal nasal semiabierta posterior redondeada	wąż	O
u	u	vocal semicerrada posterior redondeada	uczta	u

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ɨ	1	vocal cerrada central no redondeada	byk	i
Símbolos adicionales				
ˈ	"	acento principal	Alabama	
ˌ	%	acento secundario	Alabama	
.	.	límite de sílaba	A.la.ba.ma	

Portugués (pt-PT)

En la tabla siguiente se muestra el conjunto de fonemas del Alfabeto Fonético Internacional (IPA, por sus siglas en inglés), los símbolos del Alfabeto Fonético Extendido SAM (X-SAMPA, por sus siglas en inglés) y los correspondientes visemas para las voces del idioma portugués admitidas por Amazon Polly.

Tabla de fonemas/visemas

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Consonantes				
ɾ	4	vibrante simple alveolar	pira	t
b	b	oclusiva bilabial sonora	dato	p
d	d	oclusiva alveolar sonora	dato	t
f	f	fricativa labiodental sorda	facto	f

IPA	X-SAMPA	Descripción	Ejemplo	Visema
g	g	oclusiva velar sonora	gato	k
j	j	aproximante palatal	paraguay	i
k	k	oclusiva velar sorda	cacto	k
l	l	aproximante alveolar lateral	galo	t
ʎ	L	aproximante palatal lateral	galho	J
m	m	nasal bilabial	mato	p
n	n	nasal alveolar	nato	t
ɲ	J	nasal palatal	pinha	J
p	p	oclusiva bilabial sorda	pato	p
ʁ	R\	vibrante uvular	barroso	k
s	s	fricativa alveolar sorda	saca	s
ʃ	S	fricativa postalveolar sorda	chato	S
t	t	oclusiva alveolar sorda	tacto	t
v	v	fricativa labiodental sonora	vaca	f

IPA	X-SAMPA	Descripción	Ejemplo	Visema
w	w	aproximante velo-labial	mau	u
z	z	fricativa alveolar sonora	zaca	s
ʒ	Z	fricativa postalveolar sonora	jacto	S
Vocales				
a	a	vocal abierta anterior no redondeada	parto	a
ã	a~	vocal nasal abierta anterior no redondeada	pega	a
e	e	vocal semicerrada anterior no redondeada	pega	e
ẽ	e~	vocal nasal semicerrada anterior no redondeada	movem	e
ɛ	E	vocal semiabierta anterior no redondeada	café	E
i	i	vocal cerrada anterior no redondeada	lingueta	i

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ĩ	i~	vocal nasal cerrada anterior no redondeada	cinto	i
o	o	vocal semicerrada posterior redondeada	poder	o
õ	o~	vocal nasal semicerrada posterior redondeada	compra	o
ɔ	O	vocal semiabierta posterior redondeada	cotó	O
u	u	vocal semicerrada posterior redondeada	fui	u
ũ	u~	vocal nasal cerrada posterior redondeada	sunto	u
Símbolos adicionales				
ˈ	"	acento principal	Alabama	
ˌ	%	acento secundario	Alabama	
.	.	límite de sílaba	A.la.ba.ma	

Portugués (Brasil) (pt-BR)

En la tabla siguiente se muestra el conjunto de fonemas del Alfabeto Fonético Internacional (IPA, por sus siglas en inglés), los símbolos del Alfabeto Fonético Extendido SAM (X-SAMPA, por sus siglas en inglés) y los correspondientes visemas para las voces del idioma portugués de Brasil admitidas por Amazon Polly.

Tabla de fonemas/visemas

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Consonantes				
ɾ	4	vibrante simple alveolar	pira	t
b	b	oclusiva bilabial sonora	bato	p
d	d	oclusiva alveolar sonora	dato	t
͡dʒ	dZ	africada postalveolar sonora	idade	S
f	f	fricativa labiodental sorda	facto	f
g	g	oclusiva velar sonora	gato	k
j	j	aproximante palatal	paraguay	i
k	k	oclusiva velar sorda	cacto	k
l	l	aproximante alveolar lateral	galo	t

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ʎ	L	aproximante palatal lateral	galho	J
m	m	nasal bilabial	mato	p
n	n	nasal alveolar	nato	t
ɲ	J	nasal palatal	pinha	J
p	p	oclusiva bilabial sorda	pato	p
s	s	fricativa alveolar sorda	saca	s
ʃ	S	fricativa postalveo lar sorda	chato	S
t	t	oclusiva alveolar sorda	tacto	t
͡tʃ	tS	africada postalveo lar sorda	noite	S
v	v	fricativa labiodental sonora	vaca	f
w	w	aproximante velo-labial	mau	u
χ	X	fricativa uvular sorda	carro	k
z	z	fricativa alveolar sonora	zaca	s
ʒ	Z	fricativa postalveo lar sonora	jacto	S

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Vocales				
a	a	vocal abierta anterior no redondeada	parto	a
ã	a~	vocal nasal abierta anterior no redondeada	pensamos	a
e	e	vocal semicerrada anterior no redondeada	pega	e
ẽ	e~	vocal nasal semicerrada anterior no redondeada	movem	e
ɛ	E	vocal semiabierta anterior no redondeada	café	E
i	i	vocal cerrada anterior no redondeada	lingueta	i
ĩ	i~	vocal nasal cerrada anterior no redondeada	cinto	i
o	o	vocal semicerrada posterior redondeada	poder	o

IPA	X-SAMPA	Descripción	Ejemplo	Visema
õ	o~	vocal nasal semicerrada posterior redondeada	compra	o
ɔ	O	vocal semiabierta posterior redondeada	cotó	O
u	u	vocal semicerrada posterior redondeada	fui	u
ũ	u~	vocal nasal cerrada posterior redondeada	sunto	u
Símbolos adicionales				
ˈ	"	acento principal	Alabama	
ˌ	%	acento secundario	Alabama	
.	.	límite de sílaba	A.la.ba.ma	

Rumano (ro-RO)

En la tabla siguiente se muestra el conjunto de fonemas del Alfabeto Fonético Internacional (IPA, por sus siglas en inglés), los símbolos del Alfabeto Fonético Extendido SAM (X-SAMPA, por sus siglas en inglés) y los correspondientes visemas para las voces del idioma rumano admitidas por Amazon Polly.

Tabla de fonemas/visemas

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Consonantes				

IPA	X-SAMPA	Descripción	Ejemplo	Visema
b	b	oclusiva bilabial sonora	bubă	p
d	d	oclusiva alveolar sonora	după	t
͡dʒ	dZ	africada postalveolar sonora	george	S
f	f	fricativa labiodental sorda	afacere	f
g	g	oclusiva velar sonora	agri#	k
h	h	fricativa glotal sorda	harpă	k
j	j	aproximante palatal	baie	i
k	k	oclusiva velar sorda	co#	k
l	l	aproximante alveolar lateral	lampa	t
m	m	nasal bilabial	mama	p
n	n	nasal alveolar	nor	t
p	p	oclusiva bilabial sorda	pilă	p
r	r	vibrante alveolar	rampă	r
s	s	fricativa alveolar sorda	soare	s

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ʃ	S	fricativa postalveolar sorda	ma#ină	S
t	t	oclusiva alveolar sorda	tata	t
͡ts	ts	africada alveolar sorda	#ară	s
͡tʃ	tS	africada postalveolar sorda	ceai	S
v	v	fricativa labiodental sonora	via#ă	f
w	w	aproximante velo-labial	beau	u
z	z	fricativa alveolar sonora	mozol	s
ʒ	Z	fricativa postalveolar sonora	joacă	S
Vocales				
ə	@	vocal media central	babă	@
a	a	vocal abierta anterior no redondeada	casa	a
e	e	vocal semicerrada anterior no redondeada	elan	e

IPA	X-SAMPA	Descripción	Ejemplo	Visema
e̯	e_^	Vocal semicerrada anterior no redondeada y no silábica	beau	e
i	i	vocal cerrada anterior no redondeada	mie	i
o	o	vocal semicerrada posterior redondeada	oră	o
oa	o_^a	diptongo	oare	o
u	u	vocal semicerrada posterior redondeada	unde	u
ɨ	1	vocal cerrada central no redondeada	România	i
Símbolos adicionales				
ˈ	"	acento principal	Alabama	
ˌ	%	acento secundario	Alabama	
.	.	límite de sílaba	A.la.ba.ma	

Ruso (ru-RU)

En la tabla siguiente se muestra el conjunto de fonemas del Alfabeto Fonético Internacional (IPA, por sus siglas en inglés), los símbolos del Alfabeto Fonético Extendido SAM (X-SAMPA, por sus siglas

en inglés) y los correspondientes visemas para las voces del idioma ruso admitidas por Amazon Polly.

Tabla de fonemas/visemas

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Consonantes				
b	b	oclusiva bilabial sonora	борт	p
b^j	b'	oclusiva bilabial palatal sonora	бюро	p
d	d	oclusiva alveolar sonora	дом	t
d^j	d'	oclusiva alveolar palatal sonora	дядя	t
f	f	fricativa labiodental sorda	флаг	f
f^j	f'	fricativa labiodental palatal sorda	февраль	f
g	g	oclusiva velar sonora	нога	k
g^j	g'	oclusiva velar palatal sonora	герой	k
j	j	aproximante palatal	дизайн, ящик	i
k	k	oclusiva velar sorda	кот	k
k^j	k'	oclusiva velar palatal sorda	кино	k

IPA	X-SAMPA	Descripción	Ejemplo	Visema
l	l	aproximante alveolar lateral	лампа	t
lʲ	l'	aproximante lateral alveolar palatal	лес	t
m	m	nasal bilabial	мама	p
mʲ	m'	nasal bilabial palatal	мяч	p
n	n	nasal alveolar	нос	t
nʲ	n'	nasal alveolar palatal	няня	t
p	p	oclusiva bilabial sorda	папа	p
pʲ	p'	oclusiva bilabial palatal sorda	перо	p
r	r	vibrante alveolar	роза	r
rʲ	r'	vibrante alveolar palatal	рюмка	r
s	s	fricativa alveolar sorda	сыр	s
sʲ	s'	fricativa alveolar palatal sorda	сердце, русь	s
ɕː	s\:	fricativa larga alveolo-palatal sorda	щека	J

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ʂ	s`	fricativa retrofleja sorda	шум	S
t	t	oclusiva alveolar sorda	точка	t
tʲ	t'	oclusiva alveolar palatal sorda	тётя	t
t͡s	ts	africada alveolar sorda	царь	s
t͡ɕ	ts\	africada alveolo-palatal sorda	час	J
v	v	fricativa labiodental sonora	вор	f
vʲ	v'	fricativa labiodental palatal sonora	верфь	f
x	x	fricativa velar sorda	хор	k
xʲ	x'	fricativa velar palatal sorda	химия	k
z	z	fricativa alveolar sonora	зуб	s
zʲ	z'	fricativa alveolar palatal sonora	зима	s
ʑː	z\:	fricativa larga alveolo-palatal sonora	уезжать	J
ʐ	z`	fricativa retrofleja sonora	жена	S

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Vocales				
ə	@	vocal media central	канарейка	@
a	a	vocal abierta anterior no redondeada	два, яблоко	a
e	e	vocal semicerrada anterior no redondeada	печь	e
ɛ	E	vocal semiabierta anterior no redondeada	это	E
i	i	vocal cerrada anterior no redondeada	один, четыре	i
o	o	vocal semicerrada posterior redondeada	кот	o
u	u	vocal semicerrada posterior redondeada	муж, вьюга	u
ɨ	1	vocal cerrada central no redondeada	мышь	i

Español (es-ES)

En la tabla siguiente se muestra el conjunto de fonemas del Alfabeto Fonético Internacional (IPA, por sus siglas en inglés), los símbolos del Alfabeto Fonético Extendido SAM (X-SAMPA, por sus siglas

en inglés) y los correspondientes visemas para las voces del idioma español admitidas por Amazon Polly.

Tabla de fonemas/visemas

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Consonantes				
ɾ	4	vibrante simple alveolar	pero, bravo, amor, eterno	t
b	b	oclusiva bilabial sonora	bestia	p
β	B	fricativa bilabial sonora	bebé	B
d	d	oclusiva alveolar sonora	cuando	t
ð	D	fricativa dental sonora	arder	T
f	f	fricativa labiodental sorda	fase, café	f
g	g	oclusiva velar sonora	gato, lengua, guerra	k
ɣ	G	fricativa velar sonora	trigo, Argos	k
j	j	aproximante palatal	hacia, tierra, radio, viuda	i
ʝ	j\	fricativa palatal sonora	enhielar, sayo, inyectado, desyerba	J

IPA	X-SAMPA	Descripción	Ejemplo	Visema
k	k	oclusiva velar sorda	caña, laca, quisimos	k
l	l	aproximante alveolar lateral	lino, calor, principal	t
ʎ	L	aproximante palatal lateral	llave, pollo	J
m	m	nasal bilabial	madre, comer, anfibio	p
n	n	nasal alveolar	nido, anillo, sin	t
ɲ	J	nasal palatal	cabaña, ñoquis	J
ŋ	N	nasal velar	cinco, venga	k
p	p	oclusiva bilabial sorda	pozo, topo	p
r	r	vibrante alveolar	perro, enrachado	r
s	s	fricativa alveolar sorda	saco, casa, puertas	s
t	t	oclusiva alveolar sorda	tamiz, átomo	t
͡tʃ	tS	africada postalveolar sorda	chubasco	S
θ	T	fricativa dental sorda	cereza, zorro, lacero, paz	T
w	w	aproximante velo-labial	fuego, fuimos, cuota, cuadro	u

IPA	X-SAMPA	Descripción	Ejemplo	Visema
x	x	fricativa velar sorda	jamón, general, suje, reloj	k
z	z	fricativa alveolar sonora	rasgo, mismo	s
Vocales				
a	a	vocal abierta anterior no redondeada	tanque	a
e	e	vocal semicerrada anterior no redondeada	peso	e
i	i	vocal cerrada anterior no redondeada	cinco	i
o	o	vocal semicerrada posterior redondeada	bosque	o
u	u	vocal semicerrada anterior no redondeada	publicar	u
Símbolos adicionales				
ˈ	"	acento principal	Alabama	
ˌ	%	acento secundario	Alabama	
.	.	límite de sílaba	A.la.ba.ma	

Español (México) (es-MX)

En la tabla siguiente se muestra el conjunto de fonemas del Alfabeto Fonético Internacional (IPA, por sus siglas en inglés), los símbolos del Alfabeto Fonético Extendido SAM (X-SAMPA, por sus siglas en inglés) y los correspondientes visemas para la voz del idioma español de México admitida por Amazon Polly.

Tabla de fonemas/visemas

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Consonantes				
ɾ	4	vibrante simple alveolar	pero, bravo, amor, eterno	t
b	b	oclusiva bilabial sonora	bestia	p
β	B	fricativa bilabial sonora	bebé	B
d	d	oclusiva alveolar sonora	cuando	t
ð	D	fricativa dental sonora	arder	T
f	f	fricativa labiodental sorda	fase, café	f
g	g	oclusiva velar sonora	gato, lengua, guerra	k
ɣ	G	fricativa velar sonora	trigo, Argos	k
j	j	aproximante palatal	hacia, tierra, radio, viuda	i

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ʝ	j\	fricativa palatal sonora	enhielar, sayo, inyectado, desyerba	J
k	k	oclusiva velar sorda	caña, laca, quisimos	k
l	l	aproximante alveolar lateral	lino, calor, principal	t
m	m	nasal bilabial	madre, comer, anfibio	p
n	n	nasal alveolar	nido, anillo, sin	t
ɲ	J	nasal palatal	cabaña, ñoquis	J
ŋ	N	nasal velar	angosto, increíble	k
p	p	oclusiva bilabial sorda	pozo, topo	p
r	r	vibrante alveolar	perro, enrachado	r
s	s	fricativa alveolar sorda	saco, casa, puertas	s
ʃ	S	fricativa postalveolar sorda	show, flash	S
t	t	oclusiva alveolar sorda	tamiz, átomo	t
͡tʃ	tS	africada postalveolar sorda	chubasco	S
w	w	aproximante velo-labial	fuego, fuimos, cuota, cuadro	u

IPA	X-SAMPA	Descripción	Ejemplo	Visema
x	x	fricativa velar sorda	jamón, general, peaje, reloj	k
z	z	fricativa alveolar sonora	rasgo, mismo	s
Vocales				
a	a	vocal central abierta no redondeada	tanque	a
e	e	vocal semicerrada anterior no redondeada	peso	e
i	i	vocal cerrada anterior no redondeada	cinco	i
o	o	vocal semicerrada posterior redondeada	bosque	o
u	u	vocal semicerrada posterior redondeada	publicar	u
Símbolos adicionales				
ˈ	"	acento principal	Alabama	
ˌ	%	acento secundario	Alabama	
.	.	límite de sílaba	A.la.ba.ma	

Español (EE.UU.) (es-US)

En la tabla siguiente se muestra el conjunto de fonemas del Alfabeto Fonético Internacional (IPA, por sus siglas en inglés), los símbolos del Alfabeto Fonético Extendido SAM (X-SAMPA, por sus siglas en inglés) y los correspondientes visemas para las voces del idioma español de EE.UU. admitidas por Amazon Polly.

Tabla de fonemas/visemas

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Consonantes				
ɾ	4	vibrante simple alveolar	pero, bravo, amor, eterno	t
b	b	oclusiva bilabial sonora	bestia	p
β	B	fricativa bilabial sonora	bebé	B
d	d	oclusiva alveolar sonora	cuando	t
ð	D	fricativa dental sonora	arder	T
f	f	fricativa labiodental sorda	fase, café	f
g	g	oclusiva velar sonora	gato, lengua, guerra	k
ɣ	G	fricativa velar sonora	trigo, Argos	k
j	j	aproximante palatal	hacia, tierra, radio, viuda	i

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ʝ	j\	fricativa palatal sonora	enhielar, sayo, inyectado, desyerba	J
k	k	oclusiva velar sorda	caña, laca, quisimos	k
l	l	aproximante alveolar lateral	lino, calor, principal	t
m	m	nasal bilabial	madre, comer, anfibio	p
n	n	nasal alveolar	nido, anillo, sin	t
ɲ	J	nasal palatal	cabaña, ñoquis	J
ŋ	N	nasal velar	angosto, increíble	k
p	p	oclusiva bilabial sorda	pozo, topo	p
r	r	vibrante alveolar	perro, enrachado	r
s	s	fricativa alveolar sorda	saco, casa, puertas	s
ʃ	S	fricativa postalveolar sorda	show, flash	S
t	t	oclusiva alveolar sorda	tamiz, átomo	t
͡tʃ	tS	africada postalveolar sorda	chubasco	S
w	w	aproximante velo-labial	fuego, fuimos, cuota, cuadro	u

IPA	X-SAMPA	Descripción	Ejemplo	Visema
x	x	fricativa velar sorda	jamón, general, peaje, reloj	k
z	z	fricativa alveolar sonora	rasgo, mismo	s
Vocales				
a	a	vocal central abierta no redondeada	tanque	a
e	e	vocal semicerrada anterior no redondeada	peso	e
i	i	vocal cerrada anterior no redondeada	cinco	i
o	o	vocal semicerrada posterior redondeada	bosque	o
u	u	vocal semicerrada posterior redondeada	publicar	u
Símbolos adicionales				
ˈ	"	acento principal	Alabama	
ˌ	%	acento secundario	Alabama	
.	.	límite de sílaba	A.la.ba.ma	

Sueco (sv-SE)

En la tabla siguiente se muestra el conjunto de fonemas del Alfabeto Fonético Internacional (IPA, por sus siglas en inglés), los símbolos del Alfabeto Fonético Extendido SAM (X-SAMPA, por sus siglas en inglés) y los correspondientes visemas para las voces del idioma sueco admitidas por Amazon Polly.

Tabla de fonemas/visemas

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Consonantes				
b	b	oclusiva bilabial sonora	bil	p
d	d	oclusiva alveolar sonora	dal	t
ɖ	d`	oclusiva retrofleja sonora	bord	t
f	f	fricativa labiodental sorda	fil	f
g	g	oclusiva velar sonora	gås	k
h	h	fricativa glotal sorda	hal	k
j	j	aproximante palatal	jag	i
k	k	oclusiva velar sorda	kal	k
l	l	aproximante alveolar lateral	lös	t

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ɭ	l`	aproximante lateral retrofleja	härlig	t
m	m	nasal bilabial	mil	p
n	n	nasal alveolar	nålar	t
ɳ	n`	nasal retrofleja	barn	t
ŋ	N	nasal velar	ring	k
p	p	oclusiva bilabial sorda	pil	p
r	r	vibrante alveolar	ris	r
s	s	fricativa alveolar sorda	sil	s
ɕ	s\	fricativa alveolo-p alatal sorda	tjock	J
ʂ	s`	fricativa retrofleja sorda	fors, schlager	S
t	t	oclusiva alveolar sorda	tal	t
ʈ	t`	oclusiva retrofleja sorda	hjort	t
v	v	fricativa labiodental sonora	vår	f
w	w	aproximante velo-labial	aula, airways	u

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ɧ	x\	fricativa palatal velar sorda	sjuk	k
Vocales				
ø	2	vocal semicerrada anterior redondeada	föll, förr	o
øː	2:	vocal larga semicerrada anterior redondeada	föl, nöt, för	o
ɵ	8	vocal semicerrada central redondeada	buss, full	o
ə	@	vocal media central	pojken	@
ʉː	}:	vocal larga cerrada central redondeada	hus, ful	u
a	a	vocal abierta anterior no redondeada	hall, matt	a
æ	{	vocal casi abierta anterior no redondeada	herr	a
ɑː	A:	vocal larga abierta posterior no redondeada	hal, mat	a

IPA	X-SAMPA	Descripción	Ejemplo	Visema
e:	e:	vocal larga semicerrada anterior no redondeada	vet, hel	e
ɛ	E	vocal semiabierta anterior no redondeada	vett, rätt, hetta, häll	E
ɛː	E:	vocal larga semiabierta anterior no redondeada	säl, häl, här	E:
i:	i:	vocal larga cerrada anterior no redondeada	vit, sil	i:
ɪ	I	vocal casi cerrada semianterior no redondeada	vitt, sill	i
o:	o:	vocal larga semicerrada posterior redondeada	hål, mål	o
ɔ	O	vocal semiabierta posterior redondeada	håll, moll	O
u:	u:	vocal larga cerrada posterior redondeada	sol, bot	u

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ʊ	U	vocal casi cerrada semiposterior redondeada	bott	u
y	y	vocal cerrada anterior redondeada	bytt	u
y:	y:	vocal larga cerrada anterior redondeada	syl, syl	u
Símbolos adicionales				
ˈ	"	acento principal	Alabama	
ˌ	%	acento secundario	Alabama	
.	.	límite de sílaba	A.la.ba.ma	

Turco (tr-TR)

En la tabla siguiente se muestra el conjunto de fonemas del Alfabeto Fonético Internacional (IPA, por sus siglas en inglés), los símbolos del Alfabeto Fonético Extendido SAM (X-SAMPA, por sus siglas en inglés) y los correspondientes visemas para las voces del idioma turco admitidas por Amazon Polly.

Tabla de fonemas/visemas

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Consonantes				
ɾ	4	vibrante simple alveolar	durum	t

IPA	X-SAMPA	Descripción	Ejemplo	Visema
r̥	4_0_r	vibrante alveolar africada sorda	bir	t
ɾ	4_r	vibrante alveolar africada	raf	t
b	b	oclusiva bilabial sonora	raf	p
c	c	oclusiva palatal sorda	kedi	k
d	d	oclusiva alveolar sonora	dede	t
d͡ʒ	dZ	africada postalveolar sonora	cam	S
f	f	fricativa labiodental sorda	fare	f
g	g	oclusiva velar sonora	galibi	k
h	h	fricativa glotal sorda	hasta	k
j	j	aproximante palatal	yat	i
ɟ	J\	oclusiva palatal sonora	genç	J
k	k	oclusiva velar sorda	akıl	k
l	l	aproximante alveolar lateral	lale	t

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ɬ	5	aproximante lateral alveolar velarizada	labirent	t
m	m	nasal bilabial	maaş	p
n	n	nasal alveolar	anı	t
p	p	oclusiva bilabial sorda	ip	p
s	s	fricativa alveolar sorda	ses	s
ʃ	S	fricativa postalveolar sorda	aşı	S
t	t	oclusiva alveolar sorda	ütü	t
t͡ʃ	tS	africada postalveolar sorda	çaba	S
v	v	fricativa labiodental sonora	ekvator, kahveci, akvaryum, isveçli, teşviki, cetvel	f
z	z	fricativa alveolar sonora	ver	s
ʒ	Z	fricativa postalveolar sonora	azık	S
Vocales				
ø	2	vocal semicerrada anterior redondeada	göl	0

IPA	X-SAMPA	Descripción	Ejemplo	Visema
œ	9	vocal semiabierta anterior redondeada	banliyö	O
a	a	vocal abierta anterior no redondeada	kal	a
a:	a:	vocal larga abierta anterior no redondeada	davacı	a
æ	{	vocal casi abierta anterior no redondeada	özlem, güvenlik, gürel, somersault	a
e	e	vocal semicerrada anterior no redondeada	keçi	e
ɛ	E	vocal semiabierta anterior no redondeada	dede	E
i	i	vocal cerrada anterior no redondeada	bir	i
i:	i:	vocal larga cerrada anterior no redondeada	izah	i
ɪ	I	vocal casi cerrada semianterior no redondeada	keçi	i

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ɯ	M	vocal cerrada posterior no redondeada	kıl	i
o	o	vocal semicerrada posterior redondeada	kol	o
o:	o:	vocal larga semicerrada posterior redondeada	dolar	o
u	u	vocal semicerrada posterior redondeada	durum	u
u:	u:	vocal larga cerrada posterior redondeada	ruhum	u
ʊ	U	vocal casi cerrada semiposterior redondeada	dolu	u
y	y	vocal cerrada anterior redondeada	güvenlik	u
ʏ	S	vocal casi cerrada semianterior redondeada	aşı	u
Símbolos adicionales				
ˈ	"	acento principal	Alabama	

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ˈ	%	acento secundario	Alabama	
.	.	límite de sílaba	A.la.ba.ma	

Galés (cy-GB)

En la tabla siguiente se muestra el conjunto de fonemas del Alfabeto Fonético Internacional (IPA, por sus siglas en inglés), los símbolos del Alfabeto Fonético Extendido SAM (X-SAMPA, por sus siglas en inglés) y los correspondientes visemas para las voces del idioma galés admitidas por Amazon Polly.

Tabla de fonemas/visemas

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Consonantes				
b	b	oclusiva bilabial sonora	baban	p
d	d	oclusiva alveolar sonora	deg	t
ʤ	dZ	africada postalveolar sonora	garej	S
ð	D	fricativa dental sonora	deuddeg	T
f	f	fricativa labiodental sorda	ffacs	f
g	g	oclusiva velar sonora	gadael	k
h	h	fricativa glotal sorda	haearn	k

IPA	X-SAMPA	Descripción	Ejemplo	Visema
j	j	aproximante palatal	astudio	i
k	k	oclusiva velar sorda	cant	k
l	l	aproximante alveolar lateral	lan	t
ɬ	K	fricativa lateral alveolar sorda	llan	t
m	m	nasal bilabial	mae	p
m̥	m_0	nasal bilabial sorda	ymhen	p
n	n	nasal alveolar	naw	t
n̥	n_0	nasal alveolar sorda	anhawster	t
ŋ	N	nasal velar	argyfwng	k
ŋ̊	N_0	nasal velar sorda	anghenion	k
p	p	oclusiva bilabial sorda	pump	p
r	r	vibrante alveolar	rhoi	r
r̥	r_0	vibrante alveolar sorda	garw	r
s	s	fricativa alveolar sorda	saith	s
ʃ	S	fricativa postalveolar sorda	siawns	S

IPA	X-SAMPA	Descripción	Ejemplo	Visema
t	t	oclusiva alveolar sorda	tegan	t
͡tʃ	tS	africada postalveolar sorda	cytsain	S
θ	T	fricativa dental sorda	aberth	T
v	v	fricativa labiodental sonora	prawf	f
w	w	aproximante velo-labial	rhagweld	u
χ	X	fricativa uvular sorda	chwech	k
z	z	fricativa alveolar sonora	aids	s
ʒ	Z	fricativa postalveolar sonora	rouge	S
Vocales				
ə	@	vocal media central	ychwanega	@
a	a	vocal abierta anterior no redondeada	acen	a
ai	ai	diptongo	dau	a
au	au	diptongo	awdur	a

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ɑː	A:	vocal larga abierta posterior no redondeada	mab	a
ɑːɨ	A:1	diptongo	aelod	a
eː	e:	vocal larga semicerrada anterior no redondeada	peth	e
ɛ	E	vocal semiabierta anterior no redondeada	pedwar	E
ɛi	Ei	diptongo	beic	E
iː	i:	vocal larga cerrada anterior no redondeada	tri	i
ɪ	I	vocal casi cerrada semianterior no redondeada	miliwn	i
ɨu	1u	diptongo	unigryw	i
oː	o:	vocal larga semicerrada posterior redondeada	oddi	o
ɔ	O	vocal semiabierta posterior redondeada	oddieithr	O
ɔi	Oi	diptongo	troi	O

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ɔu	Ou	diptongo	rownd	O
u:	u:	vocal larga cerrada posterior redondeada	cwch	u
ʊ	U	vocal casi cerrada semiposterior redondeada	acwstig	u
ʊi	Ui	diptongo	wyth	u
Símbolos adicionales				
ˈ	"	acento principal	Alabama	
ˌ	%	acento secundario	Alabama	
.	.	límite de sílaba	A.la.ba.ma	

Motores de voz de Amazon Polly

Amazon Polly tiene cuatro motores de voz que convierten el texto de entrada en voz realista. Son: Generativo, Forma larga, Neural y Estándar. Para usar una voz de Amazon Polly, seleccione un motor y una operación de la API de síntesis de voz. A continuación, proporcione el texto para que el motor lo sintetice y seleccione un formato de salida de audio. A partir de estos datos de entrada, Amazon Polly sintetiza el texto proporcionado en un flujo de audio de voz de gran calidad.

En las siguientes secciones se incluyen detalles sobre los motores de voz que ofrece Amazon Polly.

Temas

- Voces generativas
- Voces de formato largo
- Voces neuronales
- Voces estándar
- Elección de un motor de voz

Voces generativas

El motor generativo text-to-speech (TTS) de Amazon Polly ofrece las voces conversacionales más parecidas a las humanas, emocionalmente comprometidas y adaptables disponibles para su uso a través de la consola Amazon Polly.

El motor generativo es el modelo de Amazon Polly TTS más grande hasta la fecha. Implementa un transformador de mil millones de parámetros que convierte el texto sin procesar en códigos de voz, seguido de un descodificador basado en la convolución que convierte estos códigos de voz en formas de onda de forma gradual y reproducible. Este método muestra las habilidades emergentes de los modelos lingüísticos de gran tamaño (LLMs), que se conocen ampliamente, cuando se entrenan con volúmenes cada vez mayores de datos privados y disponibles al público que comprenden una variedad de voces, idiomas y estilos.

El motor generativo crea un discurso sintético que es emocionalmente expresivo, asertivo y muy coloquial, de una manera considerablemente similar a la voz humana. Puedes usar estas voces como un asistente de atención al cliente experto, un instructor virtual o un anunciante con un discurso sintético casi humano.

> **Note**
>
> La state-of-the-art tecnología en la que se basan estas voces se inscribe en el paradigma de la IA generativa para el modelado del lenguaje y la voz. Un efecto secundario de la tecnología es que cualquier actualización de los datos de entrenamiento y del modelo podría provocar ligeras variaciones en la forma en que suenan las voces, incluso en el caso de que la calidad general mejore con las actualizaciones del modelo. Esto podría repercutir en los casos de uso en los que distintas partes del contenido se sintetizan durante un período de tiempo prolongado (por ejemplo, una temporada de podcasts).

Voces generativas disponibles

Amazon Polly ofrece actualmente 20 voces en una variante generativa. Estas voces de formato largo también están disponibles en una variante NTTS conversacional.

	Idioma	Código de idioma	Nombre/ID	Gender
1	Inglés (Australia)	en-AU	Olivia	Mujer
2	Inglés (India)	en-IN	Kajal	Mujer
3	Inglés (Reino Unido)	en-GB	Amy	Mujer
4	English (EE. UU.)	en-US	Danielle	Mujer
5	English (EE. UU.)	en-US	Joanna	Mujer
6	English (EE. UU.)	en-US	Matthew	Hombre
7	English (EE. UU.)	en-US	Ruth	Mujer

	Idioma	Código de idioma	Nombre/ID	Gender
8	English (EE. UU.)	en-US	Stephen	Hombre
9	Inglés (sudafricano)	en-ZA	Ayanda	Mujer
10	Francés (Francia)	fr-FR	Léa	Mujer
11	Francés (Francia)	fr-FR	Rémi	Hombre
12	Español (España)	es-ES	Lucía	Mujer
13	Español (España)	es-ES	Sergio	Hombre
14	Español (México)	es-MX	Mía	Mujer
15	Español (México)	es-MX	Andrés	Hombre
16	Español (EE. UU.)	es-US	Lupe	Mujer
17	Español (EE. UU.)	es-US	Pedro	Hombre
18	Alemán (Alemania)	de-DE	Vicki	Mujer
19	Alemán (Alemania)	de-DE	Daniel	Hombre
20	Italiano (Italia)	it-IT	Bianca	Mujer

> **Note**
>
> El costo de las voces generativas se especifica en la página de información de precios de Amazon Polly.

Compatibilidad con características y regiones

Las voces de formato largo de Amazon Polly están disponibles en las siguientes regiones:

- Este de EE. UU. (Norte de Virginia): us-east-1
- Europa (Fráncfort) eu-central-1
- EE.UU. Oeste (Oregón): us-west-2
- No hay disponibles otras regiones

Se admiten las siguientes características con las voces generativas:

- Operaciones de síntesis de voz asíncronas y en tiempo real.
- El estilo Presentador no se admite en el motor generativo.
- La mayoría (pero no todas) de las etiquetas SSML son compatibles con Amazon Polly. Para obtener más información acerca de las etiquetas SSML compatibles con NTTS, consulte Etiquetas SSML admitidas
- Al igual que ocurre con las voces estándar, puede elegir entre varias frecuencias de muestreo para optimizar el ancho de banda y la calidad de audio de su aplicación. Las velocidades de muestreo válidas para las voces estándar y neuronales son 8 kHz, 16 kHz, 22 kHz o 24 kHz. El valor predeterminado para las voces estándar es 22 kHz. El valor predeterminado para las voces estándar es 24 kHz. Amazon Polly admite los formatos de MP3 transmisión de audio OGG (Vorbis) y PCM sin procesar.

En estos momentos, no está disponible la compatibilidad con la generación de marcas de voz.

> **Note**
>
> En el improbable caso de que se produzca una alucinación del modelo (y dado el modelo de comportamiento del motor generativo, que consiste en reproducir el discurso símbolo por

símbolo), se impone un mecanismo de parada de emergencia. El mecanismo incorporado impide que el modelo siga reproduciendo la voz. Esta característica de seguridad se basa en un análisis de datos en el que el modelo tiene el potencial de alucinar, por lo general, al final de la frase.
Puede haber casos en los que el modelo piense que va a alucinar y acabe cortando una palabra durante un paso de generación, por lo que traduce la mitad de la palabra. Esto podría generar resultados inapropiados.

Voces de formato largo

Amazon Polly tiene un motor de formato largo que produce voces parecidas a las humanas, altamente expresivas y emocionalmente expertas. Las voces de formato largo están diseñadas para captar la atención de los oyentes con contenido más extenso, como artículos de noticias, materiales de formación o vídeos de marketing.

Las voces de formato largo de Amazon Polly están desarrolladas con una tecnología TTS de aprendizaje profundo de última generación. El modelo aprende a reproducir los fonemas, la prosodia, la entonación y otros aspectos fonéticos y acústicos del lenguaje humano, lo que da como resultado una salida de voz muy natural.

El motor de formato largo utiliza incrustaciones de texto para interpretar el significado de un texto. Con el uso de incrustaciones de texto, el motor de formato largo puede generar el énfasis, las pausas y el tono correctos de una voz natural. El resultado es una voz que combina toda la gama de elementos emocionales presentes en la comunicación humana. Esto incluye imitar la sorpresa o diferenciar el diálogo de la narración. En conjunto, esto crea un producto de voz de primera calidad que suena como el de un ser humano vivo.

> **Note**
>
> La state-of-the-art tecnología en la que se basan estas voces se inscribe en el paradigma de la IA generativa para el modelado del lenguaje y la voz. Un efecto secundario de la tecnología es que cualquier actualización de los datos de entrenamiento y del modelo podría provocar ligeras variaciones en la forma en que suenan las voces, incluso en el caso de que la calidad general mejore con las actualizaciones del modelo. Esto podría repercutir en los casos de uso en los que distintas partes del contenido se sintetizan durante un período de tiempo prolongado (por ejemplo, una temporada de podcasts).

Voces de formato largo disponibles

Amazon Polly ofrece actualmente cuatro voces de formato largo en-US y dos es-ES. Ambos idiomas tienen voces femeninas y masculinas disponibles. Las voces largas en inglés de Daniel, Gregory y Ruth también están disponibles en una variante NTTS conversacional.

	Idioma	Código de idioma	Nombre/ID	Gender
1	English (EE. UU.)	en-US	Danielle Gregory Ruth Patricio	Mujer Hombre Mujer Hombre
2	Español (España)	es-ES	Alba Raúl	Mujer Hombre

Compatibilidad con características y regiones

Las voces de formato largo de Amazon Polly están disponibles en las siguientes regiones:

- Este de EE. UU. (Norte de Virginia): us-east-1
- Otras regiones no están disponibles

El motor de formato largo de Amazon Polly admite las siguientes características:

- Operaciones de síntesis de voz asíncronas y en tiempo real.
- Todas las marcas de voz.
- La mayoría (pero no todas) de las etiquetas SSML son compatibles con Amazon Polly. Para obtener más información acerca de las etiquetas SSML compatibles con NTTS, consulte Etiquetas SSML admitidas
- Al igual que ocurre con las voces estándar, puede elegir entre varias frecuencias de muestreo para optimizar el ancho de banda y la calidad de audio de su aplicación. Las velocidades de muestreo

válidas para las voces estándar, de formato largo y neuronales son: 8 kHz, 16 kHz, 22 kHz o 24 kHz. El valor predeterminado para las voces estándar es 22 kHz. El valor predeterminado para las voces neuronales y de formato largo es 24 kHz. Amazon Polly admite los formatos de MP3 transmisión de audio OGG (Vorbis) y PCM sin procesar.

> **Note**
>
> El costo de las voces de formato largo se especifica en la página de información de precios de Amazon Polly.

Voces neuronales

Amazon Polly tiene un motor neuronal text-to-speech (NTTS) que puede producir voces de una calidad aún mayor que las voces estándar. Las voces TTS estándar utilizan síntesis concatenativa. Este método concatena los fonemas de los fragmentos grabados, lo que produce un habla sintetizada con un sonido muy natural. Sin embargo, las variaciones inevitables del habla y las técnicas utilizadas para segmentar las formas de onda limitan la calidad del habla. El sistema NTTS de Amazon Polly no utiliza la síntesis de concatenación estándar para producir voz. Este sistema tiene dos componentes:

- Una red neuronal, que convierte una secuencia de fonemas (las unidades más básicas del lenguaje) en una secuencia de espectrogramas. (Los espectogramas son instantáneas de los niveles de energía en diferentes bandas de frecuencia).
- Un codificador de voz, que convierte los espectrogramas en una señal de audio casi continua.

El primer componente del sistema TTS neuronal es un modelo. sequence-to-sequence Este modelo no crea sus resultados únicamente a partir de la entrada correspondiente, sino que también considera el funcionamiento conjunto de la secuencia de los elementos de entrada. El modelo elige los espectrogramas que genera, de modo que sus bandas de frecuencia enfaticen las características acústicas que el cerebro humano utiliza al procesar el habla.

A continuación, la salida de este modelo se transfiere a un vocoder neuronal. Este convierte los espectrogramas en formas de onda de voz. Si se entrena en los grandes conjuntos de datos que se utilizan para crear sistemas de síntesis concatenativa de uso general, este sequence-to-sequence enfoque producirá voces de mayor calidad y con un sonido más natural.

Voces neuronales disponibles

Las voces neuronales están disponibles en 36 idiomas y variantes lingüísticas. En la siguiente tabla se enumeran las voces.

	Idioma y variantes lingüísticas	Código de idioma	Nombre/ID	Gender
1	Árabe (Golfo)	ar-AE	Hala Zayd	Mujer Hombre
2	Neerlandés belga (flamenco)	nl-BE	Lisa	Mujer
3	Catalán	ca-ES	Arlet	Mujer
4	Checo	cs-CZ	Jitka	Mujer
5	Chino (cantonés)	yue-CN	Hiujin	Mujer
6	Chino (mandarín)	cmn-CN	Zhiyu	Mujer
7	Danés	da-DK	Sofie	Mujer
8	Neerlandés	nl-NL	Laura	Mujer
9	Inglés (Australia)	en-AU	Olivia	Mujer
10	Inglés (Reino Unido)	en-GB	Amy* Emma Brian Arthur	Mujer Mujer Hombre Hombre
11	Inglés (India)	en-IN	Kajal	Mujer

	Idioma y variantes lingüísticas	Código de idioma	Nombre/ID	Gender
12	Inglés (irlandés)	en-IE	Niamh	Mujer
13	Inglés (Nueva Zelanda)	en-NZ	Aria	Mujer
14	Inglés (singapurense)	En-SG	Jazmín	Mujer
15	Inglés (sudafricano)	en-ZA	Ayanda	Mujer
16	English (EE. UU.)	en-US	Danielle Gregory Ivy Joanna* Kendra Kimberly Salli Joey Justin Kevin Matthew* Ruth Stephen	Mujer Hombre Mujer (niña) Mujer Mujer Mujer Mujer Hombre Hombre (niño) Hombre (niño) Hombre Mujer Hombre
17	Finés	fi-FI	Suvi	Mujer

	Idioma y variantes lingüísticas	Código de idioma	Nombre/ID	Gender
18	Francés (belga)	fr-BE	Isabelle	Mujer
19	Francés (Canadá)	fr-CA	Gabrielle Liam	Mujer Hombre
20	Francés	fr-FR	Léa Rémi	Mujer Hombre
21	Alemán	de-DE	Vicki Daniel	Mujer Hombre
22	Alemán (austriaco)	de-AT	Hannah	Mujer
23	Alemán (suizo)	de-CH	Sabrina	Mujer
24	Hindi	hi-IN	Kajal	Mujer
25	Italiano	it-IT	Bianca Adriano	Mujer Hombre
26	Japonés	ja-JP	Takumi Kazuha Tomoko	Hombre Mujer Mujer
27	Coreano	ko-KR	Seoyeon Jihye	Mujer Mujer
28	Noruego	nb-NO	Ida	Mujer
29	Polaco	pl-PL	Ola	Mujer

	Idioma y variantes lingüísticas	Código de idioma	Nombre/ID	Gender
30	Portugués (Brasil)	pt-BR	Camila Vitória/Vitoria Thiago	Mujer Mujer Hombre
31	Portugués (Europa)	pt-PT	Inês/Ines	Mujer
32	Español (España)	es-ES	Lucía Sergio	Mujer Hombre
33	Español (México)	es-MX	Mia Andrés	Mujer Hombre
34	Español (EE. UU.)	es-US	Lupe* Pedro	Mujer Hombre
35	Sueco	sv-SE	Elin	Mujer
36	Turco	tr-TR	Burcu	Mujer

*Las voces de Matthew y Joanna se pueden usar con los estilos de habla Newscaster y Conversacional. Para obtener más información, consulte Aplicación de la voz de presentador.

Compatibilidad con características y regiones

Las voces neuronales no están disponibles en todas AWS las regiones ni son compatibles con todas las funciones de Amazon Polly.

Las voces neuronales se admiten en las siguientes regiones:

- Este de EE. UU. (Norte de Virginia): us-east-1

- EE.UU. Oeste (Oregón): us-west-2
- África (Ciudad del Cabo) (af-south-1)
- Asia-Pacífico (Tokio) ap-northeast-1
- Asia-Pacífico (Seúl) ap-northeast-2
- Asia-Pacífico (Osaka): ap-northeast-3
- Asia-Pacífico (Bombay) ap-south-1
- Asia-Pacífico (Singapur) ap-southeast-1
- Asia Pacífico (Sídney): ap-southeast-2
- Asia Pacífico (Malasia): ap-southeast-5
- Canadá (centro) ca-central-1
- Europa (Fráncfort) eu-central-1
- Europa (Irlanda) (eu-west-1)
- Europa (Londres) eu-west-2
- Europa (París): eu-west-3
- Europa (España): eu-south-2
- AWS GovCloud (EEUU-Oeste): -1 us-gov-west

Los puntos de conexión y protocolos de estas regiones son idénticos a los que se utilizan para las voces estándar. Para obtener más información, consulte Cuotas y puntos de conexión de Amazon Polly.

Se admiten las siguientes características con las voces neuronales:

- Operaciones de síntesis de voz asíncronas y en tiempo real.
- Estilo de habla Newscaster. Para obtener más información sobre los estilos de habla, consulte Aplicación de la voz de presentador.
- Todas las marcas de voz.
- La mayoría de las etiquetas SSML son compatibles con Amazon Polly, pero no todas. Para obtener más información acerca de las etiquetas SSML compatibles con NTTS, consulte Etiquetas admitidas.

Al igual que ocurre con las voces estándar, puede elegir entre varias frecuencias de muestreo para optimizar el ancho de banda y la calidad de audio de su aplicación. Las velocidades de

muestreo válidas para las voces estándar y neuronales son 8 kHz, 16 kHz, 22 kHz o 24 kHz. El valor predeterminado para las voces estándar es 22 kHz. El valor predeterminado para las voces estándar es 24 kHz. Amazon Polly admite los formatos de MP3 transmisión de audio OGG (Vorbis) y PCM sin procesar.

Voces estándar

Amazon Polly tiene un motor estándar que utiliza síntesis de concatenación. Este método concatena los fonemas de los fragmentos grabados, lo que produce un habla sintetizada con un sonido muy natural.

Voces estándar disponibles

En estos momentos, Amazon Polly ofrece 40 voces estándar femeninas y 20 masculinas en 29 idiomas y variantes lingüísticas.

	Idioma	Código de idioma	Nombre/ID	Gender
1	Árabe	arb	Zeina	Mujer
2	Chino (mandarín)	cmn-CN	Zhiyu	Mujer
3	Danés	da-DK	Naja Mads	Mujer Hombre
4	Neerlandés	nl-NL	Lotte Ruben	Mujer Hombre
5	Inglés (Australia)	en-AU	Nicole Russell	Mujer Hombre
6	Inglés (Reino Unido)	en-GB	Amy Emma	Mujer Mujer

	Idioma	Código de idioma	Nombre/ID	Gender
			Brian	Hombre
7	Inglés (India)	en-IN	Aditi	Mujer
			Raveena	Mujer
8	English (EE. UU.)	en-US	Ivy	Mujer
			Joanna	Mujer
			Kendra	Mujer
			Kimberly	Mujer
			Salli	Mujer
			Joey	Hombre
			Kevin	Hombre
9	Inglés (Gales)	en-GB-WLS	Geraint	Hombre
10	Francés	fr-FR	Céline/Celine	Mujer
			Léa	Mujer
			Mathieu	Hombre
11	Francés (Canadá)	fr-CA	Chantal	Mujer
12	Alemán	de-DE	Marlene	Mujer
			Vicki	Mujer
			Hans	Hombre
13	Hindi	hi-IN	Aditi	Mujer

	Idioma	Código de idioma	Nombre/ID	Gender
14	Islandés	is-IS	Dóra/Dora	Mujer
			Karl	Hombre
15	Italiano	it-IT	Carla	Mujer
			Bianca	Mujer
			Giorgio	Hombre
16	Japonés	ja-JP	Mizuki	Mujer
			Takumi	Hombre
17	Coreano	ko-KR	Seoyeon	Mujer
18	Noruego	nb-NO	Liv	Mujer
19	Polaco	pl-PL	Ewa	Mujer
			Maja	Mujer
			Jacek	Hombre
			Ene	Hombre
20	Portugués (Brasil)	pt-BR	Camila	Mujer
			Vitória/Vitoria	Mujer
			Ricardo	Hombre
21	Portugués (Europa)	pt-PT	Inês/Ines	Mujer
			Cristiano	Hombre
22	Rumano	ro-RO	Carmen	Mujer

	Idioma	Código de idioma	Nombre/ID	Gender
23	Ruso	ru-RU	Tatyana Maxim	Mujer Hombre
24	Español (España)	es-ES	Conchita Lucía Enrique	Mujer Mujer Hombre
25	Español (México)	es-MX	Mia	Mujer
26	Español (EE. UU.)	es-US	Lupe Penélope/ Penelope Miguel	Mujer Mujer Hombre
27	Sueco	sv-SE	Astrid	Mujer
28	Turco	tr-TR	Filiz	Hombre
29	Galés	cy-GB	Gwyneth	Mujer

Compatibilidad con características y regiones

Las voces estándar de Amazon Polly están disponibles en las siguientes regiones de Amazon Polly:

- Este de EE. UU. (Norte de Virginia): us-east-1
- Este de EE. UU. (Ohio): us-east-2
- Oeste de EE. UU. (Norte de California): us-west-1
- EE.UU. Oeste (Oregón): us-west-2
- África (Ciudad del Cabo) (af-south-1)
- Asia-Pacífico (Hong Kong): ap-east-1

- Asia-Pacífico (Tokio) ap-northeast-1
- Asia-Pacífico (Seúl) ap-northeast-2
- Asia-Pacífico (Osaka): ap-northeast-3
- Asia-Pacífico (Bombay) ap-south-1
- Asia-Pacífico (Singapur) ap-southeast-1
- Asia Pacífico (Sídney): ap-southeast-2
- Asia Pacífico (Malasia): ap-southeast-5
- China (Ningxia): cn-northwest-1
- Canadá (centro) ca-central-1
- Europa (Fráncfort) eu-central-1
- Europa (Irlanda) (eu-west-1)
- Europa (Londres) eu-west-2
- Europa (París): eu-west-3
- Europa (España): eu-south-2
- Europa (Estocolmo): eu-north-1
- Medio Oriente (Baréin): me-south-1
- América del Sur (São Paulo): sa-east-1
- AWS GovCloud (EEUU-Oeste): -1 us-gov-west

Los puntos de conexión y los protocolos de estas regiones son idénticos a los que se utilizan para las voces neuronales. Para obtener más información, consulte Cuotas y puntos de conexión de Amazon Polly.

El motor estándar de Amazon Polly admite las siguientes características (por determinar):

- Operaciones de síntesis de voz asíncronas y en tiempo real.
- Todas las marcas de voz.
- La mayoría (pero no todas) de las etiquetas SSML son compatibles con Amazon Polly. Para obtener más información acerca de las etiquetas SSML compatibles con NTTS, consulte Etiquetas SSML admitidas.
- Puede elegir entre varias frecuencias de muestreo para optimizar el ancho de banda y la calidad de audio de la aplicación. Las frecuencias de muestreo predeterminadas para las voces estándar

son de 22 kHz. Amazon Polly admite los formatos de MP3 transmisión de audio OGG (Vorbis) y PCM sin procesar.

> **Note**
>
> El costo de las voces estándar se especifica en la página de información de precios de Amazon Polly.

Elección de un motor de voz

Puede acceder a las voces de Amazon Polly a través de la consola Amazon Polly o. AWS CLI

Elección de un motor de voz en la consola

1. Abre la consola Amazon Polly en. https://console.aws.amazon.com/polly/
2. Desde la consola de Amazon Polly, elija el motor de voz que desee.
3. Elija la voz que desee en el menú desplegable de voz.
4. Genere el audio de TTS con el texto que desee.

Para elegir un motor de voz en las operaciones de API AWS CLI, especifique las operaciones `Engine` y `VoiceId` en la `SyntheszieSpeech` `StartSpeechSynthesisTask` API. Para ver algunos ejemplos, consulte las muestras de código de inicio rápido y los ejemplos de Python.

Marcas de voz

Las marcas de voz son metadatos que describen el fragmento que se sintetiza; por ejemplo, dónde empieza y termina una palabra o una frase en la secuencia de audio. Si solicita las marcas de voz de un texto, Amazon Polly devolverá estos metadatos en lugar del segmento de habla sintetizado. Si utiliza las marcas de voz con la secuencia de audio del fragmento hablado, podrá proporcionar a las aplicaciones una experiencia visual mejorada.

Por ejemplo, si combina los metadatos con la secuencia de audio del texto, podrá sincronizar el habla con animación facial (sincronización de los labios) o resaltar las palabras escritas a medida que se pronuncian.

Las marcas de voz están disponibles cuando se utilizan motores neuronales, de formato largo o estándar text-to-speech.

Temas

- Tipos de marcas de voz
- Visemas y Amazon Polly
- Salida de marcas de voz
- Solicitud de marcas de voz
- Ejemplo de marcas de voz sin SSML
- Ejemplo de marcas de voz con SSML

Tipos de marcas de voz

Las marcas de voz se solicitan mediante la SpeechMarkTypesopción de los comandos SynthesizeSpeecho StartSpeechSynthesisTask. Debe especificar los metadatos del texto de entrada que quiere que se devuelvan. Puede solicitar hasta cuatro tipos de metadatos y debe especificar al menos uno en cada solicitud. Esta solicitud no genera ninguna salida de audio.

En AWS CLI, por ejemplo:

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

Amazon Polly genera las marcas de voz utilizando los siguientes elementos:

- sentence: sirve para señalar frases en el texto de entrada.
- word: sirve para señalar palabras en el texto.
- viseme: describe los movimientos de la cara y la boca que corresponden a cada fonema que se pronuncia. Para obtener más información, consulte Visemas y Amazon Polly.
- ssml describe un elemento <mark> del texto de entrada SSML. Para obtener más información, consulte Generación de fragmentos hablados desde documentos SSML.

Visemas y Amazon Polly

Un visema representa la posición de la cara y la boca al hablar. Es el equivalente visual de un fonema, que es la unidad acústica básica que forma una palabra. Los visemas son los componentes visuales básicos del habla.

Cada idioma tiene un conjunto de visemas que se corresponden con los propios fonemas. En un idioma, a cada fonema le corresponde un visema, que representa la forma de la boca al emitir el sonido. Sin embargo, no todos los visemas pueden asignarse a un único fonema, ya que muchos fonemas se parecen al hablar, incluso aunque el sonido sea diferente. Por ejemplo, en inglés, las palabras "pet" y "bet" son acústicamente diferentes. Sin embargo, cuando se observan visualmente (sin sonido), parecen exactamente iguales.

En la tabla siguiente se muestra una lista parcial de fonemas del Alfabeto Fonético Internacional (IPA, por sus siglas en inglés) y símbolos del Alfabeto Fonético Extendido SAM (X-SAMPA, por sus siglas en inglés), así como sus correspondientes visemas para las voces de inglés de Estados Unidos.

Para ver la tabla completa y las tablas de todos los idiomas disponibles, consulte Idiomas en Amazon Polly.

IPA	X-SAMPA	Descripción	Ejemplo	Visema
Consonantes				
b	b	Oclusiva bilabial sonora	bed	p
d	d	Oclusiva alveolar sonora	dig	t

IPA	X-SAMPA	Descripción	Ejemplo	Visema
ʤ	dZ	Africada postalveolar sonora	jump	S
ð	D	Fricativa dental sonora	then	T
f	f	Fricativa labiodental sorda	five	f
g	g	Oclusiva velar sonora	game	k
h	h	Fricativa glotal sorda	house	k
...	...	...	...	...

Salida de marcas de voz

Amazon Polly devuelve los objetos de marcas de voz en una secuencia JSON delimitada por saltos de línea. Los objetos de marca de voz tienen los siguientes campos:

- time: marca temporal en milisegundos que indica el comienzo de la secuencia de audio correspondiente.
- type: tipo de marca de voz (frase, palabra, visema o ssml)
- start: desplazamiento en bytes (no caracteres) del inicio del objeto en el texto de entrada (sin incluir marcas de visemas)
- end: desplazamiento en bytes (no caracteres) del final del objeto en el texto de entrada (sin incluir marcas de visemas)
- value: varía en función del tipo de marca de voz.
  - SSML: etiqueta SSML <mark>.
  - viseme: nombre del visema
  - word o sentence: subcadena del texto de entrada delimitada por los campos start y end.

Por ejemplo, Amazon Polly genera el objeto de marca de voz `word` siguiente a partir del texto "Mary had a little lamb":

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

La palabra descrita ("had") comienza 373 milisegundos después de que comience la secuencia de audio; además, se inicia en el byte 5 y termina en el byte 8 del texto de entrada.

> **Note**
>
> Estos metadatos corresponden al ID de voz `Joanna`. Si utiliza otra voz con el mismo texto de entrada, los metadatos pueden variar.

Solicitud de marcas de voz

Puede utilizar la consola o el comando `synthesize-speech` para solicitar las marcas de voz de Amazon Polly. Una vez hecho esto, podrá ver los metadatos o guardarlos en un archivo.

Console

Generación de marcas de voz en la consola

1. Inicia sesión en la consola Amazon Polly AWS Management Console y ábrela en. https://console.aws.amazon.com/polly/
2. Elija la pestaña Text-to-Speech.
3. Active SSML para usar SSML.
4. Escriba o pegue el texto en el cuadro de entrada.
5. En Idioma, elija el idioma del texto.
6. En Voz, elija la voz que desee usar.
7. Para cambiar la pronunciación del texto, expanda Configuración adicional, active Personalizar pronunciación y, en Aplicar lexicón, elija el lexicón deseado.
8. Para verificar la voz, elija Escuchar.
9. Active Configuración del formato del archivo de locución.

> **Note**
>
> Los formatos de descarga MP3, OGG o PCM no generarán marcas de voz.

10. En File Format, elija Speech Marks.
11. En Speech Mark Types, elija los tipos de marcas de voz que desea generar. La opción para elegir metadatos SSML solo está disponible cuando SSML está activado. Para obtener más información sobre el uso de SSML con Amazon Polly, consulte Generación de fragmentos hablados desde documentos SSML.
12. Elija Descargar.

AWS CLI

Además del texto de entrada, son necesarios los siguientes elementos para devolver estos metadatos:

- `output-format`

  Amazon Polly solo es compatible con el formato JSON cuando se devuelven marcas de voz.

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

  Si utiliza un formato de salida no compatible, Amazon Polly genera una excepción.

- `voice-id`

  Para asegurarse de que los metadatos coinciden con la secuencia de audio asociada, especifique la misma voz que se utilizó para generar la secuencia de audio del fragmento sintetizado. La velocidad de habla no es la misma en todas las voces. Si utiliza una voz diferente que la que se empleó para generar el fragmento hablado, los metadatos no coincidirán con los de la secuencia de audio.

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

- `speech-mark-types`

  Especifique el tipo o los tipos de marcas de voz que desee. Puede solicitar algunos o todos los tipos de marcas de voz, pero debe especificar al menos un tipo.

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

- `text-type`

  El texto sin formato es el texto de entrada predeterminado en Amazon Polly, por tanto, debe usar `text-type ssml` si desea obtener marcas de voz SSML.

- `outfile`

  Especifique el archivo de salida en el que se escribirán los metadatos.

  ```
  MaryLamb.txt
  ```

El siguiente AWS CLI ejemplo está formateado para Unix, Linux y macOS. En Windows, sustituya la barra invertida (\) del carácter de continuación de Unix al final de cada línea por un signo de intercalación (^) y utilice comillas completas (") alrededor del texto introducido con comillas simples (') para las etiquetas interiores.

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

Ejemplo de marcas de voz sin SSML

En el ejemplo siguiente se muestra el aspecto aproximado que tendrán en su pantalla los metadatos de la frase "Mary had a little lamb" solicitados. Por simplificar, no se han incluido las marcas de voz SSML en el ejemplo.

El siguiente AWS CLI ejemplo está formateado para Unix, Linux y macOS. En Windows, sustituya la barra invertida (\) del carácter de continuación de Unix al final de cada línea por un signo de intercalación (^) y utilice comillas completas (") alrededor del texto introducido con comillas simples (') para las etiquetas interiores.

```
aws polly synthesize-speech \
  --output-format json \
```

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

Si realiza esta solicitud, Amazon Polly devuelve lo siguiente en el archivo .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"}
```

En este resultado, cada parte del texto está dividido en función de las marcas de voz:

- Por un lado, la frase "Mary had a little lamb".
- Por otro lado, cada palabra del texto: "Mary", "had", "a", "little" y "lamb".
- Por último, el visema de cada sonido de la secuencia de audio correspondiente: "p", "E", "r", "i", etc. Para obtener más información sobre los visemas, consulte Visemas y Amazon Polly.

Ejemplo de marcas de voz con SSML

El proceso de generar marcas de voz desde un texto mejorado con SSML es muy parecido al proceso que se utiliza cuando SSML no está presente. Utilice el comando `synthesize-speech`,

especifique el texto mejorado con SSML y determine el tipo de marcas de voz que desea usar, tal y como se muestra en el ejemplo siguiente. Para que el ejemplo resulte más fácil de leer, no hemos incluido las marcas de voz de los visemas, pero también podrían incluirse.

El siguiente AWS CLI ejemplo está formateado para Unix, Linux y macOS. En Windows, sustituya la barra invertida (\) del carácter de continuación de Unix al final de cada línea por un signo de intercalación (^) y utilice comillas completas (") alrededor del texto introducido con comillas simples (') para las etiquetas interiores.

```
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
```

Si realiza esta solicitud, Amazon Polly devuelve lo siguiente en el archivo .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"}
```

Generación de fragmentos hablados desde documentos SSML

Puede utilizar Amazon Polly para generar fragmentos hablados a partir de una entrada de texto sin formato o de documentos etiquetados con idioma de marcado de síntesis de voz (SSML). El uso de texto mejorado con SSML ofrece mayor control sobre el modo en que Amazon Polly genera el fragmento hablado a partir del texto proporcionado.

Con las etiquetas SSML, puede personalizar y controlar diferentes aspectos del fragmento hablado, como la pronunciación, el volumen y la velocidad de habla. En el AWS Management Console, el texto mejorado con SSML que desea convertir a audio se introduce en la pestaña SSML de la página. Text-to-Speech Aunque el texto que se escribe sin formato se basa en los valores predeterminados elegidos para el idioma y la voz, el texto mejorado con SSML indica a Amazon Polly no solo lo que se desea decir, sino también cómo se desea decirlo. Si se exceptúan las etiquetas SSML añadidas, Amazon Polly sintetiza el texto mejorado con SSML del mismo modo que sintetiza el texto sin formato. Para obtener más información, consulte Ejemplo de síntesis de voz con Amazon Polly.

Cuando utilice SSML, incluya todo el texto en una etiqueta `<speak>` para indicar a Amazon Polly que está utilizando SSML. Por ejemplo:

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

A continuación, utilice etiquetas SSML específicas en el texto que está dentro de las etiquetas `<speak>` para personalizar la forma en la que desea que suene el texto. Puede añadir una pausa, cambiar el ritmo de la locución, reducir o aumentar el volumen de la voz, o añadir muchas otras personalizaciones para hacer modo que el texto suene como usted desea. Para obtener una lista completa de las etiquetas SSML que puede utilizar, consulte Etiquetas SSML admitidas.

Por ejemplo, puede incluir una larga pausa dentro del texto o cambiar la velocidad de habla o el tono. Otras opciones incluyen:

- enfatizar palabras o frases específicas
- usar la pronunciación fonética
- incluir sonidos de respiración
- susurrar

- usar el estilo de habla Newscaster.

Para obtener información completa sobre las etiquetas SSML admitidas por Amazon Polly y cómo utilizarlas, consulte Etiquetas SSML admitidas

Cuando se utiliza SSML, existen varios caracteres reservados que requieren un tratamiento especial. Esto se debe a que SSML utiliza estos caracteres como parte de su código. Para utilizarlos, debe usar una entidad específica para aplicarles caracteres de escape. Para obtener más información, consulte Caracteres reservados en SSML

Amazon Polly proporciona estos tipos de control con un subconjunto de etiquetas de marcado SSML, tal y como se especifica en la recomendación de W3C sobre idioma de marcado de síntesis de voz (SSML) versión 1.1.

Puede utilizar SSML en la consola de Amazon Polly o mediante la AWS CLI. En los siguientes temas se indica cómo puede utilizar SSML para generar fragmentos hablados y controlar la salida con precisión para que se adapte a sus necesidades.

Temas

- Caracteres reservados en SSML
- Uso de SSML en la consola
- Uso de SSML con el comando de síntesis de voz
- Síntesis de un documento mejorado con SSML
- Etiquetas SSML admitidas

Caracteres reservados en SSML

Existen cinco caracteres predefinidos que no se pueden utilizar normalmente dentro de una instrucción SSML. Estas entidades están reservadas para la especificación del idioma. Estos caracteres son los siguientes:

Carácter	Código de escape	Nombre
"	"	comillas (comillas dobles)
&	&	ampersand
'	'	apóstrofo o comillas simples
<	<	signo menor que
>	>	signo mayor que

Como SSML utiliza estos caracteres como parte de su código, para usar estos símbolos en SSML, debe incluir el carácter en una secuencia de escape cuando lo utilice. Puede utilizar el código de escape en lugar del carácter real para que se muestre correctamente mientras crea un documento SSML válido. Por ejemplo, la frase siguiente:

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

se representaría en SSML como

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

En este caso, se aplica escape a los caracteres especiales del apóstrofo y el signo ampersand para que el documento SSML siga siendo válido.

Para los símbolos &, <, y > los códigos de escape siempre son necesarios cuando se utiliza SSML. Además, cuando utilice el apóstrofo/comillas simples (') como apóstrofo, también debe usar el código de escape.

Sin embargo, cuando se utiliza las comillas dobles (") o el apóstrofo/comillas simples (') como comillas, el uso del código de escape dependerá del contexto.

Comillas dobles

- Debe aplicarse escape cuando se encuentre en un valor de atributo delimitado por comillas dobles. Por ejemplo, en el código siguiente AWS CLI

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

- No es necesario aplicar escape cuando se encuentre en contexto textual. Por ejemplo, en la siguiente frase

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

- No es necesario aplicar escape cuando está en un valor de atributo delimitado por comillas simples. Por ejemplo, en el código AWS CLI siguiente

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

Comillas simples

- Se debe aplicar escape cuando se utiliza como apóstrofo. Por ejemplo, en la siguiente frase

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

- No es necesario aplicar escape cuando se encuentre en contexto textual. Por ejemplo, en la siguiente frase

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

- No es necesario aplicar escape cuando está en un atributo de código delimitado por comillas dobles. Por ejemplo, en el código AWS CLI siguiente

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

Uso de SSML en la consola

En el ejemplo siguiente, se usa una etiqueta SSML para indicar a Amazon Polly que sustituya "World Wide Web Consortium" por "W3C" cuando lea un breve párrafo. También puede utilizar etiquetas para introducir una pausa y susurrar una palabra. Compare los resultados de este ejercicio con los de Aplicación de lexicones (síntesis del habla).

Para obtener más información y ejemplos de SSML, consulte Etiquetas SSML admitidas.

Para sintetizar la voz de texto mejorado con SSML (consola)

1. Inicia sesión en la consola Amazon Polly AWS Management Console y ábrela en. https://console.aws.amazon.com/polly/
2. Si aún no está abierta, elija la pestaña Text-to-Speech (Texto a voz).
3. Active SSML.
4. Escriba o pegue el siguiente texto en el cuadro de texto:

```
<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>
```

   Las etiquetas SSML indican a Amazon Polly cómo representar el texto:

   - `<break time="1s"/>` indica a Amazon Polly que debe detenerse un segundo entre las dos primeras frases.
   - `<sub alias="World Wide Web Consortium">W3C</sub>` indica a Amazon Polly que debe sustituir "World Wide Web Consortium" por el acrónimo "W3C".
   - `<amazon:effect name="whispered">Nice job</amazon:effect>` indica a Amazon Polly que susurre la segunda instancia de "Nice job".

> **Note**
>
> Al utilizar el AWS CLI, se escribe el texto introducido entre comillas para diferenciarlo del código que lo rodea. La consola de Amazon Polly no muestra el código, de modo que no incluya el texto de entrada entre comillas cuando la utilice.

5. En Idioma, seleccione Inglés (EE. UU.) y, a continuación, seleccione una voz.
6. Si desea escuchar el fragmento hablado, elija Escuchar.
7. Para guardar el archivo de voz, elija Descargar. Si quiere guardarlo en un formato diferente, expanda Configuración adicional, active la Configuración del formato del archivo de locución, elija el formato que desee y, a continuación, seleccione Descargar.

Uso de SSML con el comando de síntesis de voz

Este ejemplo muestra cómo utilizar el comando `synthesize-speech` con una cadena SSML. Cuando se utiliza el comando `synthesize-speech`, normalmente se proporciona lo siguiente:

- El texto de entrada (obligatorio)
- Las etiquetas de apertura y cierre (obligatorio)
- El formato de salida
- Una voz

En este ejemplo, especifica una cadena de texto sencilla entre comillas junto con las etiquetas `<speak></speak>` de apertura y cierre necesarias.

> **Important**
>
> Aunque no se utilizan comillas alrededor del texto de entrada en la consola de Amazon Polly, debe usarlas en la AWS CLI . También es importante que diferencie entre las comillas alrededor de texto de entrada y las comillas necesarias para las etiquetas individuales. Por ejemplo, puede usar comillas normales (") para rodear el texto de entrada y comillas simples (') para las etiquetas internas, o viceversa. Cualquiera de las dos opciones funciona para Unix, Linux y macOS. Sin embargo, con Windows debe incluir el texto de entrada entre comillas normales y usar las comillas simples para las etiquetas.

Para todos los sistemas operativos, puede usar comillas normales (") para rodear el texto de entrada y comillas simples (') para las etiquetas internas. Por ejemplo:

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

Para Unix, Linux y macOS, también puede usarlas al revés: las comillas simples (') para el texto de entrada y las comillas normales (") para las etiquetas internas:

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

El siguiente AWS CLI ejemplo está formateado para Unix, Linux y macOS. En Windows, sustituya la barra invertida (\) del carácter de continuación de Unix al final de cada línea por un signo de intercalación (^) y utilice comillas completas (") alrededor del texto introducido con comillas simples (') para las etiquetas interiores.

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

Para oír el fragmento hablado sintetizado, reproduzca el archivo `speech.mp3` resultante con un reproductor de sonido.

Síntesis de un documento mejorado con SSML

Para el texto de entrada largo, es posible que le resulte más fácil guardar el contenido SSML en un archivo y especificar simplemente el nombre de archivo en el comando `synthesize-speech`. Por ejemplo, puede guardar lo siguiente en un archivo llamado `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>
```

El atributo `xml:lang` establece `en-US` (inglés de EE. UU.) como el idioma del texto de entrada. Si desea obtener más información acerca de cómo afecta el idioma del texto de entrada y de la voz elegido al funcionamiento de `SynthesizeSpeech`, consulte Especificación de otro idioma para palabras específicas.

Para ejecutar un archivo mejorado con SSML

1. Guarde el SSML en un archivo (por ejemplo, `example.xml`).
2. Ejecute el siguiente comando `synthesize-speech` desde la ruta en la que guardó el archivo XML y especifique el archivo SSML como entrada sustituyendo `file:\\example.xml` para el texto de entrada. Como este comando apunta a un archivo en lugar de contener el texto de entrada real, no es necesario usar comillas.

   > **Note**
   >
   > El siguiente AWS CLI ejemplo está formateado para Unix, Linux y macOS. Para Windows, sustituya la barra diagonal invertida (\) utilizada como carácter de continuación de Unix al final de cada línea por el signo de intercalación (^).

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

3. Para oír el fragmento hablado sintetizado, reproduzca el archivo `speech.mp3` resultante con un reproductor de sonido.

Etiquetas SSML admitidas

Todas las etiquetas, excepto esta, `<amazon:domain name="news">` son compatibles con las voces estándar. La disponibilidad de etiquetas para otras voces se indica en la siguiente tabla.

Amazon Polly es compatible con las siguientes etiquetas SSML:

Acción	Etiqueta SSML	Disponibilidad de voz neuronal	Disponibilidad de voz de formato largo	Disponibilidad generativa de voz
Agregación de una pausa	<break>	Disponibilidad completa	Disponibilidad completa	Disponibilidad completa
Énfasis de palabras	<emphasis>	No disponible	No disponible	No disponible
Especificación de otro idioma para palabras específicas	<lang>	Disponibilidad completa	Disponibilidad completa	Disponibilidad completa
Colocación de una etiqueta personalizada en el texto	<mark>	Disponibilidad completa	Disponibilidad completa	Disponibilidad parcial
Agregación de una pausa entre párrafos	<p>	Disponibilidad completa	Disponibilidad completa	Disponibilidad completa
Uso de la pronunciación fonética	<phoneme>	Disponibilidad completa	Disponibilidad completa	Disponibilidad parcial
Control del volumen, velocidad de habla y tono	<prosody>	Disponibilidad parcial	Disponibilidad parcial	Disponibilidad parcial
Establecimiento de una duración máxima para voz sintetizada	<prosody amazon:max-duration>	No disponible	No disponible	No disponible

Acción	Etiqueta SSML	Disponibilidad de voz neuronal	Disponibilidad de voz de formato largo	Disponibilidad generativa de voz
Agregación de una pausa entre frases	<s>	Disponibilidad completa	Disponibilidad completa	Disponibilidad completa
Control de cómo se leen los tipos especiales de palabras	<say-as>	Disponibilidad parcial	Disponibilidad completa	Disponibilidad completa
Identificación de texto mejorado con SSML	<speak>	Disponibilidad completa	Disponibilidad completa	Disponibilidad completa
Pronunciación de acrónimos y abreviaturas	<sub>	Disponibilidad completa	Disponibilidad completa	Disponibilidad completa
Mejora de la pronunciación especificando partes del discurso	<w>	Disponibilidad completa	Disponibilidad completa	Disponibilidad completa
Adición de sonido de respiración	<amazon:auto-breaths>	No disponible	No disponible	No disponible
Estilo de habla Newscaster	<amazon: domain name =" news">	Solo en algunas voces neuronales	No disponible	No disponible
Adición de compresión de rango dinámico	<amazon:effect name="drc">	Disponibilidad completa	Disponibilidad completa	No disponible

Acción	Etiqueta SSML	Disponibilidad de voz neuronal	Disponibilidad de voz de formato largo	Disponibilidad generativa de voz
Habla de forma suave	<amazon:effect phonation="soft">	No disponible	No disponible	No disponible
Control del timbre	<amazon:effect > vocal-tract-length	No disponible	No disponible	No disponible
Susurros	<amazon:effect name="whispered">	No disponible	No disponible	No disponible

> **Note**
>
> Si utiliza etiquetas SSML no admitidas en formato largo, neuronal o estándar, aparecerá un error.

Identificación de texto mejorado con SSML

<speak>

Esta etiqueta es compatible con los formatos TTS generativo, de formato largo, neuronal y estándar.

La etiqueta `<speak>` es el elemento raíz de todo el texto SSML de Amazon Polly. Todo el texto mejorado con SSML debe incluirse entre un par de etiquetas <speak>.

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

Agregación de una pausa

<break>

Esta etiqueta es compatible con los formatos TTS generativo, de formato largo, neuronal y estándar.

Para añadir una pausa al texto, utilice la etiqueta <break>. Puede establecer una pausa basada en la intensidad (equivalente a la pausa después de una coma, una frase o un párrafo), o bien puede establecerla en un periodo de tiempo específico en segundos o milisegundos. Si no se especifica un atributo para determinar la duración de la pausa, Amazon Polly utiliza el valor predeterminado `<break strength="medium"/>`, que añade una pausa con la duración de una pausa después de una coma.

Valores del atributo `strength`:

- `none`: sin pausa. Use `none` para eliminar una pausa que se produce normalmente, como después de un punto.
- `x-weak`: tiene la misma fuerza que `none`, sin pausa.
- `weak`: establece una pausa de la misma duración que la pausa después de una coma.
- `medium`: tiene la misma fuerza que `weak`.
- `strong`: establece una pausa de la misma duración que la pausa después de una frase.
- `x-strong`: establece una pausa de la misma duración que la pausa después de un párrafo.

Valores del atributo `time`:

- *`[number]`*`s`: la duración de la pausa en segundos. La duración máxima es `10s`.
- *`[number]`*`ms`: la duración de la pausa en milisegundos. La duración máxima es `10000ms`.

Por ejemplo:

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

Si no utiliza un atributo con la etiqueta `break`, el resultado varia en función del texto:

- Si no hay ningún otro tipo de puntuación junto a la etiqueta `break`, se crea una de tipo `<break strength="medium"/>` (pausa después de una coma).
- Si la etiqueta está junto a una coma, se actualiza la etiqueta a `<break strength="strong"/>` (pausa después de una frase).
- Si la etiqueta está junto a un punto, se actualiza la etiqueta a `<break strength="x-strong"/>` (pausa después de un párrafo).

Énfasis de palabras

<emphasis>

Esta etiqueta solo es compatible con el formato TTS estándar.

Para enfatizar palabras, utilice la etiqueta <emphasis>. Enfatizar las palabras cambia el volumen y la velocidad de la voz. Más énfasis significa que Amazon Polly lee el texto más alto y lento. Menos énfasis hace que el texto se lea más bajo y rápido. Para especificar el grado de énfasis, utilice el atributo `level`.

Valores del atributo `level`:

- `Strong`: aumenta el volumen y ralentiza la velocidad de habla, para que la voz sea más alta y lenta.
- `Moderate`: aumenta el volumen y ralentiza la velocidad de habla, pero menos que `strong`. `Moderate` es el valor predeterminado.
- `Reduced`: aumenta el volumen e incrementa la velocidad de habla. La voz es más baja y rápida.

> **Note**
>
> La velocidad y el volumen de habla normales para una voz se encuentran entre los niveles `moderate` y `reduced`.

Por ejemplo:

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

Especificación de otro idioma para palabras específicas

<lang>

Esta etiqueta es compatible con los formatos TTS generativo, de formato largo, neuronal y estándar.

Especifique otro idioma para una palabra específica, frase u oración con la etiqueta <lang>. Las palabras y frases extranjeras normalmente se leen mejor cuando se incluyen entre un par de

etiquetas <lang>. Para especificar el idioma, utilice el atributo xml:lang. Para ver una lista completa de los idiomas disponibles, consulte Idiomas en Amazon Polly.

A menos que aplique la etiqueta <lang>, todas las palabras del texto de entrada se leen en el idioma de la voz especificada en voice-id. Si aplica la etiqueta <lang>, las palabras se leen en ese idioma.

Por ejemplo, si el voice-id es Joanna (que habla inglés de Estados Unidos), Amazon Polly lee lo siguiente en la voz de Joanna sin acento francés:

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

Si utiliza la voz de Joanna con la etiqueta <lang>, Amazon Polly lee la frase en la voz de Joanna en francés con acento americano:

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

Como Joanna no es una hablante nativa del francés, la pronunciación tendrá como base su idioma nativo, inglés de Estados Unidos. Por ejemplo, aunque una pronunciación francesa perfecta pronunciaría la palabra français con una /R/ vibrante uvular, la voz de Joanna, cuyo idioma nativo es el inglés americano, pronuncia este fonema con el sonido /r/ correspondiente.

Si utiliza el voice-id de Giorgio, que habla italiano, con el siguiente texto, Amazon Polly lee la frase en la voz de Giorgio con pronunciación italiana:

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

Si utiliza la misma voz con la siguiente etiqueta <lang>, Amazon Polly pronuncia Bruce Springsteen en inglés con acento italiano:

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

Esta etiqueta también se puede utilizar como sustituto de la DefaultLangCodeopción opcional al sintetizar voz. Sin embargo, esto requiere dar formato al texto con SSML.

Colocación de una etiqueta personalizada en el texto

<mark>

Esta etiqueta es compatible con los formatos TTS de formato largo, neuronal y estándar. Esta etiqueta no sirve para las voces generativas porque las marcas de voz no están disponibles para las voces generativas.

Para colocar una etiqueta personalizada dentro del texto, utilice la etiqueta **<mark>**. Amazon Polly no realiza ninguna acción en la etiqueta, pero devuelve la ubicación de la etiqueta en los metadatos SSML. Esta etiqueta puede ser cualquier cosa que se le ocurra, siempre que mantenga el siguiente formato:

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

Por ejemplo, suponga que el nombre de la etiqueta es "animal" y el texto de entrada es:

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

Amazon Polly podría devolver los siguientes metadatos SSML:

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

Agregación de una pausa entre párrafos

<p>

Esta etiqueta es compatible con los formatos TTS generativo, de formato largo, neuronal y estándar.

Para añadir una pausa entre los párrafos del texto, utilice la etiqueta <p>. El uso de esta etiqueta proporciona una pausa que incluyen normalmente los hablantes nativos cuando hay una coma o al final de una frase. Utilice la etiqueta <p> para incluir el párrafo:

```
<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>
```

Esto equivale a especificar una pausa con <break strength="x-strong"/>.

Uso de la pronunciación fonética

<phoneme>

Esta etiqueta es compatible con los formatos TTS de formato largo, neuronal y estándar.

Para que Amazon Polly utilice la pronunciación fonética para un determinado texto, utilice la etiqueta <phoneme>.

La etiqueta `<phoneme>` requiere dos atributos. Indican el alfabeto fonético que utiliza Amazon Polly y los símbolos fonéticos de la pronunciación corregida:

- `alphabet`
  - `ipa`: indica que se usará el Sistema Fonético Internacional (IPA, por sus siglas en inglés).
  - `x-sampa`: indica que se usará el Alfabeto Fonético Extendido SAM (X-SAMPA, por sus siglas en inglés).
- `ph`
  - Especifica los símbolos fonéticos para la pronunciación. Para obtener más información, consulte Idiomas en Amazon Polly

Con la etiqueta `<phoneme>`, Amazon Polly utiliza la pronunciación especificada por el atributo ph en lugar de la pronunciación estándar asociada de forma predeterminada con el idioma utilizado por la voz seleccionada.

Por ejemplo, la palabra "pecan" puede pronunciarse de dos formas. En el ejemplo siguiente, a la palabra “pecan” se le asigna una pronunciación diferente en cada línea. Amazon Polly

pronuncia "pecan" tal y como se especifica en los atributos ph, en lugar de utilizar la pronunciación predeterminada.

Sistema Fonético Internacional (IPA)

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

Métodos ampliados de evaluación del habla: Alfabeto Fonético Extendido SAM (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>
```

Asimismo, el chino mandarín utiliza el pinyin para la pronunciación fonética.

Pinyin

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

En japonés se usa yomigana y la pronunciación 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>
```

Pronunciación kana

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

Control del volumen, velocidad de habla y tono

<prosody>

Los atributos de las etiquetas Prosody son totalmente compatibles con las voces TTS estándar. Las voces generativas, neuronales y de formato largo admiten los `rate` atributos `volume` y, pero no admiten el atributo. `pitch` En el caso de las voces generativas, la etiqueta de prosodia solo se puede usar en oraciones completas.

Para controlar el volumen, la velocidad o el tono de la voz seleccionada, utilice la etiqueta`prosody`.

El volumen, la velocidad de habla, y el tono dependen de la voz seleccionada. Además de las diferencias entre las voces de diferentes idiomas, existen diferencias entre las voces que hablan el mismo idioma. Por este motivo, aunque los atributos son similares en todos los idiomas, existen claras variaciones de un idioma a otro y no hay ningún valor absoluto disponible.

La etiqueta `prosody` tiene tres atributos, cada uno de los cuales tiene varios valores disponibles para establecer el atributo. Todos los atributos utilizan la misma sintaxis:

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

- `volume`
  - `default`: restablece el volumen al nivel predeterminado de la voz actual.
  - `silent`, `x-soft`, `soft`, `medium`, `loud`, `x-loud`: establece el volumen en un valor predefinido de la voz actual.
  - `+ndB`, `-ndB`: cambia el volumen en función del nivel actual. El valor `+0dB` significa que no se producen cambios, `+6dB` significa aproximadamente el doble del volumen actual y `-6dB` significa aproximadamente la mitad del volumen actual.

  Por ejemplo, puede establecer el volumen de un pasaje tal y como se indica a continuación:

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

  También se puede hacer de este 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`: establece el tono en un valor predefinido para la voz seleccionada.
  - n%: un cambio de porcentaje no negativo en la velocidad de habla. Por ejemplo, un valor de 100% significa que no hay ningún cambio en la velocidad de habla; un valor de 200% significa una velocidad dos veces superior a la velocidad predeterminada y un valor de 50% significa una velocidad de habla a la mitad de la velocidad predeterminada. Este valor tiene un intervalo de 20-200%.

  Por ejemplo, puede establecer la velocidad de habla de un pasaje tal y como se indica a continuación:

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

  También se puede hacer de este 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`: restablece el tono en el valor predeterminado de la voz actual.
  - `x-low`, `low`, `medium`, `high`, `x-high`: establece el tono en un valor predefinido de la voz actual.
  - `+n%` o `-n%`: ajusta el tono aplicando un porcentaje relativo. Por ejemplo, un valor de `+0%` significa que no hay ningún cambio en el tono de base de referencia, `+5%` da como resultado un tono de base de referencia un poco más alto y `-5%` da como resultado un tono de base de referencia un poco más bajo.

Por ejemplo, puede establecer el tono de un pasaje tal y como se indica a continuación:

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

También se puede hacer de este modo:

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

La etiqueta **<**prosody**>** debe contener al menos un atributo, pero puede incluir más dentro de la misma etiqueta.

```
<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>
```

También se puede combinar con etiquetas anidadas, tal y como se indica a continuación:

```
<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>
```

Establecimiento de una duración máxima para voz sintetizada

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

Esta etiqueta solo es compatible actualmente con el formato TTS estándar.

Para controlar el tiempo que desea que tarde un fragmento de voz cuando se sintetiza, utilice la etiqueta `<prosody>` con el atributo `amazon:max-duration`.

La duración del fragmento de voz sintetizado varía ligeramente, en función de la voz que seleccione. Esto puede dificultar la coincidencia del fragmento de voz sintetizado con elementos visuales u otras actividades que requieran una sincronización precisa. Este problema aumenta en el caso de aplicaciones de traducción, ya que el tiempo que se tarda en decir frases concretas puede variar notablemente en distintos idiomas.

La etiqueta `<prosody amazon:max-duration>` asigna el fragmento de voz sintetizada a la cantidad de tiempo que desea que tarde (la duración).

Esta etiqueta utiliza la siguiente sintaxis:

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

Con la etiqueta `<prosody amazon:max-duration>`, puede especificar la duración en segundos o milisegundos:

- *n*`s`: la duración máxima en segundos
- *n*`ms`: la duración máxima en milisegundos

Por ejemplo, el siguiente texto hablado tiene una duración máxima de 2 segundos:

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

Texto colocado dentro de la etiqueta, no supera la duración especificada. Si la voz o el idioma elegido normalmente requiere más tiempo que la duración, Amazon Polly acelera el fragmento de voz de modo que se ajuste a la duración especificada.

Si la duración especificada es superior a lo que se tarda en leer el texto en una velocidad normal, Amazon Polly lee el fragmento con normalidad. No ralentiza el fragmento de voz ni añade silencio, por lo que el audio resultante es más corto de lo necesario.

> **Note**
>
> Amazon Polly aumenta la velocidad no más de 5 veces la velocidad normal. Si el texto se lee más rápido que esto, por lo general no tiene sentido. Si un fragmento de voz no puede

ajustarse a la duración especificada, incluso aunque la velocidad se acelere al máximo, el audio se acelerará, pero durará más de la duración especificada.

Puede incluir una sola frase o varias frases dentro de una etiqueta `<prosody amazon:max-duration>` y puede utilizar varias etiquetas `<prosody amazon:max-duration>` en su texto.

Por ejemplo:

```
<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>
```

El uso de la etiqueta `<prosody amazon:max-duration>` puede aumentar la latencia cuando Amazon Polly devuelve un fragmento de voz sintetizada. El grado de latencia depende del fragmento y de su longitud. Le recomendamos que utilice texto compuesto por fragmentos de texto relativamente cortos.

Limitaciones

Existen limitaciones, tanto en la forma de utilizar la etiqueta `<prosody amazon:max-duration>` y en cómo funciona con otras etiquetas de SSML:

- El texto dentro de una etiqueta `<prosody amazon:max-duration>` no puede tener más de 1500 caracteres.

- No puede anidar etiquetas <prosody amazon:max-duration>. Si coloca una etiqueta <prosody amazon:max-duration> dentro de otra, Amazon Polly omite la etiqueta interior.

  Por ejemplo, en el caso siguiente se omite la etiqueta <prosody amazon:max-duration="5s">:

```
<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>
```

- No se pueden utilizar las etiquetas <prosody> con el atributo rate dentro de una etiqueta <prosody amazon:max-duration>. Esto se debe a que ambas afectan a la velocidad a la que se dicta el texto.

  En el ejemplo siguiente, Amazon Polly omite la etiqueta <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>
```

Pausas y **max-duration**

Cuando se utiliza la etiqueta `max-duration`, puede insertar pausas en el texto. Sin embargo, Amazon Polly incluye la longitud de la pausa al calcular la duración máxima del fragmento de voz. Además, Amazon Polly conserva las pausas breves que se producen cuando hay comas y puntos en un fragmento y las incluye en la duración máxima.

Por ejemplo, en el siguiente bloque, la interrupción de 600 milisegundos y la interrupción provocada por comas y puntos se produce en el fragmento de texto de 8 segundos:

```
<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>
```

Agregación de una pausa entre frases

<s>

Esta etiqueta es compatible con los formatos TTS generativo, de formato largo, neuronal y estándar.

Para añadir una pausa entre líneas o frases del texto, utilice la etiqueta <s>. El uso de esta etiqueta tiene el mismo efecto que:

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

A diferencia de la etiqueta `<break>`, la etiqueta **<s>** incluye la frase. Esto es útil para sintetizar la voz de texto organizado en líneas, en lugar de en frases, como los poemas.

En el siguiente ejemplo, la etiqueta <s> inserta una breve pausa después de la primera y segunda frase. La última frase no tiene ninguna etiqueta <s>, pero también se aplica una breve pausa detrás de ella porque termina 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>
```

Control de cómo se leen los tipos especiales de palabras

<say-as>

La `<say-as>` etiqueta es compatible con motores TTS generativos, de formato largo, neuronales y estándar. Sin embargo, ten en cuenta que si Amazon Polly utiliza una voz neuronal y encuentra la `<say-as>` etiqueta con la `characters` opción en tiempo de ejecución, la frase afectada se sintetizará con la voz estándar relacionada. Sin embargo, la frase afectada se seguirá facturando como si utilizara una voz neuronal.

Utilice la etiqueta `<say-as>` con el atributo `interpret-as` para indicar a Amazon Polly cómo leer determinados caracteres, palabras y números. De este modo, puede proporcionar información adicional para eliminar cualquier ambigüedad sobre cómo Amazon Polly debe representar el texto.

La etiqueta `<say-as>` utiliza un atributo, `interpret-as`, que usa una serie de valores posibles disponibles. Todos ellos utilizan la misma sintaxis:

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

Los valores siguientes están disponibles con `interpret-as`:

- `characters` o `spell-out`: deletrea cada letra del texto, como en a-b-c.

> **Note**
>
> Esta opción no se admite actualmente para voces neuronales. Si utiliza una voz neuronal y este código SSML encontrado por Amazon Polly en tiempo de ejecución, la frase afectada se sintetizará mediante la voz estándar relacionada. Tenga en cuenta, sin embargo, que esta frase se seguirá facturando como si utilizara una voz neuronal.

- `cardinal` o `number`: interpreta el valor numérico como un número cardinal; por ejemplo, 1.234.
- `ordinal`: interpreta el texto numérico como un número ordinal; por ejemplo, 1.234º.
- `digits`: deletrea cada dígito por separado; por ejemplo, 1-2-3-4.

- `fraction`: interpreta el texto numérico como una fracción. Esto es aplicable a las fracciones comunes, como 3/20, y a las fracciones mixtas, como 2 ½. Para obtener más información, consulte las secciones siguientes.
- `unit`: interpreta un texto numérico como una medida. El valor debe ser un número o una fracción seguido de una unidad (sin espacio entre ellos) como en `1/2inch`, o solo de una unidad, como en `1meter`.
- `date`: interpreta el texto como una fecha. El formato de la fecha debe especificarse con el atributo format. Para obtener más información, consulte las secciones siguientes.
- `time`: interpreta el texto numérico como una cantidad de tiempo en minutos y segundos; por ejemplo, `1'21"`.
- `address`: interpreta el texto como parte de una dirección postal.
- `expletive`: emite un pitido en lugar del contenido incluido en la etiqueta.
- `telephone`: interpreta el texto numérico como un número de teléfono de 7 o 10 dígitos, como en `2025551212`. También puede utilizar este valor para las extensiones telefónicas, como en `2025551212x345`. Para obtener más información, consulte las secciones siguientes.

> **Note**
>
> En la actualidad, la opción `telephone` no está disponible para todos los idiomas. Sin embargo, está disponible para voces que hablan variantes del idioma inglés (en-AU, en-GB, en-IN, en-US y en-GB-WLS), variantes del idioma español (es-ES, es-MX y es-US), variantes del idioma francés (fr-FR y fr-CA) y variantes portuguesas (pt-BR y pt-PT), así como para alemán (de-DE), italiano (it-IT), japonés (ja-JP) y ruso (ru-IT) RU). También debe tenerse en cuenta que, en determinados casos, algunos idiomas como el árabe (arb) administran automáticamente el número establecido como número de teléfono y, por lo tanto, no implementan realmente la etiqueta SSML `telephone`.

Fracciones

Amazon Polly interpreta los valores dentro de la etiqueta `say-as` que tengan el atributo `interpret-as="fraction"` como fracciones comunes. A continuación se presenta la sintaxis de las fracciones:

- Fracción

  Sintaxis:*`cardinal number`*/*`cardinal number`*, como 2/9.

Por ejemplo: `<say-as interpret-as="fraction">2/9</say-as>` se pronuncia "dos novenos."

- Número mixto no negativo

  Sintaxis: *`cardinal number`* +*`cardinal number`*/*`cardinal number`*, como 3+1/2.

  Por ejemplo, `<say-as interpret-as="fraction">3+1/2</say-as>` se pronuncia "tres y medio".

  > **Note**
  >
  > Debe haber un signo + entre "3" y "1/2". Amazon Polly no admite un número mixto sin el signo +, como, por ejemplo, "3 1/2".

Fechas

Cuando `interpret-as` se establece en `date`, también se deberá indicar el formato de la fecha.

Utiliza la siguiente sintaxis:

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

Por ejemplo:

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

Los siguientes formatos pueden utilizarse con el atributo `date`.

- `mdy`: Month-day-year.
- `dmy`: Day-month-year.
- `ymd`: Year-month-day.
- `md`: mes-día.
- `dm`: día-mes.
- `ym`: año-mes.

- `my`: mes-año.
- `d`: Día.
- `m`: Month.
- `y`: Year.
- `yyyymmdd`:. Year-month-day Si utilizas este formato, puedes hacer que Amazon Polly se salte partes de la fecha con signos de interrogación.

  Por ejemplo, Amazon Polly reproduce lo siguiente como "22 de septiembre":

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

  `Format` no es necesario.

Teléfono

Amazon Polly intenta interpretar correctamente el texto proporcionado a partir de su formato aun cuando no aparece la etiqueta `<say-as>`. Por ejemplo, si el texto incluye "202-555-1212", Amazon Polly lo interpreta como un número de teléfono de 10 cifras y lee cada cifra individualmente, aplicando una breve pausa en cada guion. En este caso, no es necesario usar `<say-as interpret-as="telephone">`. Sin embargo, si proporciona el texto “2025551212” y quiere que Amazon Polly lo lea como un número de teléfono, debería especificar `<say-as interpret-as="telephone">`.

La lógica para interpretar cada elemento depende del idioma. Por ejemplo, en inglés de EE. UU. y en inglés de Reino Unido, los números de teléfono no se expresan igual (en inglés de Reino Unido, se agrupan las secuencias del mismo número; por ejemplo, "doble cinco" o "triple cuatro"). Para ver la diferencia, puede probar el siguiente ejemplo con una voz de Estados Unidos y otra de Reino Unido:

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

Pronunciación de acrónimos y abreviaturas

<sub>

Esta etiqueta es compatible con los formatos TTS generativo, de formato largo, neuronal y estándar.

Utilice la etiqueta <sub> con el atributo `alias` para sustituir una sola palabra (o pronunciación) del texto seleccionado como un acrónimo o una abreviatura.

Se utiliza esta sintaxis:

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

En el siguiente ejemplo, el nombre "Mercury" (Mercurio) se sustituye por el símbolo químico del elemento para que el contenido de audio sea más claro.

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

Mejora de la pronunciación especificando partes del discurso

<w>

Esta etiqueta es compatible con los formatos TTS generativo, de formato largo, neuronal y estándar.

Puede utilizar la etiqueta **<w>** para personalizar la pronunciación de las palabras especificando su categoría gramatical o su significado alternativo. Esto se realiza mediante el atributo `role`.

Esta etiqueta utiliza la siguiente sintaxis:

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

Los siguientes valores pueden utilizarse con el atributo `role`:

Para especificar la categoría gramatical:

- `amazon:VB`: interpreta la palabra como un verbo (presente simple).
- `amazon:VBD`: interpreta la palabra como un verbo en tiempo pasado.
- `amazon:DT`: interpreta la palabra como determinante.
- `amazon:IN`: interpreta la palabra como una preposición.
- `amazon:JJ`: interpreta la palabra como un adjetivo.
- `amazon:NN`: interpreta la palabra como un sustantivo.

Por ejemplo, dependiendo de la categoría gramatical, la pronunciación en inglés de Estados Unidos de la palabra "read" varia en función de la etiqueta:

```
<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>
```

Para especificar un significado específico:

- `amazon:DEFAULT`: usa el sentido predeterminado de la palabra.
- `amazon:SENSE_1`: cuando es aplicable, utiliza un sentido de la palabra que no es el predeterminado. Por ejemplo, el nombre "bass" se pronuncia de forma diferente en función de su significado. El significado predeterminado es la parte más baja de la escala musical. El significado alternativo es una especie de pez de agua dulce, también denominado "bass", pero que se pronuncia de forma diferente. Si se utiliza `<w role="amazon:SENSE_1">bass</w>`, se aplica la pronunciación no predeterminada (es decir, pez de agua dulce) en el texto del audio.

Esta diferencia en la pronunciación y el significado se puede oír si se sintetiza lo siguiente:

```
<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**
>
> Algunos idiomas pueden tener otra selección de categorías gramaticales.

Adición de sonido de respiración

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

Esta etiqueta solo es compatible con el formato TTS estándar.

La voz con sonido natural incluye tanto palabras habladas correctamente como sonidos de respiración. Al añadir sonidos de respiración a la voz sintetizada, puede hacer que suene más natural. Las etiquetas `<amazon:breath>` y `<amazon:auto-breaths>` proporcionan respiraciones. Dispone de las opciones siguientes:

- Modo manual: puede establecer la ubicación, la duración y el volumen de un sonido de respiración en el texto
- Modo automático: Amazon Polly inserta automáticamente sonidos de respiración en la salida de voz
- Modo mixto: tanto usted como Amazon Polly añaden sonidos de respiración

Modo manual

En el modo manual, debe colocar la etiqueta `<amazon:breath/>` en el texto de entrada donde desee ubicar una respiración. Puede personalizar la duración y el volumen de las respiraciones con los atributos `duration` y `volume`, respectivamente:

- `duration`: controla la duración de la respiración. Los valores válidos son: `default`, `x-short`, `short`, `medium`, `long`, `x-long`. El valor predeterminado es `medium`.
- `volume`: controla el volumen de la respiración. Los valores válidos son: `default`, `x-soft`, `soft`, `medium`, `loud`, `x-loud`. El valor predeterminado es `medium`.

> **Note**
>
> La duración y el volumen exactos de cada valor de atributo dependen de la voz utilizada de Amazon Polly.

Para establecer un sonido de respiración con los valores predeterminados, utilice `<amazon:breath/>` sin atributos.

Por ejemplo, para utilizar atributos con el objeto de definir la duración y el volumen de una respiración a la mitad, configure los atributos de la siguiente forma:

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

```
</speak>
```

Para utilizar los valores predeterminados, se usa la etiqueta:

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

Puede añadir sonidos de respiración individuales en un pasaje, tal y como se indica a continuación:

```
<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>
```

Modo automático

En el modo automático, se utiliza la etiqueta `<amazon:auto-breaths>` para indicar a Amazon Polly que cree automáticamente ruidos de respiración en los intervalos apropiados. Puede establecer la frecuencia de los intervalos, su volumen y su duración. Coloque la etiqueta `</amazon:auto-breaths>` al principio del texto al que desee aplicar respiración automática y cierre la etiqueta al final.

> **Note**
>
> A diferencia de la etiqueta de modo manual, `<amazon:breath/>`, la etiqueta `<amazon:auto-breaths>` necesita una etiqueta de cierre (`</amazon:auto-breaths>`).

Puede utilizar los siguientes atributos opcionales con la etiqueta `<amazon:auto-breaths>`:

- `volume`: controla el volumen de la respiración. Los valores válidos son: `default`, `x-soft`, `soft`, `medium`, `loud`, `x-loud`. El valor predeterminado es `medium`.

- `frequency`: controla la frecuencia con la que se producen los sonidos de respiración en el texto. Los valores válidos son: `default`, `x-low`, `low`, `medium`, `high`, `x-high`. El valor predeterminado es `medium`.
- `duration`: controla la duración de la respiración. Los valores válidos son: `default`, `x-short`, `short`, `medium`, `long`, `x-long`. El valor predeterminado es `medium`.

De forma predeterminada, la frecuencia de sonidos de respiración depende del texto de entrada. Sin embargo, sonidos de respiración se suelen producir después de comas y puntos.

En los siguientes ejemplos se muestra cómo usar la etiqueta `<amazon:auto-breaths>`. Para decidir qué opciones se utilizarán en su contenido, copie los ejemplos correspondientes en la consola de Amazon Polly y escuche las diferencias.

- Uso del modo automático sin parámetros opcionales.

```
<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>
```

- Uso del modo automático con control de volumen. Los parámetros no especificados (`duration` y `frequency`) se establecen en los valores predeterminados (`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>
```

- Uso del modo automático con control de frecuencia. Los parámetros no especificados (`duration` y `volume`) se establecen en los valores predeterminados (`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>
```

- Uso del modo automático con varios parámetros. Para el parámetro sin especificar `Duration`, Amazon Polly utiliza el valor predeterminado (`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>
```

Estilo de habla Newscaster

<amazon: domain name =" news">

El estilo presentador solo está disponible para las voces de Matthew o Joanna, que solo están disponibles en inglés de Estados Unidos (en-US), y Lupe, en español estadounidense (es-US). Solo se admite cuando se usa el formato `Neural`.

Para utilizar el estilo Newscaster, utilice etiquetas SSML y la siguiente sintaxis:

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

Por ejemplo, puede utilizar el estilo presentador con la voz de Matthew de la siguiente manera:

```
<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>
```

Adición de compresión de rango dinámico

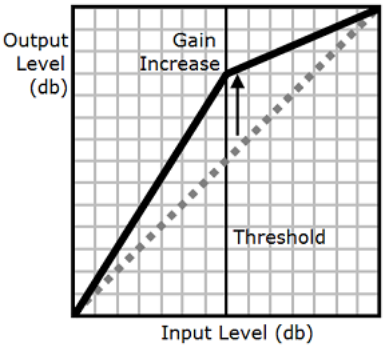
<amazon:effect name="drc">

Esta etiqueta es compatible con los formatos TTS de formato largo, neuronal y estándar.

En función del texto, el idioma y la voz utilizados en un archivo de audio, los sonidos varían desde suaves a altos. Con frecuencia, los sonidos ambientales, como el sonido de un vehículo en movimiento, pueden enmascarar los sonidos más suaves, lo que impide oír con claridad la pista de

audio. Para mejorar el volumen de determinados sonidos en el archivo de audio, utilice la etiqueta de compresión de rango dinámico (`drc`).

La etiqueta `drc` establece un umbral de "volumen" de rango medio para el audio y aumenta el volumen (la ganancia) de los sonidos alrededor de dicho umbral. El mayor aumento de ganancia se aplica cerca del umbral, y dicho aumento se va reduciendo al alejarse del umbral.



Esto facilita la audición de los sonidos de rango medio en un entorno ruidoso, lo que permite que todo el archivo de audio se oiga con más claridad.

La etiqueta `drc` es un parámetro booleano (está presente o no lo está). Utiliza la sintaxis: `<amazon:effect name="drc">` y se cierra con `</amazon:effect>`.

Puede utilizar la etiqueta `drc` con cualquier voz o idioma compatible con Amazon Polly. Puede aplicarla a una sección entera de la grabación o solo a unas pocas palabras. Por ejemplo:

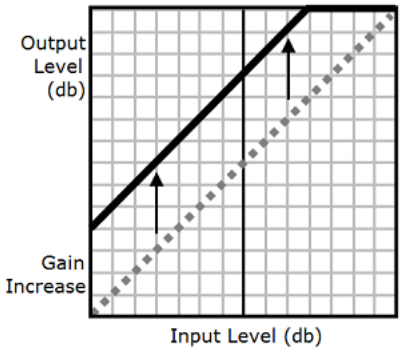
```
<speak>
     Some audio is difficult to hear in a moving vehicle, but <amazon:effect
 name="drc"> this audio
     is less difficult to hear in a moving vehicle.</amazon:effect>
</speak>
```

> **Note**
>
> Si utiliza "`drc`" en la sintaxis , esta distingue entre mayúsculas y minúsculas.`amazon:effect`

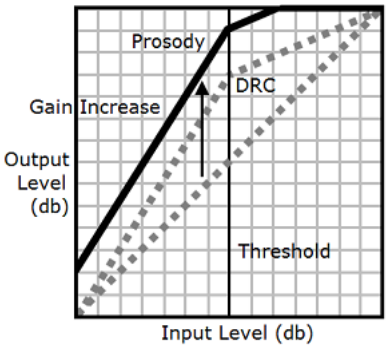
Uso de **`drc`** con la etiqueta **`prosody volume`**

Como muestra el siguiente gráfico, la etiqueta `prosody volume` aumenta de manera uniforme el volumen de un archivo de audio desde el nivel original (línea discontinua) a un nivel ajustado (línea

continua). Para aumentar aún más el volumen de determinadas partes del archivo, utilice la etiqueta `drc` con la etiqueta `prosody volume`. La combinación de etiquetas no afecta a la configuración de la etiqueta `prosody volume`.



Cuando se utilizan las etiquetas `drc` y `prosody volume` juntas, Amazon Polly aplica primero la etiqueta `drc`, aumentando los sonidos de rango medio (los que están cerca del umbral). A continuación, aplica la etiqueta `prosody volume` y aumenta el volumen de toda la pista de audio de manera uniforme.



Para utilizar las etiquetas juntas, anide una dentro de la otra. Por ejemplo:

```
<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>
```

En este texto, la etiqueta `prosody volume` establece el nivel de volumen de todo el pasaje en "loud" (alto). La etiqueta `drc` mejora el volumen de los valores de rango medio de la segunda frase.

> **Note**
>
> Cuando utilice las etiquetas `drc` y `prosody volume` juntas, emplee las prácticas estándar de XML para el anidado de etiquetas.

Habla de forma suave

<amazon:effect phonation="soft">

Esta etiqueta solo es compatible actualmente con el formato TTS estándar.

Para especificar que el texto introducido debe pronunciarse por softer-than-normal voz, usa la <amazon:effect phonation="soft">etiqueta.

Se utiliza esta sintaxis:

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

Por ejemplo, puede utilizar esta etiqueta con la voz de Matthew tal y como se indica a continuación:

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

Control del timbre

<amazon:effect > vocal-tract-length

Esta etiqueta solo es compatible actualmente con el formato TTS estándar.

El timbre es la calidad tonal de una voz que permite diferenciar entre voces distintas, aunque todas tengan el mismo tono y volumen. Una de las características fisiológicas más importantes que contribuye al timbre de voz es la longitud del tracto vocal. El tracto vocal es una cavidad de aire que se extiende desde la parte superior de las cuerdas vocales hasta el borde de los labios.

Para controlar el timbre del fragmento hablado en Amazon Polly, utilice la etiqueta `vocal-tract-length`. Esta etiqueta tiene el efecto de modificar la longitud del tracto vocal del altavoz, lo que hace que este suene como si hubiese cambiado de tamaño. Al aumentar el valor de `vocal-tract-length`, el altavoz suena como si fuera físicamente más grande. Al reducirlo, el altavoz suena como si fuera más pequeño. Puedes usar esta etiqueta con cualquiera de las voces de la cartera de Amazon Polly. Text-to-Speech

Para cambiar el timbre, utilice los siguientes valores:

- `+n%` o `-n%`: ajusta la longitud del tracto vocal aplicando un cambio de porcentaje relativo a la voz actual. Por ejemplo, +4% o -2%. Los valores válidos están comprendidos entre +100% y -50%. Los valores que están fuera de este intervalo se recortan. Por ejemplo, +111% suena como +100% y -60% suena como -50%.
- `n%`: cambia la longitud del tracto vocal en un porcentaje absoluto de la longitud del tracto de la voz actual. Por ejemplo, 110% o 75%. Un valor absoluto del 110% equivale a un valor relativo de +10%. Un valor absoluto del 100% equivale al valor predeterminado para la voz actual.

El siguiente ejemplo muestra cómo modificar la longitud del tracto vocal para cambiar el timbre:

```
<speak>
     This is my original voice, without any modifications. <amazon:effect vocal-tract-
length="+15%">
     Now, imagine that I am much bigger. </amazon:effect> <amazon:effect vocal-tract-
length="-15%">
     Or, perhaps you prefer my voice when I'm very small. </amazon:effect> You can also
 control the
     timbre of my voice by making minor adjustments. <amazon:effect vocal-tract-
length="+10%">
     For example, by making me sound just a little bigger. </
amazon:effect><amazon:effect
     vocal-tract-length="-10%"> Or, making me sound only somewhat smaller. </
amazon:effect>
</speak>
```

Combinación de varias etiquetas

Puede combinar la etiqueta `vocal-tract-length` con cualquier otra etiqueta SSML compatible con Amazon Polly. Dado que el timbre (longitud del tracto vocal) y el tono están íntimamente relacionados, es probable que obtenga resultados óptimos si utiliza las etiquetas `vocal-tract-length` y `<prosody pitch>`. Para crear la voz más realista, le recomendamos utilizar diferentes

porcentajes de cambio para ambas etiquetas. Experimente con varias combinaciones para obtener los resultados que desea.

El siguiente ejemplo muestra cómo combinar etiquetas.

```
<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>
```

Susurros

<amazon:effect name="whispered">

Esta etiqueta solo es compatible actualmente con el formato TTS estándar.

Esta etiqueta indica que la entrada de texto debe leerse en susurros y no con voz normal. Se puede usar con cualquiera de las voces de la cartera de Amazon Polly. Text-to-Speech

Utiliza la siguiente sintaxis:

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

Por ejemplo:

```
<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>
```

En este caso, la parte del discurso que dice el personaje se leerá en susurros, mientras que la frase "she said" se leerá de forma normal con la voz de Amazon Polly elegida.

Puede mejorar el efecto de "susurro" ralentizando el ritmo prosódico hasta en un 10%, en función del efecto deseado.

Por ejemplo:

```
<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>
```

Cuando se generan las comillas para una voz susurrada, la secuencia de audio debe incluir también la voz susurrada para garantizar que las comillas coinciden con la secuencia de audio.

Administración de lexicones

Los lexicones de pronunciación le permiten personalizar la pronunciación de las palabras. Amazon Polly proporciona operaciones de API que puede utilizar para almacenar léxicos en una región. AWS Estos lexicones serán específicos de esa región. Puede utilizar uno o varios lexicones de esa región cuando sintetice texto mediante la operación `SynthesizeSpeech`. Esta operación aplica el lexicón especificado al texto de entrada antes de que comience el proceso de síntesis. Para obtener más información, consulte SynthesizeSpeech.

> **Note**
>
> Los lexicones deben ajustarse a la recomendación de W3C sobre Pronunciation Lexicon Specification (PLS, Especificación de lexicones de pronunciación). Para obtener más información, consulte Pronunciation Lexicon Specification (PLS) versión 1.0 en el sitio web de W3C.

A continuación, se incluyen algunos ejemplos sobre el uso de lexicones con motores de síntesis de voz:

- A veces, en algunas palabras, se utilizan números para sustituir las letras, por ejemplo, "g3t sm4rt" (get smart). Los humanos pueden leer estas palabras correctamente. Sin embargo, un motor Text-to-Speech (TTS) lee el texto literalmente y pronuncia el nombre exactamente como se escribe. En este tipo de casos, se pueden utilizar lexicones para personalizar la síntesis de voz mediante Amazon Polly. En este ejemplo, se puede especificar un alias (get smart) para la palabra "g3t sm4rt" del lexicón.
- Supongamos que el texto contiene un acrónimo, como W3C. Puede utilizar un lexicón para definir un alias para la palabra W3C, de forma que se lea en su forma expandida (World Wide Web Consortium).

Los lexicones proporcionan mayor control acerca de cómo Amazon Polly pronuncia palabras que son poco frecuentes en el idioma seleccionado. Por ejemplo, puede especificar la pronunciación utilizando un alfabeto fonético. Para obtener más información, consulte Pronunciation Lexicon Specification (PLS) versión 1.0 en el sitio web de W3C.

Temas

- Uso de varios lexicones
- Carga de un lexicón
- Aplicación de lexicones (síntesis del habla)
- Filtrado de la lista de lexicones en la consola
- Descarga de lexicones en la consola
- Eliminación de un lexicón

Uso de varios lexicones

Puede aplicar hasta cinco lexicones al texto. Si el mismo grafema aparece en varios lexicones distintos que se aplican al texto, el orden en el que se apliquen marcará una diferencia en el fragmento hablado resultante. Por ejemplo, supongamos que tenemos el texto siguiente, "Hello, my name is Bob" ("Hola, me llamo Bob"). y dos lexemas en lexicones distintos que utilizan el grafema `Bob`.

LexA

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

LexB

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

Si el orden de los lexicones es primero LexA y luego LexB, el resultado de la síntesis de voz será "Hello, my name is Robert" ("Hola, me llamo Robert"). Si el orden es el contrario, y primero va LexB y después LexA, el resultado de la síntesis de voz será "Hello, my name is Bobby ("Hola, me llamo Bobby").

Example – Aplicación de LexA antes que LexB

```
aws polly synthesize-speech \
--lexicon-names LexA LexB \
```

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

Fragmento hablado resultante: "Hello, my name is Robert" ("Hola, me llamo Robert").

Example – Aplicación de LexB antes que LexA

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

Fragmento hablado resultante: "Hello, my name is Bobby" ("Hola, me llamo Bobby").

Para obtener más información sobre la aplicación de lexicones a través de la consola de Amazon Polly, consulte Aplicación de lexicones (síntesis del habla).

Carga de un lexicón

Los lexicones que use deben ajustarse a la recomendación de W3C sobre Pronunciation Lexicon Specification (PLS, Especificación de lexicones de pronunciación). Para obtener más información, consulte Pronunciation Lexicon Specification (PLS) versión 1.0 en el sitio web de W3C.

Console - Lexicons tab

Para utilizar un lexicón de pronunciación, primero debe cargarlo. Hay dos ubicaciones en la consola desde las que se puede cargar un lexicón, la pestaña Text-to-Speech y la pestaña Lexicons.

En los procesos siguientes se explica cómo agregar lexicones que puede utilizar para personalizar el modo en que las palabras y las frases poco comunes se pronuncian en el idioma elegido.

Para agregar un lexicón desde la pestaña Lexicons

1. Inicia sesión en la consola Amazon Polly AWS Management Console y ábrela en. https://console.aws.amazon.com/polly/

2. Elija la pestaña Lexicons.
3. Seleccione Cargar lexicón.
4. Proporcione un nombre para el lexicón y, a continuación, utilice Elija un archivo de lexicón para buscar el lexicón que desea cargar. Solo puede cargar archivos PLS con las extensiones .pls o .xml.
5. Seleccione Cargar lexicón. Si ya existe un lexicón con el mismo nombre (ya sea un archivo .pls o un archivo .xml), al cargar el lexicón, se remplazará el lexicón existente.

Console - TTS tab

Para añadir un léxico desde la pestaña text-to-Speech

1. Inicia sesión en la consola Amazon Polly AWS Management Console y ábrela en. https://console.aws.amazon.com/polly/
2. Elija la pestaña Text-to-Speech.
3. Amplíe Configuración adicional, active Personalizar pronunciación y, a continuación, seleccione Cargar lexicón.
4. Proporcione un nombre para el lexicón y, a continuación, utilice Elija un archivo de lexicón para buscar el lexicón que desea cargar. Solo puede emplear archivos PLS que utilicen las extensiones .pls y.xml.
5. Seleccione Cargar lexicón. Si ya existe un lexicón con el mismo nombre (ya sea un archivo .pls o un archivo .xml), al cargar el lexicón, se remplazará el lexicón existente.

AWS CLI - one lexeme

Con Amazon Polly, puede usarlo PutLexicon para almacenar léxicos de pronunciación en una AWS región específica de su cuenta. A continuación, podrá especificar uno o varios de los lexicones almacenados en la solicitud SynthesizeSpeech que desee aplicar antes de que el servicio comience a sintetizar el texto. Para obtener más información, consulte Administración de lexicones.

Supongamos que tiene el siguiente lexicón, que se ajusta a la recomendación PLS de 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>
```

Tenga en cuenta lo siguiente:

- Los dos atributos especificados en el elemento `<lexicon>`:
  - El atributo `xml:lang` especifica el código de idioma, `en-US`, al que se va a aplicar el lexicón. Amazon Polly puede utilizar este lexicón de ejemplo si la voz que se especifica en la llamada a `SynthesizeSpeech` tiene el mismo código de idioma (en-US).

    > **Note**
    >
    > Puede utilizar la operación `DescribeVoices` para buscar el código de idioma asociado a una voz.

  - El atributo `alphabet` está establecido en `IPA`, lo que significa que se va a utilizar el Alfabeto Fonético Internacional (IPA, por sus siglas en inglés) para especificar la pronunciación. IPA es uno de los alfabetos disponibles para escribir las pronunciaciones. Amazon Polly también admite el Alfabeto Fonético Extendido SAM (X-SAMPA, por sus siglas en inglés).

- El elemento `<lexeme>` describe la correspondencia entre `<grapheme>` (es decir, la representación gráfica textual de la palabra) y `<alias>`.

Para probar este lexicón, haga lo siguiente:

1. Guarde el lexicón como `example.pls`.
2. Ejecute el `put-lexicon` AWS CLI comando para almacenar el léxico (con el nombre`w3c`) en la región us-east-2.

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

3. Ejecute el comando `synthesize-speech` para sintetizar el texto de ejemplo en una secuencia de audio (`speech.mp3`) y especifique el parámetro opcional `lexicon-name`.

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

4. Reproduzca el archivo `speech.mp3` resultante y observe que la palabra W3C del texto se ha sustituido por World Wide Web Consortium.

En el lexicón del ejemplo anterior se utiliza un alias. El alfabeto IPA mencionado en el lexicón no se utiliza. El lexicón siguiente especifica una pronunciación fonética que utiliza el elemento <phoneme> con el 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>
```

Siga los mismos pasos para probar este lexicón. No olvide especificar un texto de entrada que tenga la palabra “pecan”; por ejemplo, “Pecan pie is delicious” ("La tarta de nueces está deliciosa").

Consulte los siguientes recursos para ver ejemplos de código adicionales para el funcionamiento de la PutLexicon API:

- Muestra de Java: PutLexicon
- Muestra de Python (Boto3): PutLexicon

AWS CLI - multiple lexemes

Con Amazon Polly, puede usarlo PutLexicon para almacenar léxicos de pronunciación en una AWS región específica de su cuenta. A continuación, podrá especificar uno o varios de los lexicones almacenados en la solicitud SynthesizeSpeech que desee aplicar antes de que el servicio comience a sintetizar el texto. Para obtener más información, consulte Administración de lexicones.

En este ejemplo, el lexema especificado en el lexicón se aplica exclusivamente al texto de entrada del proceso de síntesis. Pongamos como ejemplo el siguiente lexicón:

```
<?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>
```

El lexicón especifica tres lexemas, dos de los cuales definen un alias para el grafema W3C, tal y como se indica a continuación:

- El primer elemento <lexeme> define un alias (World Wide Web Consortium).
- El segundo elemento <lexeme> define un alias alternativo (WWW Consortium).

Amazon Polly utiliza la primera palabra de sustitución con cualquier grafema del lexicón.

El tercer elemento <lexeme> establece una palabra sustituta (Community) para "Consortium".

En primer lugar, vamos a probar este lexicón. Supongamos que desea sintetizar el siguiente texto de ejemplo en un archivo de audio (speech.mp3) y que especifica el lexicón en una llamada a SynthesizeSpeech.

```
The W3C is a Consortium
```

SynthesizeSpeech se aplica al lexicón de la forma siguiente:

- De acuerdo con el primer lexema, la palabra W3C se modifica a World Wide Web Consortium. El texto revisado sería el siguiente:

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

- El alias definido en el tercer lexema solamente se aplica a la palabra Consortium que formaba parte del texto original, por lo que se obtiene el siguiente texto:

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

Puede probarlo de la siguiente AWS CLI manera:

1. Guarde el lexicón como example.pls.
2. Ejecute el comando put-lexicon para guardar el lexicón con el nombre w3c en la región us-east-2.

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

3. Ejecute el comando list-lexicons para comprobar que el lexicón w3c está en la lista de lexicones devuelta.

```
aws polly list-lexicons
```

4. Ejecute el comando `synthesize-speech` para sintetizar el texto de ejemplo en un archivo de audio (`speech.mp3`) y especifique el parámetro opcional `lexicon-name`.

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

5. Reproduzca el archivo `speech.mp3` resultante para comprobar que la síntesis del fragmento hablado refleja los cambios del texto.

Consulta los siguientes recursos para ver ejemplos de código adicionales para el funcionamiento de la PutLexicon API:

- Muestra de Java: PutLexicon
- Muestra de Python (Boto3): PutLexicon

Aplicación de lexicones (síntesis del habla)

Los lexicones que use deben ajustarse a la recomendación de W3C sobre Pronunciation Lexicon Specification (PLS, Especificación de lexicones de pronunciación). Para obtener más información, consulte Pronunciation Lexicon Specification (PLS) versión 1.0 en el sitio web de W3C.

Console

En el procedimiento siguiente, se muestra cómo se aplica un lexicón al texto de entrada. Para ello, se utiliza el lexicón `W3c.pls`, que sustituye "World Wide Web Consortium" por "W3C". Si aplica varios lexicones al texto, se aplicarán en orden descendente, por lo que las coincidencias del primer lexicón tendrán prioridad sobre las de los siguientes. El lexicón solamente se aplica al texto si el idioma especificado en él es el mismo que el idioma elegido.

Puede aplicar un lexicón a una entrada de texto sin formato o a una entrada SSML.

Example – Aplicar el lexicón W3C.pls

Para crear el lexicón necesario para este ejercicio, consulte Carga de un lexicón. Utilice un editor de texto sin formato para crear el lexicón W3C.pls que aparece al comienzo del tema. No olvide la ubicación en la que guarda el archivo.

Para aplicar el lexicón W3C.pls a la entrada

En este ejemplo, hemos introducido un lexicón para sustituir "World Wide Web Consortium" por "W3C". Compare los resultados de este ejercicio con los de Uso de SSML en la consola tanto en inglés de Estados Unidos como en otro idioma.

1. Inicia sesión en la consola Amazon Polly AWS Management Console y ábrela en. https://console.aws.amazon.com/polly/
2. Realice una de las siguientes acciones:
   - Desactive SSML y escriba o pegue este texto en el cuadro de entrada de texto.

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

   - Active SSML y escriba o pegue este texto en el cuadro de entrada de texto.

     ```
     <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. En la lista Idioma, elija inglés de Estados Unidos y seleccione la voz que desee usar para este texto.
4. Expanda Configuración adicional y active Personalizar pronunciación.
5. En la lista de lexicones, seleccione `W3C (English, US)`.

   Si el lexicón `W3C (English, US)` no aparece en la lista, elija Upload lexicon y cárguelo. A continuación, selecciónelo en la lista. Para crear este lexicón, consulte Carga de un lexicón.
6. Si desea escuchar el fragmento hablado inmediatamente, elija Escuchar.
7. Para guardar el fragmento hablado en un archivo,

   a. Elija Descargar.

b. Para cambiar el formato de archivo, elija Configuración del formato del archivo de locución, seleccione el formato de archivo que desee y, a continuación, elija Descargar.

Repita los pasos anteriores, pero elija otro idioma y observe la diferencia en la salida.

AWS CLI

En una llamada a `SynthesizeSpeech`, puede especificar varios lexicones. En este caso, el primer lexicón especificado (de izquierda a derecha) anulará cualquier lexicón anterior.

Pongamos como ejemplo los dos lexicones siguientes. Tenga en cuenta que cada lexicón especifica alias diferentes para el mismo grafema, W3C.

- Lexicón 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>
```

- Lexicón 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>
```

Suponga que almacena estos lexicones como `w3c` y `w3cAlternate`, respectivamente. Si especifica los lexicones en orden (`w3c` seguido de `w3cAlternate`) en una llamada a `SynthesizeSpeech`, el alias de W3C definido en el primer lexicón tendrá prioridad sobre el segundo. Para probar los lexicones, siga estos pasos:

1. Guarde los lexicones localmente en sendos archivos denominados `w3c.pls` y `w3cAlternate.pls`.
2. Cargue estos léxicos mediante el `put-lexicon` AWS CLI comando.
   - Cargue el lexicón `w3c.pls` y guárdelo como `w3c`.

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

   - Cargue el lexicón `w3cAlternate.pls` en el servicio como `w3cAlternate`.

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

3. Ejecute el comando `synthesize-speech` para sintetizar el texto de ejemplo en una secuencia de audio (`speech.mp3`) y especifique los dos lexicones a través del parámetro `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. Compruebe el archivo `speech.mp3` resultante. Debería decir lo siguiente:

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

Filtrado de la lista de lexicones en la consola

En el siguiente procedimiento se describe cómo filtrar la lista de lexicones para que solamente aparezcan los del idioma elegido.

Console

Para filtrar la lista de lexicones por idioma

1. Inicia sesión en la consola Amazon Polly AWS Management Console y ábrela en. https://console.aws.amazon.com/polly/
2. Elija la pestaña Lexicons.
3. Elija Cualquier idioma.
4. En la lista de idiomas, seleccione el idioma que desea utilizar como filtro.

   En la lista solo aparecerán los lexicones del idioma seleccionado.

AWS CLI

Amazon Polly proporciona la operación de ListLexicons API que puede usar para obtener la lista de léxicos de pronunciación de su cuenta en una región específica. AWS La siguiente AWS CLI llamada muestra los léxicos de su cuenta en la región us-east-2.

```
aws polly list-lexicons
```

A continuación se incluye un ejemplo de respuesta, donde se muestran dos lexicones denominados `w3c` y `tomato`. En cada lexicón, la respuesta devuelve metadatos; por ejemplo, el código de idioma al que se aplica, el número de lexemas definidos en el lexicón, el tamaño en bytes, etc. El código de idioma indica el idioma y la configuración regional a los que se aplican los lexemas definidos en el lexicón.

```
{
    "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"
        }
    ]
}
```

Los siguientes recursos contienen información adicional sobre la operación: ListLexicons

- Muestra de Java: ListLexicons
- Muestra de Python (Boto3): ListLexicon

Descarga de lexicones en la consola

En el proceso siguiente se describe cómo puede descargar uno o varios lexicones. Puede añadir, eliminar o modificar las entradas de léxico del archivo y, a continuación, volver a cargarlo para conservar su léxico up-to-date.

Console

Para descargar uno o varios lexicones

1. Inicia sesión en la consola Amazon Polly AWS Management Console y ábrela en. https://console.aws.amazon.com/polly/
2. Elija la pestaña Lexicons.
3. Elija el lexicón o lexicones que desee descargar.

a. Para descargar un único lexicón, elija el nombre en la lista.

b. Para descargar varios lexicones en un único archivo comprimido, active la casilla situada junto a cada entrada de la lista que desee descargar.

4. Elija Descargar.
5. Abra la carpeta en la que desea descargar el lexicón.
6. Seleccione Guardar.

AWS CLI

Amazon Polly proporciona la operación de la API GetLexicon para recuperar el contenido de un lexicón de pronunciación guardado en su cuenta en una región específica.

El siguiente `get-lexicon` AWS CLI comando recupera el contenido del léxico. `example`

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

Si aún no tiene un lexicón almacenado en su cuenta, puede utilizar la operación `PutLexicon` para hacerlo. Para obtener más información, consulte Carga de un lexicón.

A continuación, se muestra una respuesta de ejemplo. Además del contenido del lexicón, la respuesta devuelve los metadatos, como el código de idioma que se aplica al lexicón, el número de lexemas definidos en el lexicón, el Nombre de recurso de Amazon (ARN) del recurso y el tamaño del lexicón en bytes. El valor de `LastModified` es una marca temporal de 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
    }
```

```
}
```

Los siguientes recursos contienen ejemplos de código adicionales para la GetLexicon operación:

- Muestra de Java: GetLexicon
- Muestra de Python (Boto3): GetLexicon

Eliminación de un lexicón

En el siguiente proceso se describe cómo se elimina un lexicón. Después de eliminar un lexicón, deberá volver a agregarlo para poder utilizarlo de nuevo. Puede eliminar uno o varios lexicones al mismo tiempo. Para ello, active las casillas situadas junto a cada lexicón.

Console

Eliminación de un lexicón

1. Inicia sesión en la consola Amazon Polly AWS Management Console y ábrela en. https://console.aws.amazon.com/polly/
2. Elija la pestaña Lexicons.
3. Seleccione en la lista uno o varios lexicones que desee eliminar.
4. Elija Eliminar.
5. Elija Eliminar para eliminar el lexicón de la región o Cancelar para conservarlo.

AWS CLI

Amazon Polly proporciona la operación de DeleteLexicon API para eliminar un léxico de pronunciación de una AWS región específica de su cuenta. Lo siguiente AWS CLI elimina el léxico especificado.

El siguiente AWS CLI ejemplo está formateado para Unix, Linux y macOS. En Windows, sustituya la barra invertida (\) del carácter de continuación de Unix al final de cada línea por un signo de intercalación (^) y utilice comillas completas (") alrededor del texto introducido con comillas simples (') para las etiquetas interiores.

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

Los siguientes recursos contienen información adicional para la DeleteLexicon operación:

- Muestra de Java: DeleteLexicon
- Muestra de Python (Boto3): DeleteLexicon

Archivo de audio largos

Para crear archivos TTS para fragmentos de texto grandes, utilice la funcionalidad síntesis asíncrona de Amazon Polly. Esto usa los tres `SpeechSynthesisTask` APIs:

- `StartSpeechSynthesisTask`: inicia una tarea de síntesis nueva.
- `GetSpeechSynthesisTask`: devuelve detalles acerca de una tarea de síntesis enviada con anterioridad.
- `ListSpeechSynthesisTasks`: enumera todas las tareas de síntesis enviadas.

La operación `SynthesizeSpeech` produce audio casi en tiempo real, con relativamente poca latencia en la mayoría de los casos. Para ello, la operación solo puede sintetizar 3000 caracteres.

La característica de síntesis asíncrona de Amazon Polly supera el desafío que supone el procesamiento de un documento de texto de mayor tamaño cambiando la manera en la que el documento se sintetiza y se devuelve. Cuando se realiza una solicitud de síntesis enviando texto de entrada con `StartSpeechSynthesisTask`, Amazon Polly pone en cola las solicitudes y, a continuación, las procesa de forma asíncrona en segundo plano tan pronto como el sistema cuenta con recursos disponibles. Amazon Polly carga a continuación el flujo de voz o de marcas de voz resultante directamente a su bucket de Amazon Simple Storage Service (Amazon S3) (obligatorio) y le informa acerca de la disponibilidad del archivo completado a través de su tema de SNS (opcional).

De esta forma, toda la funcionalidad, excepto el procesamiento casi en tiempo real está disponible para textos de hasta 100 000 caracteres facturables (o 200 000 caracteres totales) de longitud.

Para sintetizar un documento mediante este método, debe disponer de un bucket de Amazon S3 en el que pueda escribir donde pueda guardar el archivo de audio. Puede recibir una notificación cada vez que la síntesis de audio esté lista proporcionando un identificador de tema de SNS opcional. Cuando la tarea de síntesis se haya completado, Amazon Polly publicará un mensaje sobre ese tema. Este mensaje también podría contener información de errores útil en aquellos casos en los que la tarea de síntesis no se realice correctamente. Para ello, asegúrese de que el usuario que crea la tarea de síntesis pueda publicar también en el tema de SNS. Consulte la documentación de Amazon SNS para obtener más información acerca de cómo crear y suscribirse a un tema de SNS.

Cifrado

Puede almacenar el archivo de salida en formato cifrado en el bucket de S3 si lo desea. Para ello, habilite cifrado de bucket de Amazon S3, que utiliza uno de los cifrados de bloques más seguros disponibles Advanced Encryption Standard de 256 bits (AES-256).

Temas

- Configuración de la política de IAM para síntesis asíncrona
- Creación de archivos de audio largos

Configuración de la política de IAM para síntesis asíncrona

Para utilizar la funcionalidad de síntesis asíncrona, necesitará una política de IAM que permita lo siguiente:

- uso de nuevas operaciones de Amazon Polly
- escritura en el bucket de S3 de salida
- publicar en el tema de SNS de estado [opcional]

La siguiente política concede únicamente los permisos necesarios para síntesis asíncrona y se pueden adjuntar al usuario de IAM.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "polly:StartSpeechSynthesisTask",
        "polly:GetSpeechSynthesisTask",
        "polly:ListSpeechSynthesisTasks"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::bucket-name/*"
    },
    {
      "Effect": "Allow",
```

```
      "Action": "sns:Publish",
      "Resource": "arn:aws:sns:region:account:topic"
    }
  ]
}
```

Creación de archivos de audio largos

Puede utilizar la consola de Amazon Polly para crear fragmentos de voz prolongados mediante la síntesis asíncrona con la misma funcionalidad que puede utilizar con la AWS CLI. Esto se realiza mediante la pestaña Text-to-Speech (Texto a voz) de forma muy similar a cualquier otra síntesis.

Console

Las demás funcionalidades de síntesis asíncrona también se encuentran disponibles a través de la consola. La pestaña S3 synthesis tasks (Tareas de síntesis de S3) refleja la funcionalidad `ListSpeechSynthesisTasks`, que muestra todas las tareas guardadas en el bucket de S3 y le permite filtrarlas si lo desea. Al hacer clic en una única tarea específica muestra sus detalles, de tal forma que refleja la funcionalidad `GetSpeechSynthesisTask`.

Síntesis de un texto grande con la consola de Amazon Polly

1. Inicia sesión en la consola Amazon Polly AWS Management Console y ábrela en. https://console.aws.amazon.com/polly/
2. Elija la pestaña Text-to-Speech. Seleccione Formato largo como motor, si corresponde.
3. Con SSML encendido o apagado, escriba o pegue el texto en el cuadro de entrada.
4. Elija el idioma, la región y la voz para el texto.
5. Seleccione Guardar en S3.

> **Note**
>
> Tanto las opciones Descargar como Escuchar aparecerán atenuadas si la longitud del texto es superior al límite de 3000 caracteres para la operación `SynthesizeSpeech` en tiempo real.

6. La consola abre un formulario para que pueda elegir dónde almacenar el archivo de salida.

   a. Rellene el nombre del bucket de Amazon S3 de destino.

b. Opcionalmente, rellene la clave de prefijo de la salida.

> **Note**
>
> El bucket de S3 de salida debe permitir la escritura.

c. Si desea recibir una notificación cuando la tarea de síntesis se haya completado, proporcione el identificador de tema de SNS opcional.

> **Note**
>
> El SNS debe estar abierto para la publicación por el usuario de consola actual para utilizar esta opción. Para obtener más información, consulte Amazon Simple Notification Service (SNS)

d. Seleccione Guardar en S3.

Para recuperar información sobre sus tareas de síntesis de voz

1. En la consola, elija la pestaña Tareas de síntesis de S3.
2. Las tareas se muestran por orden de fecha. Para filtrar las tareas por estado, elija Todos los estados y, a continuación, elija el estado que desee usar.
3. Para ver los detalles de una tarea en particular, seleccione el Task ID (ID de tarea) vinculado.

AWS CLI

La funcionalidad de síntesis asíncrona de Amazon Polly utiliza tres `SpeechSynthesisTask` APIs para trabajar con grandes cantidades de texto:

- `StartSpeechSynthesisTask`: inicia una tarea de síntesis nueva.
- `GetSpeechSynthesisTask`: devuelve detalles acerca de una tarea de síntesis enviada con anterioridad.
- `ListSpeechSynthesisTasks`: enumera todas las tareas de síntesis enviadas.

Síntesis de grandes cantidades de texto (**`StartSpeechSynthesisTask`**)

Cuando desea crear un archivo de audio mayor de lo que puede crear con `SynthesizeSpeech` en tiempo real, utilice la operación `StartSpeechSynthesisTask`. Además de los argumentos necesarios para la operación `SynthesizeSpeech`, `StartSpeechSynthesisTask` también requiere el nombre de un bucket de Amazon S3. También se dispone de otros dos otros argumentos opcionales: un prefijo de clave para el archivo de salida y el ARN de un tema de SNS si desea recibir notificaciones de estado sobre la tarea.

- `OutputS3BucketName`: el nombre del bucket de Amazon S3; en el que debe cargarse la síntesis. Este bucket debe estar en la misma región que el servicio de Amazon Polly. Además, el usuario de IAM que se utiliza para realizar la llamada debe tener acceso al bucket. [Obligatorio]
- `OutputS3KeyPrefix`: prefijo de clave para el archivo de salida. Utilice este parámetro si desea guardar el archivo de voz de salida en una clave similar a un directorio personalizado en su bucket. [Opcional]
- `SnsTopicArn`: el ARN de tema de SNS que debe utilizar si desea recibir notificaciones acerca del estado de la tarea. Este tema de SNS debe estar en la misma región que el servicio de Amazon Polly. Además, el usuario de IAM que se utiliza para realizar la llamada debe tener acceso al tema. [Opcional]

Por ejemplo, el siguiente ejemplo se puede utilizar para ejecutar el `start-speech-synthesis-task` AWS CLI comando en la región EE.UU. Este (Ohio):

El siguiente AWS CLI ejemplo está formateado para Unix, Linux y macOS. En Windows, sustituya la barra invertida (\) del carácter de continuación de Unix al final de cada línea por un signo de intercalación (^) y utilice comillas completas (") alrededor del texto introducido con comillas simples (') para las etiquetas interiores.

```
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
```

Este comando producirá una respuesta con un aspecto similar al siguiente:

```
"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"
 }
```

La operación `start-speech-synthesis-task` devuelve varios campos nuevos:

- `OutputUri`: la ubicación del archivo de voz de salida.
- `TaskId`: un identificador único para la tarea de síntesis de voz generada por Amazon Polly.
- `CreationTime`: una marca temporal de cuando se envió inicialmente la tarea.
- `RequestCharacters`: el número de caracteres facturables en la tarea.
- `TaskStatus`: proporciona información sobre el estado de la tarea enviada.

  Cuando se envía la tarea, el estado inicial mostrará `scheduled`. Cuando Amazon Polly inicia el procesamiento de la tarea, el estado cambiará a `inProgress` y posteriormente, a `completed` o `failed`. Si la tarea falla, se devolverá un mensaje de error al llamar a la operación GetSpeechSynthesisTask o ListSpeechSynthesisTasks .

Cuando la tarea se complete, el archivo de voz está disponible en la ubicación especificada en `OutputUri`.

Recuperación información sobre la tarea de síntesis de voz

Puede obtener información acerca de una tarea como, por ejemplo, errores, estado, etc., utilizando la operación `GetSpeechSynthesisTask`. Para ello, necesitará el `task-id` devuelto por el `StartSpeechSynthesisTask`.

Por ejemplo, se puede usar el siguiente ejemplo para ejecutar el `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
```

También puede mostrar todas las tareas de síntesis de voz que haya ejecutado en la región actual mediante la operación `ListSpeechSynthesisTasks`.

Por ejemplo, se puede utilizar el siguiente ejemplo para ejecutar el `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/"
```

Cuotas en Amazon Polly

Amazon Polly aplica cuotas al tráfico de clientes al rechazar solicitudes excesivas. La cuota predeterminada para las solicitudes `SynthesizeSpeech` con voces estándar es de 80 transacciones por segundo (tps), en una sola región, para una sola Cuenta de AWS. Si los límites no aumentaran y generara 100 solicitudes `SynthesizeSpeech` por segundo con una voz estándar, 80 solicitudes por segundo se ejecutarían correctamente y Amazon Polly limitaría 20 solicitudes por segundo. Estas solicitudes devolverían una respuesta con el estado HTTP 400 y un encabezado de respuesta que indicara `ThrottlingException`. Amazon Polly también limita el tráfico a todas las operaciones en función de la tasa de solicitudes.

Ejemplos de límites de síntesis de voz

- Sintetizar las primeras 24 letras del alfabeto inglés letra por letra. Si la síntesis de cada letra tardara menos de 50 milisegundos, con un límite de operación de ocho tps, sintetizar 24 letras tardaría al menos tres segundos. Durante ese tiempo, podría sintetizar hasta ocho letras por segundo. Cualquier otra solicitud se limitaría. Como las solicitudes duran poco tiempo, se sintetizarían en serie sin superponerse.
- Sintetizar 16 párrafos de texto. Si cada párrafo se sintetizara y el cliente lo recibiera en su totalidad en dos segundos o menos, con un límite de operación de ocho solicitudes simultáneas, se necesitarían al menos cuatro segundos para sintetizar los 16 artículos. En el primer segundo, podría iniciar hasta ocho solicitudes. Durante las solicitudes simultáneas, cualquier intento de iniciar una nueva síntesis se vería limitado debido al límite de simultaneidad. Puede sintetizar los ocho párrafos restantes después de los dos primeros segundos, una vez finalizado el primer lote de solicitudes.

Tenga en cuenta las siguientes limitaciones al utilizar Amazon Polly.

Temas

- Regiones de admitidas
- Cuotas y tasas de limitación
- Lexicones de pronunciación
- SynthesizeSpeech Operaciones de API
- SpeechSynthesisTask Operaciones de API
- Speech Synthesis Markup Language (SSML, idioma de marcado de síntesis de voz)

Regiones de admitidas

Para obtener una lista de AWS las regiones en las que Amazon Polly está disponible, consulte Amazon Polly Endpoints and Quotas en. Referencia general de Amazon Web Services

- Para ver las regiones que admiten voces generativas, consulte Voces generativas.
- Para ver las regiones que admiten voces de formato largo, consulte Voces de formato largo.
- Para ver las regiones que admiten las voces neuronales, consulte the section called "Compatibilidad con características y regiones" para TTS neuronal.

Cuotas y tasas de limitación

En la siguiente tabla se definen las tasas de limitación por operación de Amazon Polly. Puede utilizarla AWS Management Console para solicitar aumentos de cuota para las cuotas ajustables cuando sea necesario.

Operación	Límite
Lexicon	

`DeleteLexicon`
`PutLexicon`
`GetLexicon`
`ListLexicons`
	2 transacciones por segundo (tps) cualesquiera de estas operaciones combinados.
	Ráfaga máxima permitida de 4 tps.

Speech	
`DescribeVoices`	80 tps con un límite de ráfaga de 100 tps
`SynthesizeSpeech`	Voz generativa: 8 tps Voz de formato largo: 8 tps con un límite de ráfaga de 10 tps Voz neuronal: 8 tps con un límite de ráfaga de 10 tps Voz estándar: 80 tps con un límite de ráfaga de 100 tps

Operación	Límite
`StartSpeechSynthesisTask`	Voz generativa: 1 tps Voz de formato largo: 1 tps Voz neuronal: 10 tps Voz estándar: 10 tps con un límite de ráfaga de 12 tps
`GetSynthesizeSpeechTask` y `ListSynthesizeSpeechTask`	Máximo permitido de 10 tps combinadas

Solicitudes simultáneas

Para la voz generativa, Amazon Polly admite hasta 26 solicitudes simultáneas. Para la voz de formato largo, Amazon Polly admite hasta 26 solicitudes simultáneas. En el caso de la voz neuronal, Amazon Polly admite 8 tps con un límite de ráfaga de 10 tps, para un máximo de 18 solicitudes simultáneas. Amazon Polly también admite límites para las solicitudes simultáneas. En el caso de la voz estándar, Amazon Polly admite 80 tps para un máximo de 80 solicitudes simultáneas.

Mejores prácticas para mitigar la limitación

- Vuelva a instalar las limitaciones en función de los tiempos de retardo y fluctuación de fase para poder repartir la carga en un breve período de tiempo y gestionar los picos de uso inesperados sin comprometer la disponibilidad. AWS Code Sample Catalog ya está configurado para hacerlo de forma predeterminada en muchos idiomas de programación. Consulte Comportamiento de los reintentos para ver los detalles.
- Usar las métricas de Amazon Polly. Amazon Polly publica automáticamente CloudWatch para analizar su uso actual y pronosticar su crecimiento.

> **Note**
>
> Antes de solicitar un aumento de cuota (si procede), calcule sus necesidades de tps siguiendo las pautas de esta página. Amazon Polly asegura solo los recursos

computacionales necesarios de acuerdo con la demanda de los clientes para mantener sus costos bajos.

Lexicones de pronunciación

- Puede almacenar hasta 100 lexicones por cuenta.
- Los nombres de los lexicones pueden contener caracteres alfanuméricos y tener una longitud máxima de 20 caracteres.
- Cada lexicón puede tener un tamaño máximo de 40 000 caracteres. (Tenga en cuenta que el tamaño del léxico afecta a la latencia de la SynthesizeSpeech operación).
- Puede especificar un máximo de 100 caracteres para cada sustitución de <phoneme> o <alias> en un lexicón.

Para obtener más información sobre el uso de los lexicones, consulte Administración de lexicones.

SynthesizeSpeech Operaciones de API

Al estimar el uso de `SynthesizeSpeech`, tenga en cuenta que el audio producido por Amazon Polly, especialmente para aplicaciones interactivas, suele tardar al menos varios segundos en reproducirse. Esto reduce la tasa de solicitudes para `SynthesizeSpeech`, incluso para un gran número de consumidores simultáneos. Además, Amazon Polly limita las solicitudes `SynthesizeSpeech` en función del número de solicitudes simultáneas que sintetiza. No existe una configuración independiente para las solicitudes simultáneas. El límite de solicitudes simultáneas siempre tiene el mismo valor que el número de puntos por segundo permitido y se escala con él.

Ejemplo de aplicación de historia corta. Puede usar Amazon Polly para crear una aplicación que reproduzca una serie de historias cortas. Con este tipo de aplicaciones, se empezaría a reproducir la primera historia, luego la siguiente, y así sucesivamente, hasta que el usuario abandonara la aplicación. Cada historia tardaría unos 0,5 segundos en sintetizarse y 10 segundos en reproducirse. En este escenario, cabría esperar que se produjera una llamada a `SynthesizeSpeech` por cada 10 segundos que el cliente dedicara a utilizar la aplicación. Esto se traduciría en una llamada por segundo por cada 10 clientes que utilizaran la aplicación simultáneamente. Si tuviera 1000 clientes que utilizaran la aplicación al mismo tiempo, podría esperar que la tarifa media de llamadas a `SynthesizeSpeech` fuera de solo 100 transacciones por segundo.

Tenga en cuenta los siguientes límites relacionados con el uso de la operación de la API `SynthesizeSpeech`:

- El texto de entrada debe tener un tamaño máximo de 3000 caracteres contabilizados (6000 caracteres totales). Las etiquetas SSML no cuentan como caracteres contabilizados.
- Puede especificar hasta cinco lexicones para aplicarlos el texto de entrada.
- La secuencia de audio de salida (síntesis) tiene un límite de 10 minutos. Después de esto, se interrumpe cualquier fragmento de voz restante.

Para obtener más información, consulte SynthesizeSpeech.

> **Note**
>
> Puede utilizar la operación de la API `StartSythensizeSpeechTask` para evitar algunas de las limitaciones de la operación `SynthesizeSpeech` de la API. Para obtener más información, consulte Archivo de audio largos.

SpeechSynthesisTask Operaciones de API

Tenga en cuenta los siguientes límites relacionados con el uso de las operaciones de la API `StartSpeechSynthesisTask`, `GetSpeechSynthesisTask` y `ListSpeechSynthesisTasks`:

- El texto de entrada debe tener un tamaño máximo de 100,000 caracteres contabilizados (200,000 caracteres totales). Las etiquetas SSML no cuentan como caracteres contabilizados.
- Puede especificar hasta cinco lexicones para aplicarlos el texto de entrada.

Speech Synthesis Markup Language (SSML, idioma de marcado de síntesis de voz)

Tenga en cuenta los siguientes límites relacionados con el uso de SSML:

- No se admiten las etiquetas `<audio>`, `<lexicon>`, `<lookup>` y `<voice>`.
- Los elementos `<break>` pueden especificar una duración máxima de 10 segundos cada uno.
- La etiqueta `<prosody>` no admite valores inferiores a -80 % en el atributo de velocidad.

Para obtener más información, consulte Generación de fragmentos hablados desde documentos SSML.

Ejemplos de código y aplicaciones para Amazon Polly

En esta sección encontrará código de muestra y aplicaciones de ejemplo que puede utilizar para explorar Amazon Polly.

El tema Código de muestra contiene fragmentos de código organizados por lenguaje de programación y divididos en ejemplos para las distintas funcionalidades de Amazon Polly. El tema Aplicaciones de ejemplo contiene aplicaciones organizadas por lenguaje de programación que se pueden utilizar de forma independiente para explorar Amazon Polly.

Antes de comenzar a usar estos ejemplos, le recomendamos que primero lea Cómo funciona Amazon Polly y siga los pasos que se indican en Introducción a Amazon Polly.

Temas

- Muestras de Java
- Muestras de Python
- Ejemplo de Java
- Ejemplo de Python (HTML5 cliente y servidor Python)
- Ejemplo de iOS
- Ejemplo de Android

Muestras de Java

Las siguientes muestras de código explican cómo utilizar aplicaciones basadas en Java para realizar diversas tareas con Amazon Polly. Estas muestras no son ejemplos completos, pero se pueden incluir en aplicaciones de Java de mayor tamaño que utilicen el AWS SDK para Java.

Fragmentos de código

- DeleteLexicon
- DescribeVoices
- GetLexicon
- ListLexicons
- PutLexicon
- StartSpeechSynthesisTask

- Marcas de voz
- SynthesizeSpeech

DeleteLexicon

El siguiente ejemplo de código de Java muestra cómo utilizar aplicaciones basadas en Java para eliminar un léxico específico almacenado en una región. AWS Un léxico que se ha eliminado no está disponible para la síntesis de voz ni se puede recuperar con la tecla o. `GetLexicon` `ListLexicon` APIs

Para obtener más información sobre esta operación, consulte la referencia de la 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

El siguiente ejemplo de código Java muestra cómo utilizar aplicaciones basadas en Java para generar una lista de las voces que están disponibles para su uso cuando se solicita la síntesis de voz. También puede especificar un código de idioma para filtrar las voces disponibles. Por ejemplo, si especifica en-US, la operación devuelve una lista de todas las voces disponibles para el idioma inglés de EE. UU.

Para obtener más información sobre esta operación, consulte la referencia de la 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

En el siguiente ejemplo de código de Java, se muestra cómo utilizar aplicaciones basadas en Java para producir el contenido de un léxico de pronunciación específico almacenado en una región. AWS

Para obtener más información sobre esta operación, consulte la referencia de la 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

En el siguiente ejemplo de código de Java, se muestra cómo utilizar aplicaciones basadas en Java para crear una lista de léxicos de pronunciación almacenados en una región. AWS

Para obtener más información sobre esta operación, consulte la referencia de la 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

En el siguiente ejemplo de código de Java, se muestra cómo utilizar aplicaciones basadas en Java para almacenar un léxico de pronunciación en una región. AWS

Para obtener más información sobre esta operación, consulte la referencia de la 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

El siguiente ejemplo de código Java muestra cómo utilizar aplicaciones basadas en Java para sintetizar un fragmento de voz largo (hasta 100 000 caracteres contabilizados) y almacenarlo directamente en un bucket de Amazon S3.

Para obtener más información, consulte la referencia de la API de 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();
    }

}
```

Marcas de voz

El siguiente ejemplo de código muestra cómo utilizar aplicaciones basadas en Java para sintetizar marcas de voz para el texto introducido. Esta funcionalidad usa la SynthesizeSpeech API.

Para obtener más información sobre esta funcionalidad, consulte Marcas de voz.

Para obtener más información sobre la API, consulte la referencia de la API de 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

El siguiente ejemplo de código Java muestra cómo utilizar aplicaciones basadas en Java para sintetizar voz con textos más cortos para procesamiento casi en tiempo real.

Para obtener más información, consulte la referencia de la API de 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);
        }
    }
}
```

Muestras de Python

Los siguientes ejemplos de código muestran cómo utilizar aplicaciones basadas en Python (boto3) para realizar diversas tareas con Amazon Polly. Estas muestras no pretenden ser ejemplos completos, pero se pueden incluir en aplicaciones de Python de mayor tamaño que utilicen el AWS SDK for Python (Boto).

Fragmentos de código

- DeleteLexicon
- GetLexicon
- ListLexicon
- PutLexicon
- StartSpeechSynthesisTask
- SynthesizeSpeech

DeleteLexicon

El siguiente ejemplo de código de Python utiliza el AWS SDK for Python (Boto) para eliminar un léxico en la región especificada en la AWS configuración local. El ejemplo solamente elimina el lexicón especificado. Antes de eliminar el lexicón, se le preguntará si desea continuar con la operación.

El siguiente ejemplo de código usa las credenciales predeterminadas almacenadas en el archivo de configuración del AWS SDK. Para obtener más información acerca de cómo crear el archivo de configuración, consulte Configuración del AWS CLI.

Para obtener más información sobre esta operación, consulte la referencia de la 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

El siguiente código de Python utiliza el AWS SDK for Python (Boto) para recuperar todos los léxicos almacenados en una AWS región. El ejemplo acepta el nombre del lexicón como parámetro de línea de comandos y recupera solamente el lexicón, por lo que imprime la ruta de acceso temporal donde se guardó localmente.

El siguiente ejemplo de código usa las credenciales predeterminadas almacenadas en el archivo de configuración del AWS SDK. Para obtener más información acerca de cómo crear el archivo de configuración, consulte Configuración del AWS CLI.

Para obtener más información sobre esta operación, consulte la referencia de la 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

El siguiente ejemplo de código de Python usa el AWS SDK for Python (Boto) para enumerar los léxicos de su cuenta en la región especificada en su AWS configuración local. Para obtener más información acerca de cómo crear el archivo de configuración, consulte Configuración del AWS CLI.

Para obtener más información sobre esta operación, consulte la referencia de la 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

El siguiente ejemplo de código muestra cómo utilizar aplicaciones basadas en Python (boto3) para almacenar un léxico de pronunciación en una región. AWS

Para obtener más información sobre esta operación, consulte la referencia de la API `PutLexicon`.

Tenga en cuenta lo siguiente:

- Debe actualizar el código proporcionando el nombre de archivo del lexicón local y el nombre del lexicón guardado.
- En el ejemplo se presupone que los archivos del lexicón se han creado en un subdirectorio denominado `pls`. Deberá que actualizar la ruta como corresponda.

En el siguiente ejemplo de código, se utilizan las credenciales predeterminadas almacenadas en el archivo de configuración del AWS SDK. Para obtener más información acerca de cómo crear el archivo de configuración, consulte Configuración del AWS CLI.

Para obtener más información sobre esta operación, consulte la referencia de la 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

El siguiente ejemplo de código de Python usa el AWS SDK for Python (Boto) para enumerar los léxicos de su cuenta en la región especificada en su AWS configuración local. Para obtener más información acerca de cómo crear el archivo de configuración, consulte Configuración del AWS CLI.

Para obtener más información, consulte la referencia de la API de `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

El siguiente ejemplo de código de Python utiliza la AWS SDK for Python (Boto) síntesis de voz con textos más cortos para un procesamiento casi en tiempo real. Para obtener más información, consulte la referencia de la operación SynthesizeSpeech.

En este ejemplo, se utiliza una cadena corta de texto sin formato. Puede usar texto SSML para tener más control sobre el resultado. Para obtener más información, consulte Generación de fragmentos hablados desde documentos 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()
```

Ejemplo de Java

En este ejemplo se muestra cómo utilizar Amazon Polly para transmitir un fragmento hablado desde una aplicación basada en Java. En este ejemplo se utiliza AWS SDK para Java para leer el texto especificado con una voz seleccionada en una lista.

El código mostrado incluye la mayoría de las tareas, pero la comprobación de errores es mínima. Si Amazon Polly encuentra un error, la aplicación termina.

Para ejecutar esta aplicación de ejemplo, necesita lo siguiente:

- Java 8 Java Development Kit (JDK)
- AWS SDK para Java
- Apache Maven

Para probar la aplicación

1. Asegúrese de que la variable de entorno JAVA_HOME está configurada para JDK.

   Por ejemplo, si JDK 1.8.0_121 está instalado en Windows en la carpeta `C:\Program Files\Java\jdk1.8.0_121`, tendría que especificar lo siguiente en el símbolo del sistema:

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

   Si JDK 1.8.0_121 está instalado en Linux en la carpeta `/usr/lib/jvm/java8-openjdk-amd64` , tendría que especificar lo siguiente en el símbolo del sistema:

   ```
   export JAVA_HOME=/usr/lib/jvm/java8-openjdk-amd64
   ```

2. Configure las variables de entorno de Maven para ejecutar Maven desde la línea de comandos.

   Por ejemplo, si Maven 3.3.9 está instalado en Windows en la carpeta `C:\Program Files\apache-maven-3.3.9`, tendría que especificar lo siguiente:

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

   Si Maven 3.3.9 está instalado en Linux en la carpeta `/home/ec2-user/opt/apache-maven-3.3.9`, tendría que especificar lo siguiente:

   ```
   export M2_HOME=/home/ec2-user/opt/apache-maven-3.3.9
   export M2=$M2_HOME/bin
   export PATH=$M2:$PATH
   ```

3. Cree un nuevo directorio llamado `polly-java-demo`.
4. En el directorio `polly-java-demo`, cree un archivo llamado `pom.xml` y pegue el siguiente código en él:

```
<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. Cree un nuevo directorio llamado `polly` en `src/main/java/com/amazonaws/demos`.
6. En el directorio `polly`, cree un nuevo archivo de origen de Java llamado `PollyDemo.java` y pegue el siguiente código:

```
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();

  }
}
```

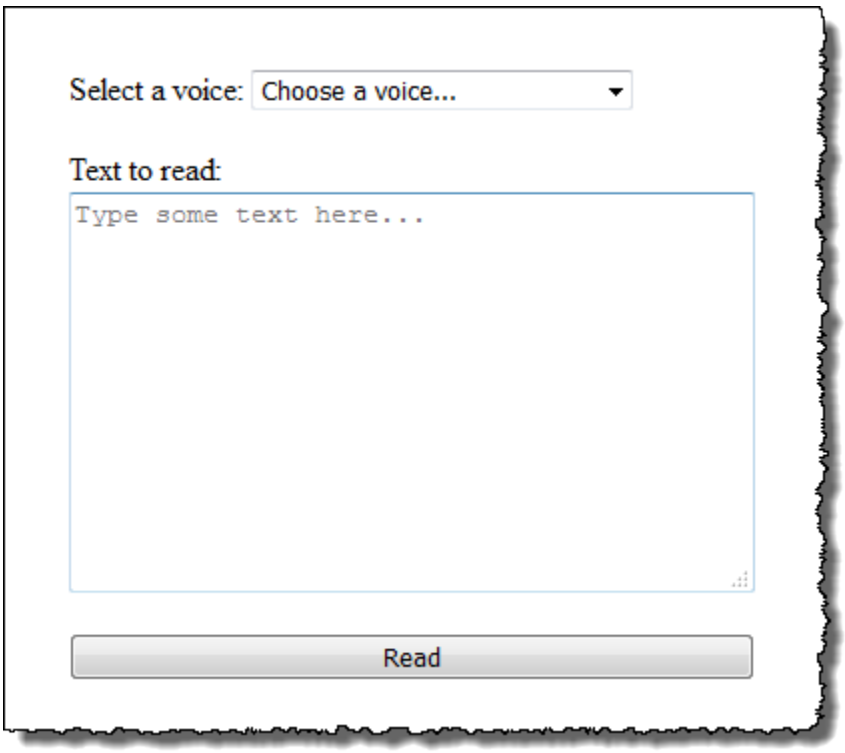
7. Vuelva al directorio `polly-java-demo` para limpiar, compilar y ejecutar la demostración:

```
mvn clean compile exec:java
```

Ejemplo de Python (HTML5 cliente y servidor Python)

Esta aplicación de ejemplo se compone de lo siguiente:

- Un servidor HTTP 1.1 que utiliza la codificación de transferencia en bloques de HTTP (consulte el siguiente tema sobre la codificación de transferencia en bloques)
- Una interfaz HTML5 de usuario sencilla que interactúa con el servidor HTTP 1.1 (se muestra a continuación):



El objetivo de este ejemplo es mostrar cómo usar Amazon Polly para transmitir voz desde una aplicación basada en un navegador HTML5 . En los casos de uso en los que la capacidad de respuesta resulta un factor importante (por ejemplo, en sistemas de diálogo, lectores de pantalla, etc.), es recomendable emplear la secuencia de audio producida por Amazon Polly a medida que el texto se va sintetizando.

Para ejecutar esta aplicación de ejemplo, necesita lo siguiente:

- Navegador web compatible con los estándares HTML5 2 y EcmaScript 5 (por ejemplo, Chrome 23.0 o superior, Firefox 21.0 o superior, Internet Explorer 9.0 o superior)
- Una versión de Python posterior a la 3.0

Para probar la aplicación

1. Guarde el código del servidor como `server.py`. Para obtener el código, consulte Ejemplo de Python: código de servidor de Python (server.py).
2. Guarde el código de HTML5 cliente como. `index.html` Para obtener el código, consulte Ejemplo de Python: interfaz de HTML5 usuario (index.html).

3. Ejecute el comando siguiente desde la ruta en la que guardó server.py para iniciar la aplicación (en algunos sistemas, tal vez tenga que usar `python3` en lugar de `python` al ejecutar el comando).

   ```
   $ python  server.py
   ```

   Una vez que se inicia la aplicación, aparece una dirección URL en el terminal.

4. Abra la dirección URL que aparece en el terminal del navegador web.

   Puede pasar a `server.py` la dirección y el puerto del servidor de la aplicación para que se utilice como parámetro. Para obtener más información, ejecute `python server.py -h`.

5. Para escuchar el fragmento hablado, elija una voz de la lista, escriba un texto y seleccione Read. Tan pronto como Amazon Polly transfiera el primer fragmento de datos de audio utilizable, comenzará a reproducirse el discurso sintetizado.

6. Para detener el servidor de Python cuando termine de probar la aplicación, pulse Ctrl+C en el terminal en el que se está ejecutando el servidor.

> **Note**
>
> El servidor crea un cliente Boto3 utilizando AWS SDK for Python (Boto). El cliente utiliza las credenciales almacenadas en el archivo de AWS configuración de su ordenador para firmar y autenticar las solicitudes a Amazon Polly. Para obtener más información sobre cómo crear el archivo de AWS configuración y almacenar las credenciales, consulte Configuración del archivo AWS Command Line Interface en la Guía del AWS Command Line Interface usuario.

Ejemplo de Python: interfaz de HTML5 usuario (index.html)

En esta sección se proporciona el código para el HTML5 cliente descrito enEjemplo de Python (HTML5 cliente y servidor Python).

```
<html>

<head>
    <title>Text-to-Speech Example Application</title>
    <script>
        /*
```

```
         * This sample code requires a web browser with support for both the
         * HTML5 and ECMAScript 5 standards; the following is a non-comprehensive
         * list of compliant browsers and their minimum version:
         *
         * - Chrome 23.0+
         * - Firefox 21.0+
         * - Internet Explorer 9.0+
         * - Edge 12.0+
         * - Opera 15.0+
         * - Safari 6.1+
         * - Android (stock web browser) 4.4+
         * - Chrome for Android 51.0+
         * - Firefox for Android 48.0+
         * - Opera Mobile 37.0+
         * - iOS (Safari Mobile and Chrome) 3.2+
         * - Internet Explorer Mobile 10.0+
         * - Blackberry Browser 10.0+
         */

        // Mapping of the OutputFormat parameter of the SynthesizeSpeech API
        // and the audio format strings understood by the browser
        var AUDIO_FORMATS = {
            'ogg_vorbis': 'audio/ogg',
            'mp3': 'audio/mpeg',
            'pcm': 'audio/wave; codecs=1'
        };

        /**
         * 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>
```

Ejemplo de Python: código de servidor de Python (server.py)

Esta sección contiene el código del servidor de Python que se describe en Ejemplo de Python (HTML5 cliente y servidor Python).

```
"""
Example Python 2.7+/3.3+ Application

This application consists of a HTTP 1.1 server using the HTTP chunked transfer
coding (https://tools.ietf.org/html/rfc2616#section-3.6.1) and a minimal HTML5
user interface that interacts with it.

The goal of this example is to start streaming the speech to the client (the
HTML5 web UI) as soon as the first consumable chunk of speech is returned in
order to start playing the audio as soon as possible.
For use cases where low latency and responsiveness are strong requirements,
this is the recommended approach.

The service documentation contains examples for non-streaming use cases where
waiting for the speech synthesis to complete and fetching the whole audio stream
at once are an option.

To test the application, run 'python server.py' and then open the URL
displayed in the terminal in a web browser (see index.html for a list of
supported browsers). The address and port for the server can be passed as
parameters to server.py. For more information, run: 'python server.py -h'
"""
from argparse import ArgumentParser
from collections import namedtuple
from contextlib import closing
from io import BytesIO
from json import dumps as json_encode
import os
import sys
```

```
if sys.version_info >= (3, 0):
    from http.server import BaseHTTPRequestHandler, HTTPServer
    from socketserver import ThreadingMixIn
    from urllib.parse import parse_qs
else:
    from BaseHTTPServer import BaseHTTPRequestHandler, HTTPServer
    from SocketServer import ThreadingMixIn
    from urlparse import parse_qs

from boto3 import Session
from botocore.exceptions import BotoCoreError, ClientError

ResponseStatus = namedtuple("HTTPStatus",
                            ["code", "message"])

ResponseData = namedtuple("ResponseData",
                          ["status", "content_type", "data_stream"])

# Mapping the output format used in the client to the content type for the
# response
AUDIO_FORMATS = {"ogg_vorbis": "audio/ogg",
                 "mp3": "audio/mpeg",
                 "pcm": "audio/wave; codecs=1"}
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()
```

Ejemplo de iOS

El ejemplo siguiente utiliza el SDK para iOS de Amazon Polly para leer el texto especificado con una voz seleccionada en una lista de voces.

El código que se muestra aquí incluye la mayoría de las tareas, pero no administra errores. Para ver el código completo, consulte la AWS Mobile SDK for iOS demostración de Amazon Polly.

Initialize

```
// 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()
```

Obtener la lista de voces disponibles

```
// 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
})
```

Realizar la síntesis de voz

```
// 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
})
```

Ejemplo de Android

El ejemplo siguiente utiliza el SDK para Android de Amazon Polly para leer el texto especificado con una voz seleccionada en una lista de voces.

El código que se muestra aquí incluye la mayoría de las tareas, pero no administra errores. Para ver el código completo, consulte la AWS Mobile SDK for Android demostración de Amazon Polly.

Initialize

```
// 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);
```

Obtener la lista de voces disponibles

```
// 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();
```

Obtener dirección URL de la secuencia de 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);
```

Reproducir fragmento sintetizado

```
// 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();
    }
});
```

Ejemplos de código para Amazon Polly usando AWS SDKs

Los siguientes ejemplos de código muestran cómo usar Amazon Polly con un kit de desarrollo de AWS software (SDK).

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

Los escenarios son ejemplos de código que muestran cómo llevar a cabo una tarea específica a través de llamadas a varias funciones dentro del servicio o combinado con otros Servicios de AWS.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Polly con un SDK AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Ejemplos de código

- Ejemplos básicos de uso de Amazon Polly AWS SDKs
  - Acciones para usar Amazon Polly AWS SDKs
    - Úselo DeleteLexicon con un AWS SDK o CLI
    - DescribeVoicesÚselo con un AWS SDK
    - Úselo GetLexicon con un AWS SDK o CLI
    - Úselo GetSpeechSynthesisTask con un AWS SDK o CLI
    - Úselo ListLexicons con un AWS SDK o CLI
    - Úselo PutLexicon con un AWS SDK o CLI
    - Úselo StartSpeechSynthesisTask con un AWS SDK o CLI
    - SynthesizeSpeechÚselo con un AWS SDK
- Escenarios para el uso de Amazon Polly AWS SDKs
  - Convierta texto a voz y devuélvalo a texto con un SDK AWS
  - Cree una aplicación de sincronización de labios con Amazon Polly mediante un SDK AWS
  - Creación de una aplicación que analice los comentarios de los clientes y sintetice el audio

Ejemplos básicos de uso de Amazon Polly AWS SDKs

Los siguientes ejemplos de código muestran cómo utilizar los conceptos básicos de Amazon Polly con. AWS SDKs

Ejemplos

- Acciones para usar Amazon Polly AWS SDKs
  - Úselo DeleteLexicon con un AWS SDK o CLI
  - DescribeVoicesÚselo con un AWS SDK
  - Úselo GetLexicon con un AWS SDK o CLI
  - Úselo GetSpeechSynthesisTask con un AWS SDK o CLI
  - Úselo ListLexicons con un AWS SDK o CLI
  - Úselo PutLexicon con un AWS SDK o CLI
  - Úselo StartSpeechSynthesisTask con un AWS SDK o CLI
  - SynthesizeSpeechÚselo con un AWS SDK

Acciones para usar Amazon Polly AWS SDKs

Los siguientes ejemplos de código muestran cómo realizar acciones individuales de Amazon Polly con. AWS SDKs Cada ejemplo incluye un enlace a GitHub, donde puede encontrar instrucciones para configurar y ejecutar el código.

Estos extractos se denominan API Amazon Polly y son extractos de código de programas más grandes que deben ejecutarse en su contexto. Puede ver las acciones en contexto en Escenarios para el uso de Amazon Polly AWS SDKs .

Los siguientes ejemplos incluyen solo las acciones que se utilizan con mayor frecuencia. Para obtener una lista completa, consulte la referencia de la API de Amazon Polly.

Ejemplos

- Úselo DeleteLexicon con un AWS SDK o CLI
- DescribeVoicesÚselo con un AWS SDK
- Úselo GetLexicon con un AWS SDK o CLI
- Úselo GetSpeechSynthesisTask con un AWS SDK o CLI
- Úselo ListLexicons con un AWS SDK o CLI

- Úselo PutLexicon con un AWS SDK o CLI
- Úselo StartSpeechSynthesisTask con un AWS SDK o CLI
- SynthesizeSpeechÚselo con un AWS SDK

Úselo **`DeleteLexicon`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `DeleteLexicon`.

.NET

SDK para .NET

> **Note**
>
> Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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;
        }
    }
```

- Para obtener más información sobre la API, consulta DeleteLexiconla Referencia AWS SDK para .NET de la API.

CLI

AWS CLI

Eliminación de un lexicón

En el siguiente ejemplo de `delete-lexicon` se elimina el lexicón especificado.

```
aws polly delete-lexicon \
```

```
    --name w3c
```

Este comando no genera ninguna salida.

Para obtener más información, consulte Uso de la DeleteLexicon operación en la Guía para desarrolladores de Amazon Polly.

- Para obtener más información sobre la API, consulte DeleteLexiconla Referencia de AWS CLI comandos.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Polly con un SDK AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`DescribeVoices`Úselo con un AWS SDK

En los siguientes ejemplos de código, se muestra cómo utilizar `DescribeVoices`.

.NET

SDK para .NET

> **Note**
>
> Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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}");
        }
    }
```

- Para obtener más información sobre la API, consulta DescribeVoiceslа Referencia AWS SDK para .NET de la API.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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);
        }
    }
}
```

- Para obtener más información sobre la API, consulta DescribeVoicesla Referencia AWS SDK for Java 2.x de la API.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
```

- Para obtener más información sobre la API, consulta DescribeVoicesla AWS Referencia de API de SDK for Python (Boto3).

Ruby

SDK para Ruby

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
```

- Para obtener más información sobre la API, consulta DescribeVoicesla Referencia AWS SDK para Ruby de la API.

Rust

SDK para Rust

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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(())
}
```

- Para obtener más información sobre la API, consulta DescribeVoicesla referencia sobre la API de AWS SDK para Rust.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Polly con un SDK AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`GetLexicon`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `GetLexicon`.

.NET

SDK para .NET

> **Note**
>
> Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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);
            }
        }
    }
```

- Para obtener más información sobre la API, consulta GetLexiconla Referencia AWS SDK para .NET de la API.

CLI

AWS CLI

Recuperación del contenido de un léxico

En el siguiente ejemplo de `get-lexicon` se recupera el contenido del lexicón de pronunciación especificado.

```
aws polly get-lexicon \
    --name w3c
```

Salida:

```
{
    "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
    }
}
```

Para obtener más información, consulte Uso de la GetLexicon operación en la Guía para desarrolladores de Amazon Polly.

- Para obtener más información sobre la API, consulte GetLexiconla Referencia de AWS CLI comandos.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
```

- Para obtener más información sobre la API, consulta GetLexiconla AWS Referencia de API de SDK for Python (Boto3).

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte. Uso de Amazon Polly con un SDK AWS En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`GetSpeechSynthesisTask`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `GetSpeechSynthesisTask`.

CLI

AWS CLI

Obtención de información sobre una tarea de síntesis de voz

En el siguiente ejemplo de `get-speech-synthesis-task` se recupera información sobre la tarea de síntesis de voz especificada.

```
aws polly get-speech-synthesis-task \
    --task-id 70b61c0f-57ce-4715-a247-cae8729dcce9
```

Salida:

```
{
    "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"
    }
}
```

Para obtener más información, consulte Creación de archivos de audio largos en la Guía para desarrolladores de Amazon Polly.

- Para obtener más información sobre la API, consulte GetSpeechSynthesisTaskla Referencia de AWS CLI comandos.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
```

- Para obtener más información sobre la API, consulta GetSpeechSynthesisTaskla AWS Referencia de API de SDK for Python (Boto3).

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte. Uso de Amazon Polly con un SDK AWS En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`ListLexicons`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `ListLexicons`.

.NET

SDK para .NET

> **Note**
>
> Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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}");
            }
        }
    }
```

- Para obtener más información sobre la API, consulta ListLexiconsla Referencia AWS SDK para .NET de la API.

CLI

AWS CLI

Creación de una lista de léxicos

En el siguiente ejemplo de `list-lexicons` se enumeran sus léxicos de pronunciación.

```
aws polly list-lexicons
```

Salida:

```
{
    "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
            }
        }
    ]
}
```

Para obtener más información, consulte Uso de la ListLexicons operación en la Guía para desarrolladores de Amazon Polly.

- Para obtener más información sobre la API, consulte ListLexiconsla Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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);
        }
    }
}
```

- Para obtener más información sobre la API, consulta ListLexiconsla Referencia AWS SDK for Java 2.x de la API.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
```

- Para obtener más información sobre la API, consulta ListLexiconsla AWS Referencia de API de SDK for Python (Boto3).

Ruby

SDK para Ruby

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
```

- Para obtener más información sobre la API, consulta ListLexiconsla Referencia AWS SDK para Ruby de la API.

Rust

SDK para Rust

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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(())
}
```

- Para obtener más información sobre la API, consulta ListLexiconsla referencia sobre la API de AWS SDK para Rust.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Polly con un SDK AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`PutLexicon`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `PutLexicon`.

.NET

SDK para .NET

> **Note**
>
> Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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);
            }
        }
    }
```

- Para obtener más información sobre la API, consulta PutLexiconla Referencia AWS SDK para .NET de la API.

CLI

AWS CLI

Almacenamiento de un léxico

En el siguiente ejemplo de `put-lexicon` se almacena el léxico de pronunciación especificado. El archivo `example.pls` especifica un léxico compatible con PLS de W3C.

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

Contenido de `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>
}
```

Este comando no genera ninguna salida.

Para obtener más información, consulte Uso de la PutLexicon operación en la Guía para desarrolladores de Amazon Polly.

- Para obtener más información sobre la API, consulte PutLexiconla Referencia de AWS CLI comandos.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
```

- Para obtener más información sobre la API, consulta PutLexiconla AWS Referencia de API de SDK for Python (Boto3).

Rust

SDK para Rust

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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(())
}
```

- Para obtener más información sobre la API, consulta PutLexiconla referencia sobre la API de AWS SDK para Rust.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Polly con un SDK AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`StartSpeechSynthesisTask`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `StartSpeechSynthesisTask`.

CLI

AWS CLI

Sintetización de texto

El siguiente `start-speech-synthesis-task` ejemplo sintetiza el texto `text_file.txt` y almacena el MP3 archivo resultante en el depósito especificado.

```
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
```

Salida:

```
{
    "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"
    }
}
```

Para obtener más información, consulte Creación de archivos de audio largos en la Guía para desarrolladores de Amazon Polly.

- Para obtener más información sobre la API, consulte StartSpeechSynthesisTaskla Referencia de AWS CLI comandos.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
```

- Para obtener más información sobre la API, consulta StartSpeechSynthesisTaskla AWS Referencia de API de SDK for Python (Boto3).

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte. Uso de Amazon Polly con un SDK AWS En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

`SynthesizeSpeech`Úselo con un AWS SDK

En los siguientes ejemplos de código, se muestra cómo utilizar `SynthesizeSpeech`.

.NET

SDK para .NET

> **Note**
>
> Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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.");
        }
    }
```

Sintetice voz a partir de texto mediante marcas de voz con Amazon Polly mediante AWS un SDK.

```
    using System;
    using System.Collections.Generic;
    using System.IO;
    using System.Threading.Tasks;
    using Amazon.Polly;
    using Amazon.Polly.Model;

    public class SynthesizeSpeechMarks
    {
        public static async Task Main()
        {
            var client = new AmazonPollyClient();
            string outputFileName = "speechMarks.json";

            var synthesizeSpeechRequest = new SynthesizeSpeechRequest()
            {
                OutputFormat = OutputFormat.Json,
                SpeechMarkTypes = new List<string>
                {
                    SpeechMarkType.Viseme,
                    SpeechMarkType.Word,
                },
                VoiceId = VoiceId.Joanna,
                Text = "This is a sample text to be synthesized.",
            };

            try
            {
                using (var outputStream = new FileStream(outputFileName,
FileMode.Create, FileAccess.Write))
```

```
                {
                    var synthesizeSpeechResponse = await
client.SynthesizeSpeechAsync(synthesizeSpeechRequest);
                    var buffer = new byte[2 * 1024];
                    int readBytes;

                    var inputStream = synthesizeSpeechResponse.AudioStream;
                    while ((readBytes = inputStream.Read(buffer, 0, 2 * 1024)) >
0)
                    {
                        outputStream.Write(buffer, 0, readBytes);
                    }
                }
            }
            catch (Exception ex)
            {
                Console.WriteLine($"Error: {ex.Message}");
            }
        }
    }
```

- Para obtener más información sobre la API, consulte la referencia de SynthesizeSpeechla AWS SDK para .NET API.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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;
    }
}
```

- Para obtener más información sobre la API, consulta SynthesizeSpeechla Referencia AWS SDK for Java 2.x de la API.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
```

- Para obtener más información sobre la API, consulta SynthesizeSpeechla AWS Referencia de API de SDK for Python (Boto3).

Ruby

SDK para Ruby

> **Note**
>
> Hay más información al respecto. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
```

- Para obtener más información sobre la API, consulta SynthesizeSpeechla Referencia AWS SDK para Ruby de la API.

Rust

SDK para Rust

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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(())
}
```

- Para obtener más información sobre la API, consulta SynthesizeSpeechla referencia sobre la API de AWS SDK para Rust.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Polly con un SDK AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Escenarios para el uso de Amazon Polly AWS SDKs

Los siguientes ejemplos de código muestran cómo implementar escenarios comunes en Amazon Polly con. AWS SDKs Estos escenarios muestran cómo realizar tareas específicas llamando a varias funciones de Amazon Polly o combinándolas con otras. Servicios de AWS En cada escenario se incluye un enlace al código fuente completo, con instrucciones de configuración y ejecución del código.

Los escenarios requieren un nivel intermedio de experiencia para ayudarlo a entender las acciones de servicio en su contexto.

Ejemplos

- Convierta texto a voz y devuélvalo a texto con un SDK AWS
- Cree una aplicación de sincronización de labios con Amazon Polly mediante un SDK AWS
- Creación de una aplicación que analice los comentarios de los clientes y sintetice el audio

Convierta texto a voz y devuélvalo a texto con un SDK AWS

En el siguiente ejemplo de código, se muestra cómo:

- Utilice Amazon Polly para sintetizar un archivo de entrada de texto sin formato (UTF-8) en un archivo de audio.
- Cargue el archivo de audio en un bucket de Amazon S3.
- Utilice Amazon Transcribe para convertir el archivo de audio en texto.
- Muestre el texto.

Rust

SDK para Rust

Utilice Amazon Polly para sintetizar un archivo de entrada de texto sin formato (UTF-8) en un archivo de audio, cargue el archivo de audio en un bucket de Amazon S3, utilice Amazon Transcribe para convertir ese archivo de audio en texto y muestre el texto.

Para ver el código fuente completo y las instrucciones sobre cómo configurarlo y ejecutarlo, consulta el ejemplo completo en GitHub.

Servicios utilizados en este ejemplo

- Amazon Polly
- Amazon S3
- Amazon Transcribe

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Polly con un SDK AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Cree una aplicación de sincronización de labios con Amazon Polly mediante un SDK AWS

El siguiente ejemplo de código muestra cómo crear una aplicación de sincronización labial con Amazon Polly.

Python

SDK para Python (Boto3)

Muestra cómo usar Amazon Polly y Tkinter para crear una aplicación de sincronización labial que muestre un rostro animado hablando junto con el discurso sintetizado por Amazon Polly. La sincronización labial se realiza solicitando una lista de visemas de Amazon Polly que coincidan con la voz sintetizada.

- Obtenga metadatos de voz de Amazon Polly y muéstrelos en una aplicación de Tkinter.
- Obtenga audio de voz sintetizado y marcas de voz de visema coincidentes de Amazon Polly.
- Reproduzca el audio con los movimientos de la boca sincronizados en una cara animada.
- Envíe tareas de síntesis asincrónicas para textos de gran tamaño y recupere los datos de salida desde un bucket de Amazon Simple Storage Service (Amazon S3).

Para ver el código fuente completo y las instrucciones sobre cómo configurarlo y ejecutarlo, consulta el ejemplo completo en GitHub.

Servicios utilizados en este ejemplo

- Amazon Polly

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Polly con un SDK AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Creación de una aplicación que analice los comentarios de los clientes y sintetice el audio

En los siguientes ejemplos de código, se muestra cómo crear una aplicación que analice las tarjetas de comentarios de los clientes, las traduzca del idioma original, determine sus opiniones y genere un archivo de audio a partir del texto traducido.

.NET

SDK para .NET

Esta aplicación de ejemplo analiza y almacena las tarjetas de comentarios de los clientes. Concretamente, satisface la necesidad de un hotel ficticio en la ciudad de Nueva York. El hotel recibe comentarios de los huéspedes en varios idiomas en forma de tarjetas de comentarios físicas. Esos comentarios se cargan en la aplicación a través de un cliente web. Una vez cargada la imagen de una tarjeta de comentarios, se llevan a cabo los siguientes pasos:

- El texto se extrae de la imagen mediante Amazon Textract.
- Amazon Comprehend determina la opinión del texto extraído y su idioma.
- El texto extraído se traduce al inglés mediante Amazon Translate.
- Amazon Polly sintetiza un archivo de audio a partir del texto extraído.

La aplicación completa se puede implementar con AWS CDK. Para obtener el código fuente y las instrucciones de implementación, consulte el proyecto en GitHub.

Servicios utilizados en este ejemplo

- Amazon Comprehend
- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

Java

SDK para Java 2.x

Esta aplicación de ejemplo analiza y almacena las tarjetas de comentarios de los clientes. Concretamente, satisface la necesidad de un hotel ficticio en la ciudad de Nueva York. El hotel recibe comentarios de los huéspedes en varios idiomas en forma de tarjetas de comentarios físicas. Esos comentarios se cargan en la aplicación a través de un cliente web. Una vez cargada la imagen de una tarjeta de comentarios, se llevan a cabo los siguientes pasos:

- El texto se extrae de la imagen mediante Amazon Textract.
- Amazon Comprehend determina la opinión del texto extraído y su idioma.
- El texto extraído se traduce al inglés mediante Amazon Translate.
- Amazon Polly sintetiza un archivo de audio a partir del texto extraído.

La aplicación completa se puede implementar con AWS CDK. Para obtener el código fuente y las instrucciones de implementación, consulte el proyecto en GitHub.

Servicios utilizados en este ejemplo

- Amazon Comprehend
- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

JavaScript

SDK para JavaScript (v3)

Esta aplicación de ejemplo analiza y almacena las tarjetas de comentarios de los clientes. Concretamente, satisface la necesidad de un hotel ficticio en la ciudad de Nueva York. El hotel recibe comentarios de los huéspedes en varios idiomas en forma de tarjetas de comentarios físicas. Esos comentarios se cargan en la aplicación a través de un cliente web. Una vez cargada la imagen de una tarjeta de comentarios, se llevan a cabo los siguientes pasos:

- El texto se extrae de la imagen mediante Amazon Textract.
- Amazon Comprehend determina la opinión del texto extraído y su idioma.

- El texto extraído se traduce al inglés mediante Amazon Translate.
- Amazon Polly sintetiza un archivo de audio a partir del texto extraído.

La aplicación completa se puede implementar con AWS CDK. Para obtener el código fuente y las instrucciones de implementación, consulte el proyecto en GitHub. Los siguientes extractos muestran cómo AWS SDK para JavaScript se usa dentro de las funciones 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 };
};
```

Servicios utilizados en este ejemplo

- Amazon Comprehend
- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

Ruby

SDK para Ruby

Esta aplicación de ejemplo analiza y almacena las tarjetas de comentarios de los clientes. Concretamente, satisface la necesidad de un hotel ficticio en la ciudad de Nueva York. El hotel recibe comentarios de los huéspedes en varios idiomas en forma de tarjetas de comentarios físicas. Esos comentarios se cargan en la aplicación a través de un cliente web. Una vez cargada la imagen de una tarjeta de comentarios, se llevan a cabo los siguientes pasos:

- El texto se extrae de la imagen mediante Amazon Textract.
- Amazon Comprehend determina la opinión del texto extraído y su idioma.
- El texto extraído se traduce al inglés mediante Amazon Translate.

- Amazon Polly sintetiza un archivo de audio a partir del texto extraído.

La aplicación completa se puede implementar con AWS CDK. Para obtener el código fuente y las instrucciones de implementación, consulte el proyecto en. GitHub

Servicios utilizados en este ejemplo

- Amazon Comprehend
- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso de Amazon Polly con un SDK AWS. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Seguridad en Amazon Polly

La seguridad en la nube AWS es la máxima prioridad. Como AWS cliente, usted se beneficia de una arquitectura de centro de datos y red diseñada para cumplir con los requisitos de las organizaciones más sensibles a la seguridad.

La seguridad es una responsabilidad compartida entre usted AWS y usted. El modelo de responsabilidad compartida la describe como seguridad de la nube y seguridad en la nube:

- Seguridad de la nube: AWS es responsable de proteger la infraestructura que ejecuta AWS los servicios en la AWS nube. AWS también le proporciona servicios que puede utilizar de forma segura. Los auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad como parte de los AWS programas de de . Para obtener información sobre los programas de conformidad que se aplican a Amazon Polly, consulte AWS Servicios incluidos en el ámbito de aplicación por programa de conformidad Servicios incluidos en el ámbito de aplicación por programa AWS .
- Seguridad en la nube: su responsabilidad viene determinada por el AWS servicio que utilice. También es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y los reglamentos aplicables.

Esta documentación le permite comprender cómo aplicar el modelo de responsabilidad compartida cuando se utiliza Amazon Polly. En los siguientes temas, se mostrará cómo configurar Amazon Polly para satisfacer sus objetivos de seguridad y conformidad. También aprenderá a utilizar otros AWS servicios que le ayudan a supervisar y proteger sus recursos de Amazon Polly.

Temas

- Protección de los datos en Amazon Polly
- Identity and Access Management en Amazon Polly
- Registro y monitoreo en Amazon Polly
- Validación de la conformidad para Amazon Polly
- Resiliencia en Amazon Polly
- Seguridad de la infraestructura de Amazon Polly
- Prácticas recomendadas de seguridad para Amazon Polly
- Uso de Amazon Polly con los puntos de conexión de VPC de la interfaz

Protección de los datos en Amazon Polly

Amazon Polly se ajusta al modelo de responsabilidad AWS compartida, que incluye normas y directrices para la protección de datos. AWS es responsable de proteger la infraestructura global en la que se ejecutan todos los AWS servicios. AWS mantiene el control de los datos alojados en esta infraestructura, incluidos los controles de configuración de seguridad para gestionar el contenido y los datos personales de los clientes. AWS los clientes y los socios de APN, que actúan como controladores o procesadores de datos, son responsables de cualquier dato personal que coloquen en la AWS nube.

Con fines de protección de datos, le recomendamos que proteja las credenciales de la AWS cuenta y configure los usuarios individuales con AWS Identity and Access Management (IAM), de modo que cada usuario reciba únicamente los permisos necesarios para cumplir con sus obligaciones laborales. También recomendamos proteger sus datos de la siguiente manera:

- Utiliza la autenticación multifactor (MFA) en cada cuenta.
- Se utiliza SSL/TLS para comunicarse con AWS los recursos.
- Configure la API y el registro de actividad de los usuarios con AWS CloudTrail.
- Utilice soluciones de AWS cifrado, junto con todos los controles de seguridad predeterminados de AWS los servicios.

Le recomendamos encarecidamente que nunca introduzca información de identificación confidencial, como, por ejemplo, números de cuenta de sus clientes, en los campos de formato libre, como el campo Nombre. Esto incluye cuando trabaja con Amazon Polly u otros AWS servicios mediante la consola, la API o. AWS CLI AWS SDKs Es posible que cualquier dato que introduzca en Amazon Polly u otros servicios se incluya en los registros de diagnóstico. Cuando proporcione una URL a un servidor externo, no incluya información de credenciales en la URL para validar la solicitud para ese servidor.

Para obtener más información sobre la protección de datos, consulte la entrada de blog relativa al modelo de responsabilidad compartida de AWS y GDPR en el blog de seguridad de AWS .

Cifrado en reposo

La salida de la síntesis de voz de Amazon Polly se puede guardar en su propio sistema. También puede llamar a Amazon Polly y, a continuación, cifrar el archivo con cualquier clave de cifrado que elija y almacenarlo en Amazon Simple Storage Service (Amazon S3) u otro almacenamiento seguro.

La operación de Amazon Polly the section called “SynthesizeSpeech” no tiene estado y no está asociada a ninguna identidad de cliente. No puede recuperarla de Amazon Polly más adelante.

Cifrado en tránsito

Todos los envíos de texto están protegidos por TLS mientras están en tránsito. Amazon Polly no retiene el contenido de los envíos de texto.

Privacidad del tráfico entre redes

El acceso a Amazon Polly se realiza a través de la AWS consola, la CLI o. SDKs Las comunicaciones utilizan el cifrado de sesión de Transport Layer Security (TLS) para la confidencialidad y firmas digitales para la autenticación y la integridad.

Identity and Access Management en Amazon Polly

AWS Identity and Access Management (IAM) es una herramienta Servicio de AWS que ayuda al administrador a controlar de forma segura el acceso a los AWS recursos. Los administradores de IAM controlan quién se puede autenticar (iniciar sesión) y autorizar (tener permisos) para utilizar los recursos de Amazon Polly. La IAM es una Servicio de AWS herramienta que puede utilizar sin coste adicional.

Temas

- Público
- Autenticación con identidades
- Administración de acceso mediante políticas
- Cómo funciona Amazon Polly con IAM
- Ejemplos de políticas basadas en identidad para Amazon Polly
- Permisos de la API de Amazon Polly: referencia de acciones, permisos y recursos
- Solución de problemas de identidad y acceso de Amazon Polly

Público

La forma de usar AWS Identity and Access Management (IAM) varía según el trabajo que realice en Amazon Polly.

Usuario de servicio: si utiliza el servicio de Amazon Polly para realizar su trabajo, su administrador le proporciona las credenciales y los permisos que necesita. A medida que utilice más características de Amazon Polly para realizar su trabajo, es posible que necesite permisos adicionales. Entender cómo se administra el acceso puede ayudarle a solicitar los permisos correctos al administrador. Si no puede acceder a una característica de Amazon Polly, consulte Solución de problemas de identidad y acceso de Amazon Polly.

Administrador de servicio: si está a cargo de los recursos de Amazon Polly de su empresa, probablemente tenga acceso completo a Amazon Polly. El trabajo consiste en determinar a qué características y recursos de Amazon Polly deben acceder los usuarios del servicio. Luego, debe enviar solicitudes a su gestionador de IAM para cambiar los permisos de los usuarios de su servicio. Revise la información de esta página para conocer los conceptos básicos de IAM. Para obtener más información acerca de cómo la empresa puede utilizar IAM con Amazon Polly, consulte Cómo funciona Amazon Polly con IAM.

Administrador de IAM: si es un administrador de IAM, es posible que desee obtener información acerca de cómo escribir políticas para administrar el acceso a Amazon Polly. Para consultar ejemplos de políticas de Amazon Polly basadas en identidades que puede utilizar en IAM, consulte Ejemplos de políticas basadas en identidad para Amazon Polly.

Autenticación con identidades

La autenticación es la forma de iniciar sesión para AWS usar sus credenciales de identidad. Debe estar autenticado (con quien haya iniciado sesión AWS) como usuario de IAM o asumiendo una función de IAM. Usuario raíz de la cuenta de AWS

Puede iniciar sesión AWS como una identidad federada mediante las credenciales proporcionadas a través de una fuente de identidad. AWS IAM Identity Center Los usuarios (Centro de identidades de IAM), la autenticación de inicio de sesión único de su empresa y sus credenciales de Google o Facebook son ejemplos de identidades federadas. Al iniciar sesión como una identidad federada, su gestionador habrá configurado previamente la federación de identidades mediante roles de IAM. Cuando accedes AWS mediante la federación, estás asumiendo un rol de forma indirecta.

Según el tipo de usuario que sea, puede iniciar sesión en el portal AWS Management Console o en el de AWS acceso. Para obtener más información sobre cómo iniciar sesión AWS, consulte Cómo iniciar sesión Cuenta de AWS en su Guía del AWS Sign-In usuario.

Si accede AWS mediante programación, AWS proporciona un kit de desarrollo de software (SDK) y una interfaz de línea de comandos (CLI) para firmar criptográficamente sus solicitudes con sus

credenciales. Si no utilizas AWS herramientas, debes firmar las solicitudes tú mismo. Para obtener más información sobre la firma de solicitudes, consulte AWS Signature Versión 4 para solicitudes API en la Guía del usuario de IAM.

Independientemente del método de autenticación que use, es posible que deba proporcionar información de seguridad adicional. Por ejemplo, le AWS recomienda que utilice la autenticación multifactor (MFA) para aumentar la seguridad de su cuenta. Para obtener más información, consulte Autenticación multifactor en la Guía del usuario de AWS IAM Identity Center y Autenticación multifactor AWS en IAM en la Guía del usuario de IAM.

Cuenta de AWS usuario root

Al crear una Cuenta de AWS, comienza con una identidad de inicio de sesión que tiene acceso completo a todos Servicios de AWS los recursos de la cuenta. Esta identidad se denomina usuario Cuenta de AWS raíz y se accede a ella iniciando sesión con la dirección de correo electrónico y la contraseña que utilizaste para crear la cuenta. Recomendamos encarecidamente que no utiliza el usuario raíz para sus tareas diarias. Proteja las credenciales del usuario raíz y utilícelas solo para las tareas que solo el usuario raíz pueda realizar. Para ver la lista completa de las tareas que requieren que inicie sesión como usuario raíz, consulta Tareas que requieren credenciales de usuario raíz en la Guía del usuario de IAM.

Identidad federada

Como práctica recomendada, exija a los usuarios humanos, incluidos los que requieren acceso de administrador, que utilicen la federación con un proveedor de identidades para acceder Servicios de AWS mediante credenciales temporales.

Una identidad federada es un usuario del directorio de usuarios de su empresa, un proveedor de identidades web AWS Directory Service, el directorio del Centro de Identidad o cualquier usuario al que acceda Servicios de AWS mediante las credenciales proporcionadas a través de una fuente de identidad. Cuando las identidades federadas acceden Cuentas de AWS, asumen funciones y las funciones proporcionan credenciales temporales.

Para una administración de acceso centralizada, le recomendamos que utiliza AWS IAM Identity Center. Puede crear usuarios y grupos en el Centro de identidades de IAM, o puede conectarse y sincronizarse con un conjunto de usuarios y grupos de su propia fuente de identidad para usarlos en todas sus Cuentas de AWS aplicaciones. Para obtener más información, consulta ¿Qué es el Centro de identidades de IAM? en la Guía del usuario de AWS IAM Identity Center .

Usuarios y grupos de IAM

Un usuario de IAM es una identidad propia Cuenta de AWS que tiene permisos específicos para una sola persona o aplicación. Siempre que sea posible, recomendamos emplear credenciales temporales, en lugar de crear usuarios de IAM que tengan credenciales de larga duración como contraseñas y claves de acceso. No obstante, si tiene casos de uso específicos que requieran credenciales de larga duración con usuarios de IAM, recomendamos rotar las claves de acceso. Para más información, consulte Rotar las claves de acceso periódicamente para casos de uso que requieran credenciales de larga duración en la Guía del usuario de IAM.

Un grupo de IAM es una identidad que especifica un conjunto de usuarios de IAM. No puedes iniciar sesión como grupo. Puedes usar los grupos para especificar permisos para varios usuarios a la vez. Los grupos facilitan la administración de los permisos para grandes conjuntos de usuarios. Por ejemplo, puede asignar un nombre a un grupo IAMAdminsy concederle permisos para administrar los recursos de IAM.

Los usuarios son diferentes de los roles. Un usuario se asocia exclusivamente a una persona o aplicación, pero la intención es que cualquier usuario pueda asumir un rol que necesite. Los usuarios tienen credenciales de larga duración permanentes; no obstante, los roles proporcionan credenciales temporales. Para obtener más información, consulte Casos de uso para usuarios de IAM en la Guía del usuario de IAM.

Roles de IAM

Un rol de IAM es una identidad dentro de usted Cuenta de AWS que tiene permisos específicos. Es similar a un usuario de IAM, pero no está asociado a una persona determinada. Para asumir temporalmente un rol de IAM en el AWS Management Console, puede cambiar de un rol de usuario a uno de IAM (consola). Puedes asumir un rol llamando a una operación de AWS API AWS CLI o usando una URL personalizada. Para más información sobre los métodos para el uso de roles, consulta Métodos para asumir un rol en la Guía del usuario de IAM.

Los roles de IAM con credenciales temporales son útiles en las siguientes situaciones:

- Acceso de usuario federado: para asignar permisos a una identidad federada, puede crear un rol y definir sus permisos. Cuando se autentica una identidad federada, se asocia la identidad al rol y se le conceden los permisos define el rol. Para obtener información acerca de roles de federación, consulte Crear un rol para un proveedor de identidad de terceros (federación) en la Guía de usuario de IAM. Si utiliza el IAM Identity Center, debe configurar un conjunto de permisos. IAM Identity Center correlaciona el conjunto de permisos con un rol en IAM para controlar a qué

pueden acceder las identidades después de autenticarse. Para obtener información acerca de los conjuntos de permisos, consulta Conjuntos de permisos en la Guía del usuario de AWS IAM Identity Center .

- Permisos de usuario de IAM temporales: un usuario de IAM puede asumir un rol de IAM para recibir temporalmente permisos distintos que le permitan realizar una tarea concreta.
- Acceso entre cuentas: puede utilizar un rol de IAM para permitir que alguien (una entidad principal de confianza) de otra cuenta acceda a los recursos de la cuenta. Los roles son la forma principal de conceder acceso entre cuentas. Sin embargo, con algunas Servicios de AWS, puedes adjuntar una política directamente a un recurso (en lugar de usar un rol como proxy). Para obtener información acerca de la diferencia entre los roles y las políticas basadas en recursos para el acceso entre cuentas, consulta Acceso a recursos entre cuentas en IAM en la Guía del usuario de IAM.
- Acceso entre servicios: algunos Servicios de AWS utilizan funciones en otros Servicios de AWS. Por ejemplo, cuando realizas una llamada en un servicio, es habitual que ese servicio ejecute aplicaciones en Amazon EC2 o almacene objetos en Amazon S3. Es posible que un servicio haga esto usando los permisos de la entidad principal, usando un rol de servicio o usando un rol vinculado al servicio.
  - Sesiones de acceso directo (FAS): cuando utilizas un usuario o un rol de IAM para realizar acciones en AWS ellas, se te considera principal. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. El FAS utiliza los permisos del principal que llama Servicio de AWS y los solicita Servicio de AWS para realizar solicitudes a los servicios descendentes. Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte Reenviar sesiones de acceso.
  - Rol de servicio: un rol de servicio es un rol de IAM que adopta un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte Creación de un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.
  - Función vinculada al servicio: una función vinculada a un servicio es un tipo de función de servicio que está vinculada a un. Servicio de AWS El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

- Aplicaciones que se ejecutan en Amazon EC2: puedes usar un rol de IAM para administrar las credenciales temporales de las aplicaciones que se ejecutan en una EC2 instancia y realizan AWS CLI solicitudes a la AWS API. Esto es preferible a almacenar las claves de acceso en la EC2 instancia. Para asignar un AWS rol a una EC2 instancia y ponerlo a disposición de todas sus aplicaciones, debe crear un perfil de instancia adjunto a la instancia. Un perfil de instancia contiene el rol y permite que los programas que se ejecutan en la EC2 instancia obtengan credenciales temporales. Para obtener más información, consulte Usar un rol de IAM para conceder permisos a las aplicaciones que se ejecutan en EC2 instancias de Amazon en la Guía del usuario de IAM.

Administración de acceso mediante políticas

El acceso se controla AWS creando políticas y adjuntándolas a AWS identidades o recursos. Una política es un objeto AWS que, cuando se asocia a una identidad o un recurso, define sus permisos. AWS evalúa estas políticas cuando un director (usuario, usuario raíz o sesión de rol) realiza una solicitud. Los permisos en las políticas determinan si la solicitud se permite o se deniega. La mayoría de las políticas se almacenan AWS como documentos JSON. Para obtener más información sobre la estructura y el contenido de los documentos de política JSON, consulte Información general de políticas JSON en la Guía del usuario de IAM.

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

De forma predeterminada, los usuarios y los roles no tienen permisos. Un administrador de IAM puede crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los recursos que necesitan. A continuación, el administrador puede añadir las políticas de IAM a roles y los usuarios pueden asumirlos.

Las políticas de IAM definen permisos para una acción independientemente del método que se utiliza para realizar la operación. Por ejemplo, suponga que dispone de una política que permite la acción `iam:GetRole`. Un usuario con esa política puede obtener información sobre el rol de la API AWS Management Console AWS CLI, la o la AWS API.

Políticas basadas en identidades

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué

condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Las políticas basadas en identidades pueden clasificarse además como políticas insertadas o políticas administradas. Las políticas insertadas se integran directamente en un único usuario, grupo o rol. Las políticas administradas son políticas independientes que puede adjuntar a varios usuarios, grupos y roles de su Cuenta de AWS empresa. Las políticas administradas incluyen políticas AWS administradas y políticas administradas por el cliente. Para obtener más información sobre cómo elegir una política administrada o una política insertada, consulte Elegir entre políticas administradas y políticas insertadas en la Guía del usuario de IAM.

Políticas basadas en recursos

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Los ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Las políticas basadas en recursos son políticas insertadas que se encuentran en ese servicio. No puedes usar políticas AWS gestionadas de IAM en una política basada en recursos.

Listas de control de acceso () ACLs

Las listas de control de acceso (ACLs) controlan qué responsables (miembros de la cuenta, usuarios o roles) tienen permisos para acceder a un recurso. ACLs son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

Amazon S3 y Amazon VPC son ejemplos de servicios compatibles. AWS WAF ACLs Para obtener más información ACLs, consulte la descripción general de la lista de control de acceso (ACL) en la Guía para desarrolladores de Amazon Simple Storage Service.

Otros tipos de políticas

AWS admite tipos de políticas adicionales y menos comunes. Estos tipos de políticas pueden establecer el máximo de permisos que los tipos de políticas más frecuentes le conceden.

- Límites de permisos: un límite de permisos es una característica avanzada que le permite establecer los permisos máximos que una política basada en identidad puede conceder a una entidad de IAM (usuario o rol de IAM). Puedes establecer un límite de permisos para una entidad. Los permisos resultantes son la intersección de las políticas basadas en la identidad de la entidad y los límites de permisos. Las políticas basadas en recursos que especifiquen el usuario o rol en el campo `Principal` no estarán restringidas por el límite de permisos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para obtener más información sobre los límites de los permisos, consulte Límites de permisos para las entidades de IAM en la Guía del usuario de IAM.
- Políticas de control de servicios (SCPs): SCPs son políticas de JSON que especifican los permisos máximos para una organización o unidad organizativa (OU). AWS Organizations AWS Organizations es un servicio para agrupar y administrar de forma centralizada varios de los Cuentas de AWS que son propiedad de su empresa. Si habilitas todas las funciones de una organización, puedes aplicar políticas de control de servicios (SCPs) a una o a todas tus cuentas. El SCP limita los permisos de las entidades en las cuentas de los miembros, incluidas las de cada una Usuario raíz de la cuenta de AWS. Para obtener más información sobre Organizations SCPs, consulte las políticas de control de servicios en la Guía del AWS Organizations usuario.
- Políticas de control de recursos (RCPs): RCPs son políticas de JSON que puedes usar para establecer los permisos máximos disponibles para los recursos de tus cuentas sin actualizar las políticas de IAM asociadas a cada recurso que poseas. El RCP limita los permisos de los recursos en las cuentas de los miembros y puede afectar a los permisos efectivos de las identidades, incluidos los permisos Usuario raíz de la cuenta de AWS, independientemente de si pertenecen a su organización. Para obtener más información sobre Organizations e RCPs incluir una lista de Servicios de AWS ese apoyo RCPs, consulte Políticas de control de recursos (RCPs) en la Guía del AWS Organizations usuario.
- Políticas de sesión: las políticas de sesión son políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal mediante programación para un rol o un usuario federado. Los permisos de la sesión resultantes son la intersección de las políticas basadas en identidades del rol y las políticas de la sesión. Los permisos también pueden proceder de una política en función de recursos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para más información, consulte Políticas de sesión en la Guía del usuario de IAM.

Varios tipos de políticas

Cuando se aplican varios tipos de políticas a una solicitud, los permisos resultantes son más complicados de entender. Para saber cómo se AWS determina si se debe permitir una solicitud

cuando se trata de varios tipos de políticas, consulte la lógica de evaluación de políticas en la Guía del usuario de IAM.

Cómo funciona Amazon Polly con IAM

Antes de utilizar IAM para administrar el acceso a Amazon Polly, obtenga información sobre qué características de IAM se encuentran disponibles con Amazon Polly.

Características de IAM que puede utilizar con Amazon Polly

Característica de IAM	Soporte de Amazon Polly
Políticas basadas en identidades	Sí
Políticas basadas en recursos	No
Acciones de políticas	Sí
Recursos de políticas	Sí
Claves de condición de política (específicas del servicio)	No
ACLs	No
ABAC (etiquetas en políticas)	No
Credenciales temporales	Sí
Sesiones de acceso directo (FAS) para Amazon Polly	Sí
Roles de servicio	No
Roles vinculados al servicio	No

Para obtener una visión general de cómo funcionan Amazon Polly y otros AWS servicios con la mayoría de las funciones de IAM, consulte los AWS servicios que funcionan con IAM en la Guía del usuario de IAM.

Políticas de Amazon Polly basadas en identidad

Compatibilidad con las políticas basadas en identidad: sí

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Con las políticas basadas en identidades de IAM, puede especificar las acciones y los recursos permitidos o denegados, así como las condiciones en las que se permiten o deniegan las acciones. No es posible especificar la entidad principal en una política basada en identidad porque se aplica al usuario o rol al que está asociada. Para obtener más información sobre los elementos que puede utilizar en una política de JSON, consulte Referencia de los elementos de las políticas de JSON de IAM en la Guía del usuario de IAM.

Ejemplos de políticas basadas en identidad para Amazon Polly

Para ver ejemplos de políticas basadas en identidad de Amazon Polly, consulte Ejemplos de políticas basadas en identidad para Amazon Polly.

Políticas basadas en recursos de Amazon Polly

Admite políticas basadas en recursos: no

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Los ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Para habilitar el acceso entre cuentas, puede especificar toda una cuenta o entidades de IAM de otra cuenta como la entidad principal de una política en función de recursos. Añadir a una política en función de recursos una entidad principal entre cuentas es solo una parte del establecimiento

de una relación de confianza. Cuando el principal y el recurso son diferentes Cuentas de AWS, el administrador de IAM de la cuenta de confianza también debe conceder a la entidad principal (usuario o rol) permiso para acceder al recurso. Para conceder el permiso, adjunte la entidad a una política basada en identidad. Sin embargo, si la política basada en recursos concede acceso a una entidad principal de la misma cuenta, no es necesaria una política basada en identidad adicional. Para obtener más información, consulte Cross account resource access in IAM en la Guía del usuario de IAM.

Acciones de política de Amazon Polly

Compatibilidad con las acciones de políticas: sí

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Action` de una política JSON describe las acciones que puede utilizar para conceder o denegar el acceso en una política. Las acciones políticas suelen tener el mismo nombre que la operación de AWS API asociada. Hay algunas excepciones, como acciones de solo permiso que no tienen una operación de API coincidente. También hay algunas operaciones que requieren varias acciones en una política. Estas acciones adicionales se denominan acciones dependientes.

Incluya acciones en una política para conceder permisos y así llevar a cabo la operación asociada.

Para ver una lista de las acciones de Amazon Polly, consulte Acciones definidas por Amazon Polly en la Referencia de autorizaciones de servicio.

Las acciones de políticas de Amazon Polly utilizan el siguiente prefijo antes de la acción:

```
polly
```

Para especificar varias acciones en una única instrucción, sepárelas con comas.

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

Para ver ejemplos de políticas basadas en identidad de Amazon Polly, consulte Ejemplos de políticas basadas en identidad para Amazon Polly.

Recursos de políticas para Amazon Polly

Compatibilidad con los recursos de políticas: sí

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puedes realizar acciones en qué recursos y en qué condiciones.

El elemento `Resource` de la política JSON especifica el objeto u objetos a los que se aplica la acción. Las instrucciones deben contener un elemento `Resource` o `NotResource`. Como práctica recomendada, especifique un recurso utilizando el Nombre de recurso de Amazon (ARN). Puedes hacerlo para acciones que admitan un tipo de recurso específico, conocido como permisos de nivel de recurso.

Para las acciones que no admiten permisos de nivel de recurso, como las operaciones de descripción, utiliza un carácter comodín (*) para indicar que la instrucción se aplica a todos los recursos.

```
"Resource": "*"
```

Para ver una lista de los tipos de recursos de Amazon Polly y sus tipos ARNs, consulte Recursos definidos por Amazon Polly en la Referencia de autorización de servicio. Para obtener información acerca de las acciones con las que puede especificar el ARN de cada recurso, consulte Acciones definidas por Amazon Polly.

Para ver ejemplos de políticas basadas en identidad de Amazon Polly, consulte Ejemplos de políticas basadas en identidad para Amazon Polly.

Claves de condición de políticas para Amazon Polly

Compatibilidad con claves de condición de políticas específicas del servicio: no

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puedes realizar acciones en qué recursos y en qué condiciones.

El elemento `Condition` (o bloque de `Condition`) permite especificar condiciones en las que entra en vigor una instrucción. El elemento `Condition` es opcional. Puedes crear expresiones

condicionales que utilizan operadores de condición, tales como igual o menor que, para que la condición de la política coincida con los valores de la solicitud.

Si especifica varios elementos de `Condition` en una instrucción o varias claves en un único elemento de `Condition`, AWS las evalúa mediante una operación `AND` lógica. Si especifica varios valores para una única clave de condición, AWS evalúa la condición mediante una `OR` operación lógica. Se deben cumplir todas las condiciones antes de que se concedan los permisos de la instrucción.

También puedes utilizar variables de marcador de posición al especificar condiciones. Por ejemplo, puedes conceder un permiso de usuario de IAM para acceder a un recurso solo si está etiquetado con su nombre de usuario de IAM. Para más información, consulta Elementos de la política de IAM: variables y etiquetas en la Guía del usuario de IAM.

AWS admite claves de condición globales y claves de condición específicas del servicio. Para ver todas las claves de condición AWS globales, consulte las claves de contexto de condición AWS globales en la Guía del usuario de IAM.

Para ver una lista de las claves de condición de Amazon Polly, consulte Claves de condición para Amazon Polly en la Referencia de autorización de servicios. Para obtener más información acerca de las acciones y los recursos con los que puede utilizar una clave de condición, consulte Acciones definidas por Amazon Polly.

Para ver ejemplos de políticas basadas en identidad de Amazon Polly, consulte Ejemplos de políticas basadas en identidad para Amazon Polly.

ACLs en Amazon Polly

Soportes ACLs: No

Las listas de control de acceso (ACLs) controlan qué directores (miembros de la cuenta, usuarios o roles) tienen permisos para acceder a un recurso. ACLs son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

ABAC con Amazon Polly

Compatibilidad con ABAC (etiquetas en las políticas): no

El control de acceso basado en atributos (ABAC) es una estrategia de autorización que define permisos en función de atributos. En AWS, estos atributos se denominan etiquetas. Puede adjuntar

etiquetas a las entidades de IAM (usuarios o roles) y a muchos AWS recursos. El etiquetado de entidades y recursos es el primer paso de ABAC. A continuación, designa las políticas de ABAC para permitir operaciones cuando la etiqueta de la entidad principal coincida con la etiqueta del recurso al que se intenta acceder.

ABAC es útil en entornos que crecen con rapidez y ayuda en situaciones en las que la administración de las políticas resulta engorrosa.

Para controlar el acceso en función de etiquetas, debe proporcionar información de las etiquetas en el elemento de condición de una política utilizando las claves de condición `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Si un servicio admite las tres claves de condición para cada tipo de recurso, el valor es Sí para el servicio. Si un servicio admite las tres claves de condición solo para algunos tipos de recursos, el valor es Parcial.

Para obtener más información sobre ABAC, consulte Definición de permisos con la autorización de ABAC en la Guía del usuario de IAM. Para ver un tutorial con los pasos para configurar ABAC, consulte Uso del control de acceso basado en atributos (ABAC) en la Guía del usuario de IAM.

Uso de credenciales temporales con Amazon Polly

Compatibilidad con credenciales temporales: sí

Algunas Servicios de AWS no funcionan cuando inicias sesión con credenciales temporales. Para obtener información adicional, incluidas las que Servicios de AWS funcionan con credenciales temporales, consulta Cómo Servicios de AWS funcionan con IAM en la Guía del usuario de IAM.

Utiliza credenciales temporales si inicia sesión en ellas AWS Management Console mediante cualquier método excepto un nombre de usuario y una contraseña. Por ejemplo, cuando accedes AWS mediante el enlace de inicio de sesión único (SSO) de tu empresa, ese proceso crea automáticamente credenciales temporales. También crea credenciales temporales de forma automática cuando inicia sesión en la consola como usuario y luego cambia de rol. Para obtener más información sobre el cambio de roles, consulte Cambio de un usuario a un rol de IAM (consola) en la Guía del usuario de IAM.

Puedes crear credenciales temporales manualmente mediante la AWS CLI API o. AWS A continuación, puede utilizar esas credenciales temporales para acceder AWS. AWS recomienda generar credenciales temporales de forma dinámica en lugar de utilizar claves de acceso a largo plazo. Para obtener más información, consulte Credenciales de seguridad temporales en IAM.

Sesiones de acceso directo (FAS) entre servicios para Amazon Polly

Admite sesiones de acceso directo (FAS): sí

Cuando utilizas un usuario o un rol de IAM para realizar acciones en AWSél, se te considera director. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. FAS utiliza los permisos del principal que llama y los que solicita Servicio de AWS para realizar solicitudes a los servicios descendentes. Servicio de AWS Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte Reenviar sesiones de acceso.

Roles de servicio para Amazon Polly

Compatible con roles de servicio: No

Un rol de servicio es un rol de IAM que asume un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte Creación de un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.

> **Warning**
>
> Cambiar los permisos de un rol de servicio podría interrumpir la funcionalidad de Amazon Polly. Edite los roles de servicio solo cuando Amazon Polly proporcione orientación para hacerlo.

Roles vinculados a servicios para Amazon Polly

Compatibilidad con roles vinculados al servicio: no

Un rol vinculado a un servicio es un tipo de rol de servicio que está vinculado a un. Servicio de AWS El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puedes ver, pero no editar, los permisos de los roles vinculados a servicios.

Para más información sobre cómo crear o administrar roles vinculados a servicios, consulta Servicios de AWS que funcionan con IAM. Busque un servicio en la tabla que incluya `Yes` en la columna Rol

vinculado a un servicio. Seleccione el vínculo Sí para ver la documentación acerca del rol vinculado a servicios para ese servicio.

Roles de IAM de Amazon Polly

Puede asociar una política de permisos basada en identidad a un rol de IAM para conceder permisos entre cuentas. Por ejemplo, el administrador de la cuenta A puede crear un rol para conceder permisos entre cuentas a otra cuenta (por ejemplo, la AWS cuenta B) o a un AWS servicio de la siguiente manera:

1. El administrador de la Cuenta A crea un rol de IAM y adjunta una política de permisos al rol que concede permisos sobre los recursos de la Cuenta A.
2. El administrador de la cuenta A asocia una política de confianza al rol que identifica la cuenta B como la entidad principal que puede asumir el rol.
3. De este modo, el administrador de la cuenta B puede delegar los permisos para asumir el rol en cualquier usuario de la cuenta B. De este modo, los usuarios de la cuenta B pueden crear o acceder a los recursos de la cuenta A. El principal de la política de confianza también puede ser un director de AWS servicio si desea conceder permisos a un AWS servicio para asumir el rol.

Para obtener más información sobre el uso de IAM para delegar permisos, consulte Administración de accesos en la Guía del usuario de IAM.

A continuación, se incluye una política de ejemplo que concede permisos para insertar y obtener lexicones, así como para mostrar la lista de los lexicones disponibles actualmente.

Amazon Polly admite políticas basadas en la identidad para acciones en el nivel de recurso. En algunos casos, el recurso puede estar limitado por un ARN. Esto es cierto para las operaciones `SynthesizeSpeech`, `StartSpeechSynthesisTask`, `PutLexicon`, `GetLexicon` y `DeleteLexicon`. En estos casos, el valor de `Resource` se indica por el ARN. Por ejemplo: `arn:aws:polly:`*`us-east-2`*`:`*`account-id`*`:lexicon/*` como el valor de `Resource` especifica permisos en todos los lexicones propios incluidos en la región `us-east-2`.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Sid": "AllowPut-Get-ListActions",
      "Effect": "Allow",
      "Action": [
```

```
            "polly:PutLexicon",
            "polly:GetLexicon",
            "polly:ListLexicons"],
        "Resource": "arn:aws:polly:us-east-2:account-id:lexicon/*"
        }
    ]
}
```

Sin embargo, no todas las operaciones lo utilizan ARNs. Este es el caso de las operaciones `DescribeVoices`, `ListLexicons`, `GetSpeechSynthesisTasks` y `ListSpeechSynthesisTasks`.

Para obtener más información sobre usuarios, grupos, roles y permisos, consulte Identidades (usuarios, grupos y roles) en la Guía del usuario de IAM.

Ejemplos de políticas basadas en identidad para Amazon Polly

De forma predeterminada, los usuarios y roles no tienen permiso para crear ni modificar los recursos de Amazon Polly. Tampoco pueden realizar tareas mediante la AWS Management Console, AWS Command Line Interface (AWS CLI) o la AWS API. Un administrador de IAM puede crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los recursos que necesitan. A continuación, el administrador puedes añadir las políticas de IAM a roles y los usuarios puedes asumirlos.

Para obtener información acerca de cómo crear una política basada en identidades de IAM mediante el uso de estos documentos de políticas JSON de ejemplo, consulte Creación de políticas de IAM (consola) en la Guía del usuario de IAM.

Para obtener más información sobre las acciones y los tipos de recursos definidos por Amazon Polly, incluido el formato de cada uno de los tipos de recursos, consulte Acciones, recursos y claves de condición de Amazon Polly en la Referencia de autorización de servicio. ARNs

Temas

- Prácticas recomendadas sobre las políticas
- Uso de la consola de Amazon Polly
- Cómo permitir a los usuarios consultar sus propios permisos
- AWS políticas gestionadas (predefinidas) para Amazon Polly
- Ejemplos de políticas administradas por los clientes

Prácticas recomendadas sobre las políticas

Las políticas basadas en identidad determinan si alguien puede crear, eliminar o acceder a los recursos de Amazon Polly de la cuenta. Estas acciones pueden generar costos adicionales para su Cuenta de AWS. Siga estas directrices y recomendaciones al crear o editar políticas basadas en identidades:

- Comience con las políticas AWS administradas y avance hacia los permisos con privilegios mínimos: para empezar a conceder permisos a sus usuarios y cargas de trabajo, utilice las políticas AWS administradas que otorgan permisos para muchos casos de uso comunes. Están disponibles en su. Cuenta de AWS Le recomendamos que reduzca aún más los permisos definiendo políticas administradas por el AWS cliente que sean específicas para sus casos de uso. Con el fin de obtener más información, consulta las políticas administradas por AWS o las políticas administradas por AWS para funciones de tarea en la Guía de usuario de IAM.
- Aplique permisos de privilegio mínimo: cuando establezca permisos con políticas de IAM, conceda solo los permisos necesarios para realizar una tarea. Para ello, debe definir las acciones que se pueden llevar a cabo en determinados recursos en condiciones específicas, también conocidos como permisos de privilegios mínimos. Con el fin de obtener más información sobre el uso de IAM para aplicar permisos, consulta Políticas y permisos en IAM en la Guía del usuario de IAM.
- Utilice condiciones en las políticas de IAM para restringir aún más el acceso: puede agregar una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puede escribir una condición de políticas para especificar que todas las solicitudes deben enviarse utilizando SSL. También puedes usar condiciones para conceder el acceso a las acciones del servicio si se utilizan a través de una acción específica Servicio de AWS, por ejemplo AWS CloudFormation. Para obtener más información, consulta Elementos de la política de JSON de IAM: Condición en la Guía del usuario de IAM.
- Utiliza el analizador de acceso de IAM para validar las políticas de IAM con el fin de garantizar la seguridad y funcionalidad de los permisos: el analizador de acceso de IAM valida políticas nuevas y existentes para que respeten el lenguaje (JSON) de las políticas de IAM y las prácticas recomendadas de IAM. El analizador de acceso de IAM proporciona más de 100 verificaciones de políticas y recomendaciones procesables para ayudar a crear políticas seguras y funcionales. Para más información, consulte Validación de políticas con el Analizador de acceso de IAM en la Guía del usuario de IAM.
- Requerir autenticación multifactor (MFA): si tiene un escenario que requiere usuarios de IAM o un usuario raíz en Cuenta de AWS su cuenta, active la MFA para mayor seguridad. Para exigir la

MFA cuando se invoquen las operaciones de la API, añada condiciones de MFA a sus políticas. Para más información, consulte Acceso seguro a la API con MFA en la Guía del usuario de IAM.

Para obtener más información sobre las prácticas recomendadas de IAM, consulte Prácticas recomendadas de seguridad en IAM en la Guía del usuario de IAM.

Uso de la consola de Amazon Polly

Para acceder a la consola de Amazon Polly, debe tener un conjunto mínimo de permisos. Estos permisos deben permitirle enumerar y ver detalles sobre los recursos de Amazon Polly que tiene en su cuenta. Cuenta de AWS Si crea una política basada en identidades que sea más restrictiva que el mínimo de permisos necesarios, la consola no funcionará del modo esperado para las entidades (usuarios o roles) que tengan esa política.

No necesita conceder permisos mínimos de consola a los usuarios que solo realicen llamadas a la AWS API AWS CLI o a la misma. En su lugar, permite el acceso únicamente a las acciones que coincidan con la operación de API que intentan realizar.

Para garantizar que los usuarios y los roles puedan seguir utilizando la consola de Amazon Polly, adjunte también la política gestionada *ReadOnly* AWS o Amazon *ConsoleAccess* Polly a las entidades. Para obtener más información, consulte Adición de permisos a un usuario en la Guía del usuario de IAM:

Para usar la consola Amazon Polly, concede permisos a todas las Amazon Polly. APIs No es necesario ningún otro permiso. Para obtener la funcionalidad completa de la consola, puede usar la siguiente política:.

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

Cómo permitir a los usuarios consultar sus propios permisos

En este ejemplo, se muestra cómo podría crear una política que permita a los usuarios de IAM ver las políticas gestionadas e insertadas que se asocian a la identidad de sus usuarios. Esta política incluye permisos para completar esta acción en la consola o mediante programación mediante la API o. 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 políticas gestionadas (predefinidas) para Amazon Polly

AWS aborda muchos casos de uso comunes al proporcionar políticas de IAM independientes que son creadas y administradas por. AWS Estas políticas AWS gestionadas conceden los permisos necesarios para los casos de uso más habituales, de forma que no tenga que investigar qué permisos son necesarios. Para más información, consulte Políticas administradas de AWS en la Guía del usuario de IAM.

Las siguientes políticas AWS gestionadas, que puedes adjuntar a los usuarios de tu cuenta, son específicas de Amazon Polly:

- AmazonPollyReadOnlyAccess— Otorga acceso de solo lectura a los recursos, permite enumerar léxicos, buscar léxicos, enumerar las voces disponibles y sintetizar el habla (incluida la aplicación de léxicos al discurso sintetizado).
- AmazonPollyFullAccess— Otorga acceso total a los recursos y a todas las operaciones compatibles.

> **Note**
>
> Para consultar estas políticas de permisos, inicie sesión en la consola de IAM y busque las políticas específicas.

También puede crear sus propias políticas de IAM personalizadas para conceder permisos a las acciones y recursos de Amazon Polly. Puede asociar estas políticas personalizadas a los grupos o usuarios de IAM que requieran esos permisos.

Ejemplos de políticas administradas por los clientes

En esta sección, encontrará ejemplos de políticas de usuario que conceden permisos para diversas acciones de Amazon Polly. Estas políticas funcionan cuando se utiliza AWS SDKs o el AWS CLI. Cuando utilices la consola, concede permisos a todas las Amazon Polly APIs.

> **Note**
>
> Todos los ejemplos utilizan la región us-east-2 y contienen una cuenta ficticia. IDs

Ejemplos

- Ejemplo 1: Permitir todas las acciones de Amazon Polly
- Ejemplo 2: Permitir todas las acciones de Amazon Polly excepto DeleteLexicon
- Ejemplo 3: Permitir DeleteLexicon
- Ejemplo 4: Permitir DeleteLexicon en una región concreta
- Ejemplo 5: Permitir DeleteLexicon un léxico específico

Ejemplo 1: Permitir todas las acciones de Amazon Polly

Después de registrarse (consulte Introducción a Amazon Polly) cree un usuario administrador para administrar la cuenta, incluida la creación de usuarios y la administración de permisos.

Puede crear un usuario que tenga permisos para todas las acciones de Amazon Polly. Piense en este usuario como un administrador específico del servicio para trabajar con Amazon Polly. Puede vincular la siguiente política de permisos con este usuario.

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

Ejemplo 2: Permitir todas las acciones de Amazon Polly excepto DeleteLexicon

La siguiente política de permisos concede permisos de usuario para realizar todas las acciones excepto `DeleteLexicon`, por lo que se deniegan los permisos para eliminar explícitamente elementos en todas las regiones.

```
{
   "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": "*"
      }
   ]
}
```

Ejemplo 3: Permitir DeleteLexicon

La siguiente política de permisos concede permisos de usuario para eliminar cualquier lexicón de su propiedad con independencia del proyecto o región en que se encuentre.

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

Ejemplo 4: Permitir DeleteLexicon en una región concreta

La siguiente política de permisos concede permisos de usuario para eliminar cualquier lexicón de cualquier proyecto de su propiedad que se encuentre en una única región (en este caso, us-east-2).

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

```
      "Sid": "AllowDeleteSpecifiedRegion",
      "Effect": "Allow",
      "Action": [
         "polly:DeleteLexicon"],
      "Resource": "arn:aws:polly:us-east-2:123456789012:lexicon/*"
      }
   ]
}
```

Ejemplo 5: Permitir DeleteLexicon un léxico específico

La siguiente política de permisos concede permisos de usuario para eliminar un lexicón específico de su propiedad (en este caso, myLexicon) de una región específica (en este caso, us-east-2).

```
{
  "Version": "2012-10-17",
  "Statement": [{
      "Sid": "AllowDeleteForSpecifiedLexicon",
      "Effect": "Allow",
      "Action": [
         "polly:DeleteLexicon"],
      "Resource": "arn:aws:polly:us-east-2:123456789012:lexicon/myLexicon"
      }
   ]
}
```

Permisos de la API de Amazon Polly: referencia de acciones, permisos y recursos

Cuando configure una política de permisos que pueda asociar a una identidad de IAM (políticas basadas en la identidad), puede utilizar la siguiente lista como referencia. La incluye cada operación de la API Amazon Polly, las acciones correspondientes para las que puede conceder permisos para realizar la acción y el AWS recurso para el que puede conceder los permisos. Las acciones se especifican en el campo `Action` de la política y el valor del recurso se especifica en el campo `Resource` de la política.

Puedes usar claves AWS de estado generales en tus políticas de Amazon Polly para expresar las condiciones. Para obtener una lista completa de las claves AWS de ancho, consulta las claves disponibles en la Guía del usuario de IAM.

> **Note**
>
> Para especificar una acción, use el prefijo `polly` seguido del nombre de operación de la API (por ejemplo, `polly:GetLexicon`).

Amazon Polly admite políticas basadas en la identidad para acciones en el nivel de recurso. Por tanto, el valor `Resource` viene especificado por el ARN. Por ejemplo: `arn:aws:polly:us-east-2:account-id:lexicon/*` como el valor de `Resource` especifica permisos en todos los lexicones propios incluidos en la región `us-east-2`.

Como Amazon Polly no admite los permisos de acciones en el nivel de recurso, la mayoría de las políticas utilizan un carácter comodín (*) como valor de `Resource`. Sin embargo, si es necesario limitar los permisos a una región específica, este carácter comodín se sustituye por el ARN correspondiente: `arn:aws:polly:region:account-id:lexicon/*`.

API de Amazon Polly y permisos necesarios para realizar acciones

Operación de la API: DeleteLexicon

Permisos necesarios (acción de la API): `polly:DeleteLexicon`

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

Operación de la API: DescribeVoices

Permisos necesarios (acción de la API): `polly:DescribeVoices`

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

Operación de la API: GetLexicon

Permisos necesarios (acción de la API): `polly:GetLexicon`

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

Operación de la API: ListLexicons

Permisos necesarios (acción de la API): `polly:ListLexicons`

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

Operación de la API: PutLexicon

Permisos necesarios (acción de la API): `polly:ListLexicons`

Recursos: *

Operación de la API: SynthesizeSpeech

Permisos necesarios (acción de la API): `polly:SynthesizeSpeech`

Recursos: *

Solución de problemas de identidad y acceso de Amazon Polly

Utilice la siguiente información para diagnosticar y solucionar los problemas comunes que es posible que surjan cuando se trabaja con Amazon Polly e IAM.

Temas

- No tengo autorización para realizar una acción en Amazon Polly
- No estoy autorizado a realizar tareas como: PassRole
- Quiero permitir que personas ajenas a mí accedan Cuenta de AWS a mis recursos de Amazon Polly

No tengo autorización para realizar una acción en Amazon Polly

Si recibe un error que indica que no tiene autorización para realizar una acción, las políticas se deben actualizar para permitirle realizar la acción.

En el siguiente ejemplo, el error se produce cuando el usuario de IAM `mateojackson` intenta utilizar la consola para consultar los detalles acerca de un recurso ficticio *`my-example-widget`*, pero no tiene los permisos ficticios `polly:`*`GetWidget`*.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 polly:GetWidget on resource: my-example-widget
```

En este caso, la política del usuario `mateojackson` debe actualizarse para permitir el acceso al recurso *`my-example-widget`* mediante la acción `polly:`*`GetWidget`*.

Si necesita ayuda, póngase en contacto con su administrador. AWS El gestionador es la persona que le proporcionó las credenciales de inicio de sesión.

No estoy autorizado a realizar tareas como: PassRole

Si recibe un error que indica que no tiene autorización para realizar la acción `iam:PassRole`, sus políticas deben actualizarse para permitirle pasar un rol a Amazon Polly.

Algunas Servicios de AWS permiten transferir una función existente a ese servicio en lugar de crear una nueva función de servicio o una función vinculada a un servicio. Para ello, debe tener permisos para transferir el rol al servicio.

En el siguiente ejemplo, el error se produce cuando un usuario de IAM denominado `marymajor` intenta utilizar la consola para realizar una acción en Amazon Polly. Sin embargo, la acción requiere que el servicio cuente con permisos que otorguen un rol de servicio. Mary no tiene permisos para transferir el rol al servicio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

En este caso, las políticas de Mary se deben actualizar para permitirle realizar la acción `iam:PassRole`.

Si necesita ayuda, póngase en contacto con su administrador. AWS El gestionador es la persona que le proporcionó las credenciales de inicio de sesión.

Quiero permitir que personas ajenas a mí accedan Cuenta de AWS a mis recursos de Amazon Polly

Puede crear un rol que los usuarios de otras cuentas o las personas externas a la organización puedan utilizar para acceder a sus recursos. Puede especificar una persona de confianza para que asuma el rol. En el caso de los servicios que admiten políticas basadas en recursos o listas de control de acceso (ACLs), puede usar esas políticas para permitir que las personas accedan a sus recursos.

Para obtener más información, consulte lo siguiente:

- Para saber si Amazon Polly admite estas características, consulte Cómo funciona Amazon Polly con IAM.
- Para obtener información sobre cómo proporcionar acceso a los recursos de su Cuentas de AWS propiedad, consulte Proporcionar acceso a un usuario de IAM en otro de su propiedad en la Cuenta de AWS Guía del usuario de IAM.

- Para obtener información sobre cómo proporcionar acceso a tus recursos a terceros Cuentas de AWS, consulta Cómo proporcionar acceso a recursos que Cuentas de AWS son propiedad de terceros en la Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso mediante una federación de identidades, consulta Proporcionar acceso a usuarios autenticados externamente (identidad federada) en la Guía del usuario de IAM.
- Para conocer sobre la diferencia entre las políticas basadas en roles y en recursos para el acceso entre cuentas, consulte Acceso a recursos entre cuentas en IAM en la Guía del usuario de IAM.

Registro y monitoreo en Amazon Polly

El monitoreo es una parte importante del mantenimiento de la fiabilidad, la disponibilidad y el rendimiento de las aplicaciones de su Amazon Polly . Para supervisar las llamadas a la API Amazon Polly, puede utilizar. AWS CloudTrail Para supervisar el estado de sus trabajos, utilice Amazon CloudWatch Logs.

- Amazon CloudWatch Alarms: al usar CloudWatch las alarmas, observas una única métrica durante un período de tiempo que especifiques. Si la métrica supera un umbral determinado, se envía una notificación a un tema o AWS Auto Scaling política de Amazon Simple Notification Service. CloudWatchlas alarmas no invocan acciones cuando una métrica se encuentra en un estado determinado. En su lugar, el estado debe haber cambiado y debe mantenerse durante el número de periodos especificado. Para obtener más información, consulte Integración CloudWatch con Amazon Polly.
- CloudTrail logs: CloudTrail proporciona un registro de las acciones realizadas por un usuario, un rol o un AWS servicio en Amazon Polly. Con la información recopilada por CloudTrail, puede determinar la solicitud que se realizó a Amazon Polly. También puede identificar la dirección IP desde la que se realizó la solicitud, quién realizó la solicitud, cuándo se realizó y detalles adicionales. Para obtener más información, consulte Registrar llamadas a la API Amazon Polly con AWS CloudTrail.

Validación de la conformidad para Amazon Polly

Los auditores externos evalúan la seguridad y el cumplimiento de Amazon Polly como parte de varios programas de AWS cumplimiento. Estos incluyen SOC, PCI, FedRAMP, HIPAA y otros.

Para ver una lista de AWS los servicios incluidos en el ámbito de los programas de conformidad específicos, consulte AWS Servicios incluidos en el ámbito de aplicación por programa de conformidad y en AWS los servicios incluidos . Para obtener información general, consulte Programas de AWS cumplimiento > Programas AWS .

Puede descargar informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulta Descarga de informes en AWS Artifact.

Su responsabilidad de conformidad al utilizar Amazon Polly viene determinada por la confidencialidad de sus datos, los objetivos de cumplimiento de su empresa y las leyes y reglamentos aplicables. AWS proporciona los siguientes recursos para ayudar con el cumplimiento:

- Security and Compliance Quick Start Guides (Guías de inicio rápido de seguridad y conformidad) (Guías de inicio rápido de seguridad y conformidad): Estas guías de implementación analizan las consideraciones en materia de arquitectura y proporcionan los pasos para implementar los entornos de referencia centrados en la seguridad y la conformidad en AWS.
- Documento técnico sobre cómo diseñar una arquitectura basada en la seguridad y el cumplimiento de la HIPAA: en este documento técnico se describe cómo pueden utilizar las empresas para crear aplicaciones que cumplan con la HIPAA. AWS
- AWS Recursos de cumplimiento Recursos de de trabajo y guías puede aplicarse a su sector y ubicación.
- Evaluación de los recursos con las reglas de la guía para AWS Config desarrolladores: el AWS Config servicio evalúa en qué medida las configuraciones de los recursos cumplen con las prácticas internas, las directrices del sector y las normas.
- AWS Security Hub— Este AWS servicio proporciona una visión integral del estado de su seguridad AWS que le ayuda a comprobar su conformidad con los estándares y las mejores prácticas del sector de la seguridad.

Resiliencia en Amazon Polly

La infraestructura AWS global se basa en AWS regiones y zonas de disponibilidad. AWS Las regiones proporcionan varias zonas de disponibilidad aisladas y separadas físicamente, que están conectadas mediante redes de baja latencia, alto rendimiento y alta redundancia. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre las zonas sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de uno o varios centros de datos.

Para obtener más información sobre AWS las regiones y las zonas de disponibilidad, consulte Infraestructura global.AWS

Seguridad de la infraestructura de Amazon Polly

Como servicio gestionado, Amazon Polly está protegido por los procedimientos de seguridad de la red AWS global que se describen en el documento técnico Amazon Web Services: Descripción general de los procesos de seguridad.

Utiliza las llamadas a la API AWS publicadas para acceder a Amazon Polly a través de la red. Los clientes deben ser compatibles con la seguridad de la capa de transporte (TLS) 1.0 o una versión posterior. Recomendamos TLS 1.2 o una versión posterior. Los clientes también deben ser compatibles con conjuntos de cifrado con confidencialidad directa total (PFS) tales como Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Además, las solicitudes deben estar firmadas mediante un ID de clave de acceso y una clave de acceso secreta que esté asociada a una entidad principal de IAM. También puedes utilizar AWS Security Token Service (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

Prácticas recomendadas de seguridad para Amazon Polly

La confianza, privacidad y seguridad de su contenido son nuestras máximas prioridades. Implementamos controles técnicos y físicos responsables y sofisticados diseñados para evitar el acceso no autorizado o la divulgación de su contenido y garantizar que nuestro uso cumple con nuestros compromisos con usted. Para obtener más información, consulte Preguntas frecuentes sobre la privacidad de datos de AWS.

Amazon Polly no retiene el contenido de los envíos de texto.

Para obtener una visión amplia de la AWS seguridad, que incluye el cumplimiento, las pruebas de penetración, los boletines y los recursos, visite el sitio web de AWS Cloud Security.

Uso de Amazon Polly con los puntos de conexión de VPC de la interfaz

Si utiliza Amazon Virtual Private Cloud (Amazon VPC) para alojar sus AWS recursos, puede establecer una conexión privada entre su VPC y Amazon Polly. Puede utilizar esta conexión para sintetizar voz con Amazon Polly sin pasar por la red pública de Internet.

Amazon VPC es un AWS servicio que puede utilizar para lanzar AWS recursos en una red virtual que usted defina. Con una VPC, puede controlar la configuración de la red, como el rango de direcciones IP, las subredes, las tablas de ruteo y las gateways de red. Para conectar su VPC a Amazon Polly, debe definir un punto de conexión de VPC de la interfaz para Amazon Polly. Este tipo de punto de conexión le permite conectar la VPC a los servicios de Servicios de AWS. Con el punto de conexión, se ofrece conectividad escalable de confianza con Amazon Polly sin necesidad de utilizar una puerta de enlace de Internet, una instancia de traducción de direcciones de red (NAT) o una conexión de VPN. Para obtener más información, consulte ¿Qué es Amazon VPC en la Guía del usuario de Amazon VPC.

Los puntos finales de VPC de interfaz funcionan con una AWS tecnología que permite la comunicación privada entre direcciones IP privadas Servicios de AWS mediante el uso de una interfaz de red elástica. AWS PrivateLink Para obtener más información, consulte New - AWS PrivateLink for. Servicios de AWS

Los siguientes pasos son para usuarios de Amazon VPC. Para obtener más información, consulte Introducción en la Guía del usuario de Amazon VPC.

Disponibilidad

Los puntos de conexión de VPC son compatibles con todas las regiones en las que Amazon Polly es compatible. Para obtener más información sobre AWS las regiones y las zonas de disponibilidad, consulte Infraestructura AWS global.

Creación de una política de puntos de conexión de VPC para Amazon Polly

Para comenzar a utilizar Amazon Polly con su VPC, cree un punto de conexión de VPC de interfaz para Amazon Polly. El servicio que debe elegir es com.amazonaws. ***Region***.polly. No necesita cambiar la configuración de Amazon Polly. Para obtener más información, consulte Creación de un punto de conexión de interfaz en la Guía del usuario de Amazon VPC.

Comprobación de la conexión entre la VPC y Amazon Polly

Una vez creado el punto de conexión, puede probar la conexión.

Para probar la conexión entre su VPC y su punto de conexión de Amazon Polly

1. Conéctese a una EC2 instancia de Amazon que resida en su VPC. Para obtener información sobre la conexión, consulte Conectarse a su instancia de Linux o Conectarse a su instancia de Windows en la EC2 documentación de Amazon.
2. Desde la instancia, utilice `aws polly describe-voices` de la lista AWS CLI de voces de Amazon Polly disponibles.

Si la respuesta al comando incluye la lista de voces disponibles de Amazon Polly, el comando se ha realizado correctamente y el punto de conexión de VPC funciona.

Control del acceso al punto de conexión de VPC de Amazon Polly

Una política de punto de conexión de VPC es una política de recursos de IAM que puede asociar a un punto de conexión cuando crea o modifica el punto de conexión. Si no asocia una política al crear un punto de conexión, se asociará automáticamente una política predeterminada que conceda acceso completo al servicio. Una política de punto de conexión no anula ni sustituye a las políticas de usuario de IAM ni a las políticas específicas del servicio. Se trata de una política independiente para controlar el acceso desde el punto de conexión al servicio especificado.

Las políticas de punto de conexión deben escribirse en formato JSON.

Para obtener más información, consulte Controlar el acceso a servicios con puntos de enlace de la VPC en la Guía del usuario de Amazon VPC.

A continuación, se muestra un ejemplo de una política de punto de conexión para Amazon Polly. Esta política permite a los usuarios que se conectan a Amazon Polly a través de la VPC describir voces y sintetizar voz con Amazon Polly e impide realizar otras acciones de Amazon Polly.

```
{
  "Statement": [
    {
      "Sid": "SynthesisAndDescribeVoicesOnly",
      "Principal": "*",
      "Action": [
```

```
        "polly:DescribeVoices",
        "polly:SynthesizeSpeech"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Modificación de la política de punto de conexión de VPC para Amazon Polly

1. Abra la consola de Amazon VPC en https://console.aws.amazon.com /vpc.
2. En el panel de navegación, elija Puntos de conexión.
3. Si todavía no ha creado el punto de conexión para Amazon Polly, elija Crear punto de conexión. A continuación, seleccione com.amazonaws. *Region*.polly y selecciona Crear punto final.
4. Selecciona com.amazonaws. ***Region***.polly endpoint y selecciona la pestaña Política en la mitad inferior de la pantalla.
5. Elija Editar política y realice los cambios en la política.

Compatibilidad con las claves de contexto de la VPC

Amazon Polly admite las claves de `aws:SourceVpce` contexto `aws:SourceVpc` y las claves que pueden limitar el acceso a puntos de enlace de VPC específicos VPCs o específicos. Estas claves funcionan solo cuando el usuario utiliza puntos de enlace de la VPC. Con el fin de obtener más información, consulte Claves disponibles para algunos servicios en la Guía del usuario de IAM.

Registrar llamadas a la API Amazon Polly con AWS CloudTrail

Amazon Polly está integrado con AWS CloudTrail un servicio que proporciona un registro de las acciones realizadas por un usuario, un rol o un AWS servicio en Amazon Polly. CloudTrail captura todas las llamadas a la API de Amazon Polly como eventos. Las llamadas capturadas incluyen las llamadas desde la consola de Amazon Polly y las llamadas desde el código a las operaciones de la API de Amazon Polly. Si crea una ruta, puede habilitar la entrega continua de CloudTrail eventos a un bucket de Amazon S3, incluidos los eventos de Amazon Polly. Si no configura una ruta, podrá ver los eventos más recientes en la CloudTrail consola, en el historial de eventos. Con la información recopilada por CloudTrail, puedes determinar la solicitud que se realizó a Amazon Polly, la dirección IP desde la que se realizó la solicitud, quién la hizo, cuándo se realizó y detalles adicionales.

Para obtener más información CloudTrail, incluido cómo configurarlo y habilitarlo, consulte la Guía del AWS CloudTrail usuario.

Información sobre Amazon Polly en CloudTrail

CloudTrail está habilitada en tu AWS cuenta al crearla. Cuando se produce una actividad de eventos admitida en Amazon Polly, esa actividad se registra en un CloudTrail evento junto con otros eventos de AWS servicio en el historial de eventos. Puede ver, buscar y descargar los eventos recientes en su AWS cuenta. Para obtener más información, consulte Visualización de eventos con el historial de CloudTrail eventos.

Para tener un registro continuo de los eventos de tu AWS cuenta, incluidos los eventos de Amazon Polly, crea una ruta. Un rastro permite CloudTrail entregar archivos de registro a un bucket de Amazon S3. De forma predeterminada, cuando crea una ruta en la consola, la ruta se aplica a todas AWS las regiones. La ruta registra los eventos de todas las regiones de la AWS partición y envía los archivos de registro al bucket de Amazon S3 que especifique. Además, puede configurar otros AWS servicios para analizar más a fondo los datos de eventos recopilados en los CloudTrail registros y actuar en función de ellos. Para más información, consulte los siguientes temas:

- Introducción a la creación de registros de seguimiento
- CloudTrail Integraciones y servicios compatibles
- Configuración de las notificaciones de Amazon SNS para CloudTrail

- Recibir archivos de CloudTrail registro de varias regiones y recibir archivos de CloudTrail registro de varias cuentas

Amazon Polly admite el registro de las siguientes acciones como eventos en los archivos de CloudTrail registro:

- DeleteLexicon
- DescribeVoices
- GetLexicon
- GetSpeechSynthesisTask
- ListLexicons
- ListSpeechSynthesisTasks
- PutLexicon
- StartSpeechSynthesisTask
- SynthesizeSpeech

Cada entrada de registro o evento contiene información sobre quién generó la solicitud. La información de identidad del usuario le ayuda a determinar lo siguiente:

- Si la solicitud se realizó con credenciales de usuario root o de usuario AWS Identity and Access Management (IAM).
- Si la solicitud se realizó con credenciales de seguridad temporales de un rol o fue un usuario federado.
- Si la solicitud la realizó otro AWS servicio.

Para obtener más información, consulte el Elemento userIdentity de CloudTrail .

Ejemplo: Entradas del archivo de registro de Amazon Polly

Un rastro es una configuración que permite la entrega de eventos como archivos de registro a un bucket de Amazon S3 que usted especifique. CloudTrail Los archivos de registro contienen una o más entradas de registro. Un evento representa una solicitud única de cualquier fuente e incluye información sobre la acción solicitada, la fecha y la hora de la acción, los parámetros de la solicitud,

etc. CloudTrail Los archivos de registro no son un registro ordenado de las llamadas a la API pública, por lo que no aparecen en ningún orden específico.

En el siguiente ejemplo, se muestra una entrada de CloudTrail registro que demuestra que`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"
            }
        }

    ]
```

```
}
```

Integración CloudWatch con Amazon Polly

Cuando interactúas con Amazon Polly, envía las siguientes métricas y dimensiones a CloudWatch cada minuto. Puede utilizar los siguientes procedimientos para consultar las métricas de Amazon Polly.

Puedes monitorizar Amazon Polly con Amazon Polly CloudWatch, que recopila y procesa datos sin procesar de Amazon Polly para convertirlos en métricas legibles prácticamente en tiempo real. Estas estadísticas se registran durante un periodo de dos semanas, de forma que pueda obtener acceso a `historical information` y obtener una mejor perspectiva sobre el desempeño de la aplicación web o el servicio. De forma predeterminada, los datos de las métricas de Amazon Polly se envían a intervalos CloudWatch de 1 minuto. Para obtener más información, consulta Qué es Amazon CloudWatch en la Guía del CloudWatch usuario de Amazon.

Obtención de CloudWatch métricas (consola)

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Métricas.
3. En el panel CloudWatch Métricas por categoría, en la categoría de métricas de Amazon Polly, selecciona una categoría de métricas y, a continuación, en el panel superior, desplázate hacia abajo para ver la lista completa de métricas.

Obtener CloudWatch las métricas en AWS CLI

El siguiente código muestra las métricas disponibles para Amazon Polly.

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

El comando anterior devuelve una lista de métricas de Amazon Polly similar a la siguiente. El elemento `MetricName` identifica qué es la métrica.

```
{
    "Metrics": [
        {
            "Namespace": "AWS/Polly",
            "Dimensions": [
```

```
            {
                "Name": "Operation",
                "Value": "SynthesizeSpeech"
            }
        ],
        "MetricName": "ResponseLatency"
    },
    {
        "Namespace": "AWS/Polly",
        "Dimensions": [
            {
                "Name": "Operation",
                "Value": "SynthesizeSpeech"
            }
        ],
        "MetricName": "RequestCharacters"
    }
```

Para obtener más información, consulta GetMetricStatisticsla referencia de la CloudWatch API de Amazon.

Métricas de Amazon Polly

Amazon Polly genera las siguientes métricas para cada solicitud. Estas métricas se agregan y se envían en intervalos de un minuto a CloudWatch donde estén disponibles.

Métrica	Descripción
`RequestCharacters`	El número de caracteres en la solicitud. Se trata únicamente de caracteres facturables y no incluye las etiquetas SSML. Dimensión válida: operación Estadísticas válidas: mínimo, máximo, promedio SampleCount, suma Unidad: recuento
`ResponseLatency`	La latencia entre el momento en que se realizó la solicitud y el inicio de la respuesta de transmisión.

Métrica	Descripción
	Dimensiones válidas: operación Estadísticas válidas: mínimo, máximo, promedio, SampleCount Unidad: microsegundos
2XXCount	El código HTTP de nivel 200 se devolvió tras una respuesta correcta. Dimensiones válidas: operación Estadísticas válidas: media SampleCount, suma Unidad: recuento
4XXCount	Se devuelve un código de error de nivel HTTP 400 en caso de error. Por cada respuesta correcta, se emite un cero (0). Dimensiones válidas: operación Estadísticas válidas: promedio SampleCount, suma Unidad: recuento
5XXCount	Se devuelve un código de error de nivel HTTP 500 en caso de error. Por cada respuesta correcta, se emite un cero (0). Dimensiones válidas: operación Estadísticas válidas: promedio SampleCount, suma Unidad: recuento

Dimensiones para las métricas de Amazon Polly

Las métricas de Amazon Polly utilizan el espacio de nombres AWS/Polly y proporcionan métricas para la siguiente dimensión:

Dimensión	Descripción
`Operation`	Las métricas se agrupan según el método de API al que se refieren. Los valores posibles son `SynthesizeSpeech` , `PutLexicon` , `DescribeVoices` , etc.

Referencia de la API de Amazon Polly

Amazon Polly es un servicio web que facilita la síntesis de voz a partir de texto.

El servicio Amazon Polly proporciona operaciones de API para sintetizar voz de alta calidad a partir de texto sin formato y el lenguaje de marcado de síntesis de voz (SSML), además de administrar lexicones de pronunciaciones que le permiten obtener los mejores resultados para el dominio de la aplicación.

Las llamadas al API autenticadas deben firmarse utilizando el Proceso de firma Signature Version 4. Para obtener más información, consulte Firmar solicitudes de AWS API en. Referencia general de Amazon Web Services

Temas

- Acciones
- Data Types
- Errores comunes
- Parámetros comunes

Acciones

Se admiten las siguientes acciones:

- DeleteLexicon
- DescribeVoices
- GetLexicon
- GetSpeechSynthesisTask
- ListLexicons
- ListSpeechSynthesisTasks
- PutLexicon
- StartSpeechSynthesisTask
- SynthesizeSpeech

DeleteLexicon

Elimina el lexicón de pronunciación especificado y almacenado en una región de Región de AWS. Un léxico que se ha eliminado no está disponible para la síntesis de voz, ni es posible recuperarlo mediante la `GetLexicon` o `ListLexicon` APIs.

Para obtener más información, consulte Administración de lexicones.

Sintaxis de la solicitud

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

Parámetros de solicitud del URI

La solicitud utiliza los siguientes parámetros URI.

LexiconName

El nombre del lexicón que se va a eliminar. Debe ser un lexicón existente en la región.

Patrón: `[0-9A-Za-z]{1,20}`

Obligatorio: sí

Cuerpo de la solicitud

La solicitud no tiene un cuerpo de la solicitud.

Sintaxis de la respuesta

```
HTTP/1.1 200
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200 con un cuerpo HTTP vacío.

Errores

LexiconNotFoundException

Amazon Polly no encuentra el lexicón especificado. Esto puede deberse a la falta de un lexicón, a que su nombre esté mal escrito o a que se especifique un lexicón que se encuentra en una región diferente.

Compruebe que el lexicón existe, se encuentra en la región (consulte ListLexicons) y que ha escrito su nombre correctamente. Inténtelo de nuevo.

Código de estado HTTP: 404

ServiceFailureException

Una condición desconocida ha provocado un fallo en el servicio.

Código de estado HTTP: 500

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulte lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

DescribeVoices

Devuelve la lista de voces que están disponibles para su uso cuando se solicita la síntesis de voz. Cada voz habla un idioma específico, es masculina o femenina, y se identifica mediante un identificador, que es la versión ASCII del nombre de la voz.

Al sintetizar la locución (`SynthesizeSpeech`), debe proporcionar el identificador de voz de la voz que desee de la lista de voces devueltas por `DescribeVoices`.

Por ejemplo, quiere que su aplicación lectora de noticias lea las noticias en un idioma específico, pero que dé al usuario la opción de elegir la voz. Con la operación `DescribeVoices`, puede proporcionar al usuario una lista de las voces disponibles entre las que elegir.

También puede especificar un código de idioma para filtrar las voces disponibles. Por ejemplo, si especifica `en-US`, la operación devuelve una lista de todas las voces disponibles para el idioma inglés de EE. UU.

Esto operación requiere permisos para realizar la acción `polly:DescribeVoices`.

Sintaxis de la solicitud

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

Parámetros de solicitud del URI

La solicitud utiliza los siguientes parámetros URI.

Engine

Especifica el motor (`standard`, `neural`, `long-form` o `generative`) que utiliza Amazon Polly al procesar el texto de entrada para la síntesis de voz.

Valores válidos: `standard | neural | long-form | generative`

IncludeAdditionalLanguageCodes

Valor booleano que indica si se deben devolver voces bilingües que utilicen el idioma especificado como idioma adicional. Por ejemplo, si solicita todos los idiomas que utilizan el inglés estadounidense (en-US) y hay una voz italiana que habla tanto italiano (it-IT) como inglés estadounidense, esa voz se incluirá si especifica `yes`, pero no si especifica no.

LanguageCode

La etiqueta de identificación del idioma (código ISO 639 para el nombre del idioma y código de país ISO 3166) para filtrar la lista de voces devueltas. Si no especifica este parámetro opcional, se devolverán todas las voces disponibles.

Valores válidos: `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 de paginación opaco devuelto por la operación `DescribeVoices` anterior. Si está presente, indica dónde continuar con el anuncio.

Limitaciones de longitud: longitud mínima de 0. La longitud máxima es de 4 096 caracteres.

Cuerpo de la solicitud

La solicitud no tiene un cuerpo de la solicitud.

Sintaxis de la respuesta

```
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" ]
      }
   ]
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

NextToken

El token de paginación que se utilizará en la próxima solicitud para continuar con la lista de voces. Se devuelve `NextToken` solo si la respuesta está truncada.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 0. La longitud máxima es de 4 096 caracteres.

Voices

Una lista de voces con sus propiedades.

Tipo: matriz de objetos Voice

Errores

InvalidNextTokenException

El no NextToken es válido. Asegúrese de que esté bien escrito y, a continuación, vuelva a intentarlo.

Código de estado HTTP: 400

ServiceFailureException

Una condición desconocida ha provocado un fallo en el servicio.

Código de estado HTTP: 500

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- Interfaz de la línea de comandos de AWS

- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

GetLexicon

Devuelve el contenido del lexicón de pronunciación especificado y almacenado en una Región de AWS. Para obtener más información, consulte Administración de lexicones.

Sintaxis de la solicitud

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

Parámetros de solicitud del URI

La solicitud utiliza los siguientes parámetros URI.

LexiconName

El nombre del lexicón.

Patrón: `[0-9A-Za-z]{1,20}`

Obligatorio: sí

Cuerpo de la solicitud

La solicitud no tiene un cuerpo de la solicitud.

Sintaxis de la respuesta

```
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
   }
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

Lexicon

Objeto de lexicón que proporciona el nombre y el contenido de la cadena del lexicón.

Tipo: objeto Lexicon

LexiconAttributes

Metadatos del lexicón, incluido el alfabeto fonético utilizado, el código de idioma, el ARN del lexicón, el número de lexemas definidos en el lexicón y el tamaño del lexicón en bytes.

Tipo: objeto LexiconAttributes

Errores

LexiconNotFoundException

Amazon Polly no encuentra el lexicón especificado. Esto puede deberse a la falta de un lexicón, a que su nombre esté mal escrito o a que se especifique un lexicón que se encuentra en una región diferente.

Compruebe que el lexicón existe, se encuentra en la región (consulte ListLexicons) y que ha escrito su nombre correctamente. Inténtelo de nuevo.

Código de estado HTTP: 404

ServiceFailureException

Una condición desconocida ha provocado un fallo en el servicio.

Código de estado HTTP: 500

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

GetSpeechSynthesisTask

Recupera un SpeechSynthesisTask objeto específico en función de su identificador de tarea. Este objeto contiene información sobre la tarea de síntesis de voz determinada, incluido el estado de la tarea, y un enlace al bucket de S3 que contiene el resultado de la tarea.

Sintaxis de la solicitud

```
GET /v1/synthesisTasks/TaskId HTTP/1.1
```

Parámetros de solicitud del URI

La solicitud utiliza los siguientes parámetros URI.

TaskId

El identificador generado por Amazon Polly para una tarea de síntesis de voz.

Patrón: `^[a-zA-Z0-9_-]{1,100}$`

Obligatorio: sí

Cuerpo de la solicitud

La solicitud no tiene un cuerpo de la solicitud.

Sintaxis de la respuesta

```
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"
   }
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

SynthesisTask

SynthesisTask objeto que proporciona información de la tarea solicitada, incluido el formato de salida, la hora de creación, el estado de la tarea, etc.

Tipo: objeto SynthesisTask

Errores

InvalidTaskIdException

El identificador de tarea proporcionado no es válido. Proporcione un identificador de tarea válido e inténtelo de nuevo.

Código de estado HTTP: 400

ServiceFailureException

Una condición desconocida ha provocado un fallo en el servicio.

Código de estado HTTP: 500

SynthesisTaskNotFoundException

No se encuentra la tarea de síntesis de voz con el identificador de tarea solicitado.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

ListLexicons

Devuelve una lista de lexicones de pronunciación almacenados en una región de Región de AWS. Para obtener más información, consulte Administración de lexicones.

Sintaxis de la solicitud

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

Parámetros de solicitud del URI

La solicitud utiliza los siguientes parámetros URI.

NextToken

Un token de paginación opaco devuelto por una operación `ListLexicons` anterior. Si está presente, indica dónde continuar con la lista de lexicones.

Limitaciones de longitud: longitud mínima de 0. La longitud máxima es de 4 096 caracteres.

Cuerpo de la solicitud

La solicitud no tiene un cuerpo de la solicitud.

Sintaxis de la respuesta

```
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"
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

Lexicons

Una lista de nombres y atributos de los lexicones.

Tipo: matriz de objetos LexiconDescription

NextToken

El token de paginación que se utilizará en la próxima solicitud para continuar con la lista de lexicones. `NextToken` se devuelve solo si la respuesta está truncada.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 0. La longitud máxima es de 4 096 caracteres.

Errores

InvalidNextTokenException

El no NextToken es válido. Asegúrese de que esté bien escrito y, a continuación, vuelva a intentarlo.

Código de estado HTTP: 400

ServiceFailureException

Una condición desconocida ha provocado un fallo en el servicio.

Código de estado HTTP: 500

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

ListSpeechSynthesisTasks

Devuelve una lista de SpeechSynthesisTask objetos ordenados por su fecha de creación. Esta operación puede filtrar las tareas por su estado, por ejemplo, lo que permite a los usuarios enumerar solo las tareas que se han completado.

Sintaxis de la solicitud

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

Parámetros de solicitud del URI

La solicitud utiliza los siguientes parámetros URI.

MaxResults

Número máximo de tareas de síntesis de voz devueltas en una operación de lista.

Rango válido: valor mínimo de 1. Valor máximo de 100.

NextToken

El token de paginación que se utilizará en la siguiente solicitud para continuar con la lista de tareas de síntesis de voz.

Limitaciones de longitud: longitud mínima de 0. La longitud máxima es de 4 096 caracteres.

Status

Estado de las tareas de síntesis de voz devueltas en una operación de lista

Valores válidos: `scheduled | inProgress | completed | failed`

Cuerpo de la solicitud

La solicitud no tiene un cuerpo de la solicitud.

Sintaxis de la respuesta

```
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"
      }
   ]
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

NextToken

Un token de paginación opaco devuelto por la operación de lista anterior de esta solicitud. Si está presente, indica dónde continuar con el anuncio.

Tipo: cadena

Limitaciones de longitud: longitud mínima de 0. La longitud máxima es de 4 096 caracteres.

SynthesisTasks

Lista de SynthesisTask objetos que proporciona información de la tarea especificada en la solicitud de lista, incluidos el formato de salida, la hora de creación, el estado de la tarea, etc.

Tipo: matriz de objetos SynthesisTask

Errores

InvalidNextTokenException

El no NextToken es válido. Asegúrese de que esté bien escrito y, a continuación, vuelva a intentarlo.

Código de estado HTTP: 400

ServiceFailureException

Una condición desconocida ha provocado un fallo en el servicio.

Código de estado HTTP: 500

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

PutLexicon

Almacena un lexicón de pronunciación en una Región de AWS. Si ya existe un lexicón con el mismo nombre en la región, el nuevo léxico lo sobrescribe. Las operaciones del léxico tienen una consistencia eventual, por lo que puede pasar algún tiempo antes de que el léxico esté disponible para la SynthesizeSpeech operación.

Para obtener más información, consulte Administración de lexicones.

Sintaxis de la solicitud

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

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

Parámetros de solicitud del URI

La solicitud utiliza los siguientes parámetros URI.

LexiconName

El nombre del lexicón. El nombre debe seguir el formato expreso normal [0-9A-Za-z] {1,20}. Es decir, el nombre es una cadena alfanumérica que distingue mayúsculas de minúsculas y tiene una longitud máxima de 20 caracteres.

Patrón: `[0-9A-Za-z]{1,20}`

Obligatorio: sí

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

Content

Contenido del lexicón PLS en forma de cadena de datos.

Tipo: cadena

Obligatorio: sí

Sintaxis de la respuesta

```
HTTP/1.1 200
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200 con un cuerpo HTTP vacío.

Errores

InvalidLexiconException

Amazon Polly no encuentra el lexicón especificado. Asegúrese de que el nombre del lexicón esté bien escrito y, a continuación, vuelva a intentarlo.

Código de estado HTTP: 400

LexiconSizeExceededException

Esta operación superaría el tamaño máximo del lexicón especificado.

Código de estado HTTP: 400

MaxLexemeLengthExceededException

Esta operación superaría el tamaño máximo del lexema.

Código de estado HTTP: 400

MaxLexiconsNumberExceededException

Esta operación superaría el número máximo de lexicones.

Código de estado HTTP: 400

ServiceFailureException

Una condición desconocida ha provocado un fallo en el servicio.

Código de estado HTTP: 500

UnsupportedPlsAlphabetException

El alfabeto especificado en el lexicón no es un alfabeto compatible. Los valores válidos son `x-sampa` y `ipa`.

Código de estado HTTP: 400

UnsupportedPlsLanguageException

El idioma especificado en el lexicón no es compatible. Para ver una lista de los idiomas compatibles, consulte Atributos del lexicón.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulte lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

StartSpeechSynthesisTask

Permite crear una tarea de síntesis asíncrona, iniciando una nueva `SpeechSynthesisTask`. Esta operación requiere toda la información estándar necesaria para la síntesis de voz, además del nombre de un bucket de Amazon S3 para que el servicio almacene el resultado de la tarea de síntesis y dos parámetros opcionales (`OutputS3KeyPrefix` y `SnsTopicArn`). Una vez creada la tarea de síntesis, esta operación devolverá un objeto `SpeechSynthesisTask`, que incluirá un identificador de la tarea y su estado actual. El objeto `SpeechSynthesisTask` estará disponible durante 72 horas después de iniciar la tarea de síntesis asíncrona.

Sintaxis de la solicitud

```
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"
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

Engine

Especifica el motor (`standard`, `neural`, `long-form` o `generative`) que utilizará Amazon Polly al procesar el texto de entrada para la síntesis de voz. El uso de una voz que no sea compatible con el motor seleccionado provocará un error.

Tipo: cadena

Valores válidos: `standard | neural | long-form | generative`

Obligatorio: no

LanguageCode

Código de idioma opcional para la solicitud de síntesis de voz. Esto solo es necesario si se utiliza una voz bilingüe, como Aditi, que se puede utilizar tanto en inglés indio (en-IN) como en hindi (hi-IN).

Si se utiliza una voz bilingüe y no se especifica ningún código de idioma, Amazon Polly utilizará el idioma predeterminado de la voz bilingüe. El idioma predeterminado de cualquier voz es el que devuelve la DescribeVoicesoperación para el `LanguageCode` parámetro. Por ejemplo, si no se especifica ningún código de idioma, Aditi utilizará el inglés indio en lugar del hindi.

Tipo: cadena

Valores válidos: `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`

Obligatorio: no

LexiconNames

Enumere uno o más nombres de lexicones de pronunciación que desee que el servicio aplique durante la síntesis. Los lexicones solamente se aplican al texto si el idioma especificado en ellos es el mismo que el idioma elegido.

Tipo: matriz de cadenas

Miembros de la matriz: número máximo de 5 elementos.

Patrón: `[0-9A-Za-z]{1,20}`

Obligatorio: no

OutputFormat

Formato en el que se codificará la salida devuelta. Para la transmisión de audio, será mp3, ogg_vorbis o pcm. Para las marcas de voz, será json.

Tipo: cadena

Valores válidos: `json | mp3 | ogg_vorbis | pcm`

Obligatorio: sí

OutputS3BucketName

Nombre del bucket de Amazon S3 en el que se guardará el archivo de salida.

Tipo: cadena

Patrón: `^[a-z0-9][\.\-a-z0-9]{1,61}[a-z0-9]$`

Obligatorio: sí

OutputS3KeyPrefix

El prefijo de clave de Amazon S3 para el archivo de voz de salida.

Tipo: cadena

Patrón: `^[0-9a-zA-Z\/\!\-_\.\*\'\(\):;\$@=+\,\?&]{0,800}$`

Obligatorio: no

SampleRate

La frecuencia de audio especificada en Hz.

Los valores válidos para mp3 y ogg_vorbis son “8000”, “16 000”, “22 050” y “24 000”. El valor predeterminado para las voces estándar es “22 050". El valor predeterminado para las voces neuronales es "24 000". El valor predeterminado para las voces de formato largo es “24 000”. El valor predeterminado para las voces generativas es "24 000".

Los valores válidos para el pcm son "8000" y "16 000". El valor por defecto es "16 000".

Tipo: cadena

Requerido: no

SnsTopicArn

El ARN para el tema de SNS se usa opcionalmente para proporcionar una notificación de estado para una tarea de síntesis de voz.

Tipo: cadena

Patrón: `^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)$`

Obligatorio: no

SpeechMarkTypes

El tipo de marcas de voz devueltas para el texto de entrada.

Tipo: matriz de cadenas

Miembros de la matriz: número máximo de 4 elementos.

Valores válidos: `sentence | ssml | viseme | word`

Obligatorio: no

Text

El texto de entrada que se va a sintetizar. Si especifica ssml como TextType, siga el formato SSML para el texto de entrada.

Tipo: cadena

Obligatorio: sí

TextType

Especifica si el texto de entrada es texto sin formato o SSML. El valor predeterminado es texto sin formato.

Tipo: cadena

Valores válidos: `ssml | text`

Obligatorio: no

VoiceId

ID de voz que se usará para la síntesis.

Tipo: cadena

Valores válidos: `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`

Obligatorio: sí

Sintaxis de la respuesta

```
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"
   }
}
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

El servicio devuelve los datos siguientes en formato JSON.

SynthesisTask

SynthesisTask objeto que proporciona información y atributos sobre una tarea de síntesis de voz recién enviada.

Tipo: objeto SynthesisTask

Errores

EngineNotSupportedException

Este motor no es compatible con la voz que ha designado. Elija una voz nueva que sea compatible con el motor o cámbielo y reinicie la operación.

Código de estado HTTP: 400

InvalidS3BucketException

El nombre de bucket de Amazon S3 proporcionado no es válido. Compruebe lo que ha introducido con los requisitos de denominación del bucket de S3 e inténtelo de nuevo.

Código de estado HTTP: 400

InvalidS3KeyException

El prefijo de clave de Amazon S3 no es válido. Proporcione un nombre de clave de objeto de S3 válido.

Código de estado HTTP: 400

InvalidSampleRateException

La frecuencia de muestreo especificada no es válida.

Código de estado HTTP: 400

InvalidSnsTopicArnException

El ARN del tema de SNS proporcionado no es válido. Introduzca un ARN de tema de SNS válido e inténtelo de nuevo.

Código de estado HTTP: 400

InvalidSsmlException

El SSML que ha proporcionado no es válido. Compruebe la sintaxis de SSML y la ortografía de las etiquetas y los valores y, a continuación, vuelva a intentarlo.

Código de estado HTTP: 400

LanguageNotSupportedException

Actualmente, el idioma especificado en esta capacidad no es compatible con Amazon Polly.

Código de estado HTTP: 400

LexiconNotFoundException

Amazon Polly no encuentra el lexicón especificado. Esto puede deberse a la falta de un lexicón, a que su nombre esté mal escrito o a que se especifique un lexicón que se encuentra en una región diferente.

Compruebe que el lexicón existe, se encuentra en la región (consulte ListLexicons) y que ha escrito su nombre correctamente. Inténtelo de nuevo.

Código de estado HTTP: 404

MarksNotSupportedForFormatException

Las marcas de voz no son compatibles con el `OutputFormat` seleccionado. Las marcas de voz solo están disponibles para el contenido en formato `json`.

Código de estado HTTP: 400

ServiceFailureException

Una condición desconocida ha provocado un fallo en el servicio.

Código de estado HTTP: 500

SsmlMarksNotSupportedForTextTypeException

Las marcas de voz SSML no son compatibles con la entrada de texto simple.

Código de estado HTTP: 400

TextLengthExceededException

El valor del parámetro "Text" supera los límites aceptados. Para la API `SynthesizeSpeech`, el límite de entrada de texto es de un máximo de 6000 caracteres en total, de los cuales no se pueden facturar más de 3000 caracteres. Para la API `StartSpeechSynthesisTask`, el máximo es de 200 000 caracteres, de los cuales no se pueden facturar más de 100 000 caracteres. Las etiquetas SSML no cuentan como caracteres contabilizados.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulte lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

SynthesizeSpeech

Sintetiza la entrada UTF-8, texto sin formato o SSML, en una secuencia de bytes. La entrada SSML debe ser un SSML válido y bien formado. Es posible que algunos alfabetos no estén disponibles con todas las voces (por ejemplo, es posible que las voces en inglés no lean en absoluto el cirílico) a menos que se utilice el mapeo de fonemas. Para obtener más información, consulte Cómo funciona.

Sintaxis de la solicitud

```
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"
}
```

Parámetros de solicitud del URI

La solicitud no utiliza ningún parámetro de URI.

Cuerpo de la solicitud

La solicitud acepta los siguientes datos en formato JSON.

Engine

Especifica el motor (`standard`, `neural`, `long-form` o `generative`) que utilizará Amazon Polly al procesar el texto de entrada para la síntesis de voz. Proporcione un motor compatible con la voz que seleccione. Si no proporciona un motor, se selecciona el motor estándar de forma predeterminada. Si una voz elegida no es compatible con el motor estándar, se producirá un error. Para obtener información sobre las voces de Amazon Polly y qué voces están disponibles para cada motor, consulte Voces disponibles.

Tipo: cadena

Valores válidos: `standard | neural | long-form | generative`

Obligatorio: no

LanguageCode

Código de idioma opcional para la solicitud de sintetizar voz. Esto solo es necesario si se utiliza una voz bilingüe, como Aditi, que se puede utilizar tanto en inglés indio (en-IN) como en hindi (hi-IN).

Si se utiliza una voz bilingüe y no se especifica ningún código de idioma, Amazon Polly utilizará el idioma predeterminado de la voz bilingüe. El idioma predeterminado de cualquier voz es el que devuelve la DescribeVoicesoperación para el `LanguageCode` parámetro. Por ejemplo, si no se especifica ningún código de idioma, Aditi utilizará el inglés indio en lugar del hindi.

Tipo: cadena

Valores válidos: `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`

Obligatorio: no

LexiconNames

Enumere uno o más nombres de lexicones de pronunciación que desee que el servicio aplique durante la síntesis. Los lexicones solamente se aplican al texto si el idioma especificado en ellos es el mismo que el idioma elegido. Para obtener información sobre el almacenamiento de léxicos, consulte PutLexicon.

Tipo: matriz de cadenas

Miembros de la matriz: número máximo de 5 elementos.

Patrón: `[0-9A-Za-z]{1,20}`

Obligatorio: no

OutputFormat

Formato en el que se codificará la salida devuelta. Para la transmisión de audio, será mp3, ogg_vorbis o pcm. Para las marcas de voz, será json.

Cuando se usa pcm, el contenido devuelto es audio/pcm en un formato firmado de 16 bits, 1 canal (mono), little-endian.

Tipo: cadena

Valores válidos: `json | mp3 | ogg_vorbis | pcm`

Obligatorio: sí

SampleRate

La frecuencia de audio especificada en Hz.

Los valores válidos para mp3 y ogg_vorbis son “8000”, “16 000”, “22 050” y “24 000”. El valor predeterminado para las voces estándar es “22 050". El valor predeterminado para las voces neuronales es "24 000". El valor predeterminado para las voces de formato largo es “24 000”. El valor predeterminado para las voces generativas es "24 000".

Los valores válidos para el pcm son "8000" y "16 000". El valor por defecto es "16 000".

Tipo: cadena

Requerido: no

SpeechMarkTypes

El tipo de marcas de voz devueltas para el texto de entrada.

Tipo: matriz de cadenas

Miembros de la matriz: número máximo de 4 elementos.

Valores válidos: `sentence | ssml | viseme | word`

Obligatorio: no

Text

Introduzca el texto para sintetizarlo. Si especifica `ssml` como el `TextType`, siga el formato SSML para el texto de entrada.

Tipo: cadena

Obligatorio: sí

TextType

Especifica si el texto de entrada es texto sin formato o SSML. El valor predeterminado es texto sin formato. Para obtener más información, consulte Uso de SSML.

Tipo: cadena

Valores válidos: `ssml | text`

Obligatorio: no

VoiceId

ID de voz que se usará para la síntesis. Puede obtener una lista de las voces disponibles IDs llamando a la DescribeVoicesoperación.

Tipo: cadena

Valores válidos: `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`

Obligatorio: sí

Sintaxis de la respuesta

```
HTTP/1.1 200
Content-Type: ContentType
```

```
x-amzn-RequestCharacters: RequestCharacters

AudioStream
```

Elementos de respuesta

Si la acción se realiza correctamente, el servicio devuelve una respuesta HTTP 200.

La respuesta devuelve los siguientes encabezados HTTP.

ContentType

Especifica el tipo de transmisión de audio. Esto debería reflejar el `OutputFormat` del parámetro de su solicitud.

- Si solicita `mp3` como el `OutputFormat`, el `ContentType` resultante es audio/mpeg.
- Si solicita `ogg_vorbis` como el `OutputFormat`, el `ContentType` resultante es audio/ogg.
- Si solicita `pcm` como el `OutputFormat`, el `ContentType` resultante es audio/pcm en un formato little-endian firmado de 16 bits y 1 canal (mono).
- Si lo solicita `json` como`OutputFormat`, el resultado `ContentType` es application/x-json-stream.

RequestCharacters

Número de caracteres sintetizados.

La respuesta devuelve lo siguiente como el cuerpo HTTP.

AudioStream

Secuencia que contiene la voz sintetizada.

Errores

EngineNotSupportedException

Este motor no es compatible con la voz que ha designado. Elija una voz nueva que sea compatible con el motor o cámbielo y reinicie la operación.

Código de estado HTTP: 400

InvalidSampleRateException

La frecuencia de muestreo especificada no es válida.

Código de estado HTTP: 400

InvalidSsmlException

El SSML que ha proporcionado no es válido. Compruebe la sintaxis de SSML y la ortografía de las etiquetas y los valores y, a continuación, vuelva a intentarlo.

Código de estado HTTP: 400

LanguageNotSupportedException

Actualmente, el idioma especificado en esta capacidad no es compatible con Amazon Polly.

Código de estado HTTP: 400

LexiconNotFoundException

Amazon Polly no encuentra el lexicón especificado. Esto puede deberse a la falta de un lexicón, a que su nombre esté mal escrito o a que se especifique un lexicón que se encuentra en una región diferente.

Compruebe que el lexicón existe, se encuentra en la región (consulte ListLexicons) y que ha escrito su nombre correctamente. Inténtelo de nuevo.

Código de estado HTTP: 404

MarksNotSupportedForFormatException

Las marcas de voz no son compatibles con el `OutputFormat` seleccionado. Las marcas de voz solo están disponibles para el contenido en formato `json`.

Código de estado HTTP: 400

ServiceFailureException

Una condición desconocida ha provocado un fallo en el servicio.

Código de estado HTTP: 500

SsmlMarksNotSupportedForTextTypeException

Las marcas de voz SSML no son compatibles con la entrada de texto simple.

Código de estado HTTP: 400

TextLengthExceededException

El valor del parámetro "Text" supera los límites aceptados. Para la API `SynthesizeSpeech`, el límite de entrada de texto es de un máximo de 6000 caracteres en total, de los cuales no se pueden facturar más de 3000 caracteres. Para la API `StartSpeechSynthesisTask`, el máximo es de 200 000 caracteres, de los cuales no se pueden facturar más de 100 000 caracteres. Las etiquetas SSML no cuentan como caracteres contabilizados.

Código de estado HTTP: 400

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- Interfaz de la línea de comandos de AWS
- AWS SDK para .NET
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

Data Types

Los siguientes tipos de datos son compatibles:

- Lexicon
- LexiconAttributes
- LexiconDescription
- SynthesisTask

- Voice

Lexicon

Proporciona el nombre y el contenido del lexicón en formato de cadena. Para obtener más información, consulte Pronunciation Lexicon Specification (PLS) versión 1.0.

Contenido

Content

Contenido de lexicón en formato de cadena. El contenido de un lexicón debe estar en formato PLS.

Tipo: cadena

Requerido: no

Name

El nombre del lexicón.

Tipo: cadena

Patrón: `[0-9A-Za-z]{1,20}`

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

LexiconAttributes

Contiene metadatos que describen el lexicón, como el número de lexemas, el código de idioma, etc. Para obtener más información, consulte Administración de lexicones.

Contenido

Alphabet

Alfabeto fonético utilizado en el lexicón. Los valores válidos son `ipa` y `x-sampa`.

Tipo: cadena

Requerido: no

LanguageCode

Código de idioma al que se aplica el lexicón. Un lexicón con un código de idioma como “en” se aplicaría a todos los idiomas ingleses (en-GB, en-US, en-AUS, en-WLS, etc.).

Tipo: cadena

Valores válidos: `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`

Obligatorio: no

LastModified

Fecha en que se modificó el lexicón por última vez (un valor de marca de tiempo).

Tipo: marca temporal

Obligatorio: no

LexemesCount

Número de lexemas en el lexicón.

Tipo: entero

Obligatorio: no

LexiconArn

El nombre de recurso de Amazon (ARN) del lexicón.

Tipo: cadena

Requerido: no

Size

Tamaño total del lexicón, en caracteres.

Tipo: entero

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

LexiconDescription

Describe el contenido del lexicón.

Contenido

Attributes

Proporciona metadatos del lexicón.

Tipo: objeto LexiconAttributes

Obligatorio: no

Name

El nombre del lexicón.

Tipo: cadena

Patrón: `[0-9A-Za-z]{1,20}`

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

SynthesisTask

SynthesisTask objeto que proporciona información sobre una tarea de síntesis de voz.

Contenido

CreationTime

Marca de tiempo del momento en que se inició la tarea de síntesis.

Tipo: marca temporal

Obligatorio: no

Engine

Especifica el motor (`standard`, `neural`, `long-form` o `generative`) que utilizará Amazon Polly al procesar el texto de entrada para la síntesis de voz. El uso de una voz que no sea compatible con el motor seleccionado provocará un error.

Tipo: cadena

Valores válidos: `standard | neural | long-form | generative`

Obligatorio: no

LanguageCode

Código de idioma opcional para una tarea de síntesis. Esto solo es necesario si se utiliza una voz bilingüe, como Aditi, que se puede utilizar tanto en inglés indio (en-IN) como en hindi (hi-IN).

Si se utiliza una voz bilingüe y no se especifica ningún código de idioma, Amazon Polly utilizará el idioma predeterminado de la voz bilingüe. El idioma predeterminado de cualquier voz es el que devuelve la DescribeVoicesoperación para el `LanguageCode` parámetro. Por ejemplo, si no se especifica ningún código de idioma, Aditi utilizará el inglés indio en lugar del hindi.

Tipo: cadena

Valores válidos: `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`

Obligatorio: no

LexiconNames

Enumere uno o más nombres de lexicones de pronunciación que desee que el servicio aplique durante la síntesis. Los lexicones solamente se aplican al texto si el idioma especificado en ellos es el mismo que el idioma elegido.

Tipo: matriz de cadenas

Miembros de la matriz: número máximo de 5 elementos.

Patrón: `[0-9A-Za-z]{1,20}`

Obligatorio: no

OutputFormat

Formato en el que se codificará la salida devuelta. Para la transmisión de audio, será mp3, ogg_vorbis o pcm. Para las marcas de voz, será json.

Tipo: cadena

Valores válidos: `json | mp3 | ogg_vorbis | pcm`

Obligatorio: no

OutputUri

Ruta para el archivo de voz de salida.

Tipo: cadena

Requerido: no

RequestCharacters

Número de caracteres facturables sintetizados.

Tipo: entero

Obligatorio: no

SampleRate

La frecuencia de audio especificada en Hz.

Los valores válidos para mp3 y ogg_vorbis son “8000”, “16 000”, “22 050” y “24 000”. El valor predeterminado para las voces estándar es “22 050". El valor predeterminado para las voces neuronales es "24 000". El valor predeterminado para las voces de formato largo es “24 000”. El valor predeterminado para las voces generativas es "24 000".

Los valores válidos para el pcm son "8000" y "16 000". El valor por defecto es "16 000".

Tipo: cadena

Requerido: no

SnsTopicArn

El ARN para el tema de SNS se usa opcionalmente para proporcionar una notificación de estado para una tarea de síntesis de voz.

Tipo: cadena

Patrón: `^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)$`

Obligatorio: no

SpeechMarkTypes

El tipo de marcas de voz devueltas para el texto de entrada.

Tipo: matriz de cadenas

Miembros de la matriz: número máximo de 4 elementos.

Valores válidos: `sentence | ssml | viseme | word`

Obligatorio: no

TaskId

El identificador generado por Amazon Polly para una tarea de síntesis de voz.

Tipo: cadena

Patrón: `^[a-zA-Z0-9_-]{1,100}$`

Obligatorio: no

TaskStatus

Estado actual de la tarea de síntesis de voz individual.

Tipo: cadena

Valores válidos: `scheduled | inProgress | completed | failed`

Obligatorio: no

TaskStatusReason

Motivo del estado actual de una tarea específica de síntesis de voz, incluidos los errores si la tarea ha fallado.

Tipo: cadena

Requerido: no

TextType

Especifica si el texto de entrada es texto sin formato o SSML. El valor predeterminado es texto sin formato.

Tipo: cadena

Valores válidos: `ssml | text`

Obligatorio: no

VoiceId

ID de voz que se usará para la síntesis.

Tipo: cadena

Valores válidos: `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
```

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

Voice

Descripción de la voz.

Contenido

AdditionalLanguageCodes

Códigos de idiomas adicionales disponibles para la voz especificada, además de su idioma predeterminado.

Por ejemplo, el idioma predeterminado de Aditi es el inglés indio (en-IN) porque se utilizó por primera vez para ese idioma. Como Aditi es bilingüe y habla con fluidez tanto el inglés indio como el hindi, este parámetro mostraría el código `hi-IN`.

Tipo: matriz de cadenas

Valores válidos: `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`

Obligatorio: no

Gender

Género de la voz.

Tipo: cadena

Valores válidos: `Female | Male`

Obligatorio: no

Id

ID de voz asignado por Amazon Polly. Este es el identificador que especifica al llamar a la operación `SynthesizeSpeech`.

Tipo: cadena

Valores válidos: `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

Obligatorio: no

LanguageCode

Código de idioma de la voz.

Tipo: cadena

Valores válidos: 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

Obligatorio: no

LanguageName

Nombre legible por humanos del idioma en inglés.

Tipo: cadena

Requerido: no

Name

Nombre de la voz (por ejemplo, Salli, Kendra, etc.). Esto proporciona un nombre de voz legible para las personas que podría mostrar en su aplicación.

Tipo: cadena

Requerido: no

SupportedEngines

Especifica qué motores (`standard`, `neural`, `long-form` o `generative`) son compatibles con una voz determinada.

Tipo: matriz de cadenas

Valores válidos: `standard | neural | long-form | generative`

Obligatorio: no

Véase también

Para obtener más información sobre el uso de esta API en uno de los idiomas específicos AWS SDKs, consulta lo siguiente:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

Errores comunes

En esta sección se enumeran los errores comunes a las acciones de la API de todos los AWS servicios. En el caso de los errores específicos de una acción de la API de este servicio, consulte el tema de dicha acción de la API.

AccessDeniedException

No tiene acceso suficiente para realizar esta acción.

Código de estado HTTP: 400

IncompleteSignature

La firma de la solicitud no se ajusta a AWS los estándares.

Código de estado HTTP: 400

InternalFailure

El procesamiento de la solicitud ha devuelto un error debido a un error o una excepción desconocidos.

Código de estado HTTP: 500

InvalidAction

La acción u operación solicitada no es válida. Compruebe que la acción se ha escrito correctamente.

Código de estado HTTP: 400

InvalidClientTokenId

El identificador de clave de AWS acceso o certificado X.509 proporcionado no existe en nuestros registros.

Código de estado HTTP: 403

NotAuthorized

No tiene permiso para realizar esta acción.

Código de estado HTTP: 400

OptInRequired

El identificador de clave de AWS acceso necesita una suscripción al servicio.

Código de estado HTTP: 403

RequestExpired

La solicitud llegó al servicio más de 15 minutos después del sello de fecha de la solicitud o más de 15 minutos después de la fecha de caducidad de la solicitud (por ejemplo, en el caso de los prefirmados URLs), o el sello de fecha de la solicitud es más de 15 minutos en el futuro.

Código de estado HTTP: 400

ServiceUnavailable

La solicitud no se ha ejecutado correctamente debido a un error temporal del servidor.

Código de estado HTTP: 503

ThrottlingException

La solicitud fue denegada debido a una limitación de la solicitud.

Código de estado HTTP: 400

ValidationError

La entrada no cumple las restricciones especificadas por un AWS servicio.

Código de estado HTTP: 400

Parámetros comunes

La siguiente lista contiene los parámetros que utilizan todas las acciones para firmar solicitudes de Signature Version 4 con una cadena de consulta. Los parámetros específicos de acción se enumeran en el tema correspondiente a la acción. Para obtener más información sobre la versión 4 de Signature, consulte Firmar solicitudes de AWS API en la Guía del usuario de IAM.

Action

Las acciones que se van a realizar.

Tipo: cadena

Obligatorio: sí

Version

La versión de API para la que está escrita la solicitud, expresada en el formato YYYY-MM-DD.

Tipo: cadena

Obligatorio: sí

X-Amz-Algorithm

El algoritmo de hash que utilizó para crear la solicitud de firma.

Condición: especifique este parámetro cuando incluya información de autenticación en una cadena de consulta en lugar de en el encabezado de autorización HTTP.

Tipo: cadena

Valores válidos: `AWS4-HMAC-SHA256`

Obligatorio: condicional

X-Amz-Credential

El valor del ámbito de la credencial, que es una cadena que incluye la clave de acceso, la fecha, la región a la que se dirige, el servicio que solicita y una cadena de terminación (“aws4_request”). El valor se expresa en el siguiente formato: access_key/AAAAMMDD/region/service/aws4_request.

Para obtener más información, consulte Crear una solicitud de AWS API firmada en la Guía del usuario de IAM.

Condición: especifique este parámetro cuando incluya información de autenticación en una cadena de consulta en lugar de en el encabezado de autorización HTTP.

Tipo: cadena

Obligatorio: condicional

X-Amz-Date

La fecha utilizada para crear la firma. El formato debe ser ISO 8601 formato básico (AAAAMMDD‘T’HHMMSS‘Z’). Por ejemplo, la siguiente fecha y hora es un X-Amz-Date valor válido:`20120325T120000Z`.

Condición: X-Amz-Date es opcional en todas las solicitudes; puede utilizarse para anular la fecha empleada para firmar solicitudes. Si el encabezado de fecha se especifica en el formato básico ISO 8601, no X-Amz-Date es obligatorio. Cuando X-Amz-Date se usa, siempre anula el valor del encabezado de fecha. Para obtener más información, consulte Elementos de la firma de una solicitud de AWS API en la Guía del usuario de IAM.

Tipo: cadena

Obligatorio: condicional

X-Amz-Security-Token

El token de seguridad temporal que se obtuvo mediante una llamada a AWS Security Token Service (AWS STS). Para obtener una lista de servicios compatibles con las credenciales de seguridad temporales de AWS STS, consulte Servicios de AWS que funcionan con IAM en la Guía del usuario de IAM.

Condición: si utilizas credenciales de seguridad temporales de AWS STS, debes incluir el token de seguridad.

Tipo: cadena

Obligatorio: condicional

X-Amz-Signature

Especifica la firma codificada hexadecimal que se calculó a partir de la cadena que se va a firmar y la clave de firma derivada.

Condición: especifique este parámetro cuando incluya información de autenticación en una cadena de consulta en lugar de en el encabezado de autorización HTTP.

Tipo: cadena

Obligatorio: condicional

X-Amz-SignedHeaders

Especifica todos los encabezados HTTP que se incluyeron como parte de la solicitud canónica. Para obtener más información sobre cómo especificar encabezados firmados, consulte Crear una solicitud de AWS API firmada en la Guía del usuario de IAM.

Condición: especifique este parámetro cuando incluya información de autenticación en una cadena de consulta en lugar de en el encabezado de autorización HTTP.

Tipo: cadena

Obligatorio: condicional

Historial de documento de Amazon Polly

En la siguiente tabla se describen los cambios importantes en cada versión de la Guía para desarrolladores de Amazon Polly. Para recibir notificaciones sobre los cambios en esta documentación, puede suscribirse a una fuente RSS.

- Última actualización de la documentación: 18 de febrero de 2025

Cambio	Descripción	Fecha
Se ha añadido una nueva región para voces neuronales y estándares	Amazon Polly ya está disponible en la región Asia Pacífico (Malasia). AWS Esta región admite el TTS neuronal (NTTS) y las voces estándar. Para obtener más información, consulte Voces neuronales y Voces estándar.	27 de marzo de 2025
Se agregó una nueva voz para Neural text-to-speech	Amazon Polly ahora ofrece una voz coreana adicional : Jihye. Consulte Voces neuronales para ver una lista de las voces NTTS.	26 de marzo de 2025
Se agregó una nueva región para voces neuronales y estándar	Amazon Polly ya está disponible en la región Europa (España). AWS Esta región admite el TTS neuronal (NTTS) y las voces estándar. Para obtener más información, consulte Voces neuronales y Voces estándar.	18 de febrero de 2025
Se ha añadido una nueva voz para las voces neuronales text-to-speech	Amazon Polly ahora ofrece una voz adicional en inglés (singapurense): Jasmine.	11 de febrero de 2025

	Consulte Voces neuronales para ver una lista de las voces NTTS.	
Se agregaron nuevas voces para generar text-to-speech	Amazon Polly ahora ofrece siete nuevas voces generativ as: Pedro, Andrés, Sergio, Daniel, Kajal, Rémi y Bianca. Consulte Voces generativas para ver una lista de las voces de TTS generativas.	21 de noviembre de 2024
Se han añadido nuevas voces de formato largo	Amazon Polly ahora ofrece más voces de formato largo. Se agregaron una voz adicional de formato largo en inglés (EE. UU.) y dos nuevas en español (España): Patrick, Alba y Raúl. Consulte Voces de formato largo para ver una lista de todas las voces de formato largo.	14 de noviembre de 2024
Se agregaron nuevas voces para generativo text-to-speech	Amazon Polly ahora ofrece seis nuevas voces generativ as: Ayanda, Léa, Lucia, Mía, Lupe y Vicki. Consulte Voces generativas para ver una lista de las voces de TTS generativ as.	6 de noviembre de 2024

Se han añadido nuevas voces para generativo text-to-speech	Amazon Polly ahora ofrece cuatro nuevas voces generativas: Olivia, Danielle, Joanna y Stephen. Consulte Voces generativas para ver una lista de las voces de TTS generativas.	10 de octubre de 2024
Se agregaron nuevos idiomas para text-to-speech	Amazon Polly ahora ofrece dos nuevos idiomas de TTS: checo (CS-CZ) y alemán (suizo) (de-CH). Para ver una lista de los lenguajes admitidos, consulte Idiomas admitidos.	26 de septiembre de 2024
Se han añadido nuevas voces para NTTS	Amazon Polly ahora ofrece dos nuevas voces NTTS: Jitka y Sabrina. Consulte Voces neuronales para ver una lista de las voces NTTS.	27 de agosto de 2024
Se ha agregado un nuevo motor de voz generativo	Amazon Polly ahora ofrece un motor de voz generativo diseñado para contenido más extenso, con tres voces en inglés en una variante generativa: Amy, Matthew y Ruth. Para obtener más información, consulte Voces generativas.	28 de marzo de 2024
Se agregó una nueva voz para NTTS	Amazon Polly ahora ofrece la voz turca NTTS de Burcu. Consulte Voces neuronales para ver una lista de las voces NTTS.	14 de febrero de 2024

Se ha agregado un nuevo motor de voz de formato largo	Amazon Polly ahora ofrece un motor de voz de formato largo diseñado para contenido más extenso, con tres voces en inglés estadounidense: Danielle, Gregory y Ruth. Consulte Voces de formato largo para obtener más información.	16 de noviembre de 2023
Se han añadido nuevas voces para NTTS	Amazon Polly ahora ofrece dos nuevas voces de NTTS en inglés estadounidense: Danielle y Gregory. Consulte Voces neuronales para ver una lista de las voces NTTS.	5 de octubre de 2023
Amazon Polly para Windows	El complemento de Amazon Polly para Windows interfaz de programación de aplicaciones de voz (SAPI) dejará de ser compatible.	26 de septiembre de 2023
Se ha actualizado la guía de cuotas para Amazon Polly	Guía actualizada de cuotas de Amazon Polly. Se han añadido ejemplos y aclaraciones de términos. Consulte Cuotas en Amazon Polly para ver las actualizaciones.	17 de agosto de 2023
Se agregó una nueva voz para NTTS	Amazon Polly ahora proporcio na la voz NTTS en árabe del Golfo Zayd. Consulte Voces neuronales para ver una lista de las voces NTTS.	16 de agosto de 2023

Se agregó una nueva voz para NTTS	Amazon Polly ahora ofrece la voz NTTS en francés belga Isabelle. Consulte Voces neuronales para ver una lista de las voces NTTS.	1 de agosto de 2023
Se agregó una nueva voz para NTTS	Amazon Polly ahora ofrece la voz NTTS para neerlandés belga (flamenco) Lisa. Consulte Voces neuronales para ver una lista de las voces NTTS.	7 de junio de 2023
Se han añadido nuevas voces para NTTS	Amazon Polly ahora ofrece dos nuevas voces en NTTS: inglés de Irlanda (Niamh) y danés (Sofie). Consulte Voces neuronales para ver una lista de las voces NTTS.	30 de mayo de 2023
Se ha actualizado la guía de IAM para Amazon Polly	Se ha actualizado la guía para implementar las prácticas recomendadas de IAM. Para obtener más información, consulta prácticas recomendadas de seguridad en IAM.	19 de abril de 2023
WordPress update	El WordPress complemento Amazon Polly dejará de ser compatible.	6 de abril de 2023

Se agregó una nueva región	Amazon Polly ya está disponible en la región AWS Asia-Pacífico (Osaka). Esta región admite el TTS neuronal (NTTS). Para obtener más información, consulte Compatibilidad con características y regiones para ver una lista de las regiones que admiten el NTTS.	5 de abril de 2023
Se han añadido nuevas voces para NTTS	Amazon Polly ahora ofrece dos nuevas voces NTTS japonesas: Kazuha y Tomoko. Consulte Voces neuronales para ver una lista de las voces NTTS.	7 de febrero de 2023
Se han añadido nuevas voces para NTTS	Amazon Polly ahora ofrece dos nuevas voces NTTS en inglés estadounidense: Stephen y Ruth. Consulte Voces neuronales para ver una lista de las voces NTTS.	31 de enero de 2023
Se han añadido nuevas voces para NTTS	Amazon Polly ahora ofrece nuevas voces NTTS para: portugués brasileño (Thiago), español castellano (Sergio), francés (Rémi), italiano (Adriano) y español mexicano (Andrés). Consulte Voces neuronales para ver una lista de las voces NTTS.	24 de enero de 2023

Se han añadido nuevas voces para NTTS	Amazon Polly ahora ofrece voces NTTS para árabe (Hala) y polaco (Ola). Consulte Voces neuronales para ver una lista de las voces NTTS.	17 de noviembre de 2022
Soporte de versiones AWS PrivateLink	Amazon Polly ahora ofrece AWS PrivateLink soporte. Consulte Uso de Amazon Polly con puntos de conexión de VPC para obtener más información.	9 de noviembre de 2022
Se agregaron nuevas voces e idiomas para NTTS	Amazon Polly ahora ofrece voces NTTS para finés (Suvi), noruego (Ida) y sueco (Elin). Consulte Voces neuronales para ver una lista de las voces NTTS.	8 de noviembre de 2022
Se agregó una nueva voz para NTTS	Amazon Polly ahora ofrece la voz holandesa NTTS de Laura. Consulte Voces neuronales para ver una lista de las voces NTTS.	2 de noviembre de 2022
Se agregó una nueva región	Amazon Polly ya está disponible en la región AWS Europa (París). Esta región admite el TTS neuronal (NTTS). Para obtener más información, consulte Compatibilidad con características y regiones para ver una lista de las regiones que admiten el NTTS.	22 de septiembre de 2022

Se han añadido nuevas voces e idiomas para NTTS	Amazon Polly ahora ofrece la voz NTTS cantonesa Hiujin. Consulte Voces neuronales para ver una lista de las voces NTTS.	20 de septiembre de 2022
Se agregó una nueva región	A partir de ahora, Amazon Polly está disponible en la región AWS Asia-Pacífico (Bombay). Esta región admite el TTS neuronal (NTTS). Para obtener más información, consulte Compatibilidad con características y regiones para ver una lista de las regiones que admiten el NTTS.	1 de septiembre de 2022
Se agregó una nueva voz para NTTS	Amazon Polly ahora ofrece la voz en mandarín Zhiyu como voz NTTS. Consulte Voces neuronales para ver una lista de las voces NTTS.	23 de agosto de 2022
Se agregó una nueva voz para NTTS	Amazon Polly ahora ofrece la voz en hindi NTTS Kajal. Consulte Voces neuronales para ver una lista de las voces NTTS.	27 de julio de 2022
Se han añadido nuevas voces para NTTS	Amazon Polly ahora ofrece voces NTTS para español estadounidense (Pedro), alemán (Daniel), francés canadiense (Liam) e inglés británico (Arthur). Consulte Voces neuronales para ver una lista de las voces NTTS.	28 de junio de 2022

Se agregó una nueva voz para NTTS	Amazon Polly ahora ofrece la voz portuguesa (brasileña) Vitória como voz NTTS. Consulte Voces neuronales para ver una lista de las voces NTTS.	27 de abril de 2022
Se agregó una nueva voz para NTTS	Amazon Polly ahora ofrece la voz portuguesa (europea) Inês como voz NTTS. Consulte Voces neuronales para ver una lista de las voces NTTS.	26 de abril de 2022
Se han añadido nuevas voces e idiomas para NTTS	Amazon Polly ahora ofrece el idioma alemán (austriaco) y la voz NTTS de Hannah. Consulte Voces neuronales para ver una lista de las voces NTTS.	19 de abril de 2022
Se agregaron nuevas voces e idiomas para NTTS	Amazon Polly ahora ofrece la voz española (mexicana) Mia como voz NTTS. Se agregó un nuevo idioma, el catalán, junto con la voz Arlet en NTTS. Consulte Voces neuronales para ver una lista de las voces NTTS.	22 de marzo de 2022
Se agregó una nueva voz para NTTS	Amazon Polly ahora ofrece la voz japonesa Takumi como voz NTTS. Consulte Voces neuronales para ver una lista de las voces NTTS.	6 de diciembre de 2021

Se agregó una nueva voz para NTTS	Amazon Polly ahora ofrece la voz francesa Léa como voz NTTS. Consulte Voces neuronales para ver una lista de las voces NTTS.	18 de noviembre de 2021
Se han añadido nuevas voces para NTTS	Amazon Polly ahora ofrece la voz italiana Bianca y la voz europea española Lucía como voces NTTS. Consulte Voces neuronales para ver una lista de las voces NTTS.	8 de noviembre de 2021
Se agregó una nueva voz para NTTS	Amazon Polly ahora ofrece una nueva voz en inglés sudafricano, Ayanda. La voz solo está disponible como voz NTTS. Consulte Voces neuronales para ver una lista de las voces NTTS.	1 de septiembre de 2021
Se agregó una nueva región	Amazon Polly ahora está disponible en la región AWS de África (Ciudad del Cabo). Esta región admite el TTS neuronal (NTTS). Para obtener más información, consulte Compatibilidad con características y regiones para ver una lista de las regiones que admiten el NTTS.	1 de septiembre de 2021

Se agregaron un nuevo idioma y voz	Amazon Polly ahora es compatible con el inglés neozelandés (en-NZ). Una nueva voz de NTTS, Aria, habla inglés neozelandés y una selección de palabras maoríes.	24 de agosto de 2021
Nueva característica	Amazon Polly convierte el estilo de conversación en la versión predeterminada para las voces neuronales de Matthew y Joanna. Eliminamos las referencias al estilo de habla conversacional.	28 de junio de 2021
Se agregó una nueva voz para NTTS	Amazon Polly ahora ofrece la voz alemana Vicki como voz NTTS.	15 de junio de 2021
Se ha añadido una voz nueva	Se ha añadido una nueva voz femenina, Gabrielle, a la configuración regional de francés (canadiense) (fr-CA). Esta nueva voz es de muy alta calidad y solo está disponible como voz NTTS. Como todas las voces neuronales, solo está disponible en determinadas regiones. Para obtener una lista de las regiones, consulte Compatibilidad con características y regiones.	1 de junio de 2021
Se agregó una nueva voz para NTTS	Amazon Polly ahora ofrece la voz coreana Seoyeon como voz NTTS.	11 de mayo de 2021

Se agregó una nueva región para NTTS	Amazon Polly ahora admite el TTS neuronal (NTTS) en la región de Canadá (Central). AWS Para obtener más información, consulte Compatibilidad con características y regiones de NTTS.	17 de marzo de 2021
Nueva voz disponible para el estilo Newscaster	Además de las voces de Matthew y Joanna para el estilo de habla Newscaster, Amazon Polly ahora proporciona una opción adicional para este estilo de habla. Mediante el motor neuronal, puede utilizar la voz de Amy en inglés británico para el estilo Newscaster. Para obtener más información, consulte Estilos de habla NTTS.	10 de noviembre de 2020
Se agregaron nuevas regiones para NTTS	Además de las regiones actuales de NTTS (us-east-1, us-west-2, eu-west-1 y ap-southeast-2), las voces neuronales ahora son compatibles con en cuatro regiones adicionales: (ap-northeast-1 (Tokio), ap-southeast-1 (Singapur), eu-central-1 (Fráncfort) y eu-west-2 (Londres). Para obtener más información, consulte Compatibilidad con características y regiones de NTTS.	3 de septiembre de 2020

Se ha añadido una voz nueva	Además de las voces infantiles Ivy y Justin, se ha añadido una nueva voz masculina infantil, Kevin, al inglés americano (en-US). Esta nueva voz es de muy alta calidad y solo está disponible como voz NTTS. Como todas las voces neuronales, solo se admite en cuatro regiones: us-east-1 (Norte de Virginia), us-west-2 (Oregón), eu-west-1 (Irlanda) y ap-southeast-2 (Sídney). Para obtener más información, consulte Voces NTTS.	16 de junio de 2020
Nueva voz disponible para el estilo Newscaster	Además de las voces de Matthew y Joanna para el estilo de habla Newscaster, Amazon Polly ahora proporcio na una opción adicional para este estilo de habla. Mediante el motor neuronal, puede utilizar la voz de Lupe en español estadounidense para el estilo Newscaster. Para obtener más información, consulte Estilos de habla NTTS.	16 de abril de 2020

Nueva característica	Además del estilo de habla Newscaster, Amazon Polly ahora proporciona un segundo estilo de habla NTTS para ayudarle a sintetizar aún mejor los pasajes de texto a voz. El estilo Conversacional utiliza el sistema neuronal para generar habla con un estilo conversacional más cordial y expresivo que se puede usar en muchos casos de uso. Para obtener más información, consulte Estilos de habla NTTS.	25 de noviembre de 2019
Se han añadido voces nuevas	Se han añadido dos voces nuevas: Camila (mujer, portugués-Brasil) y Lupe (mujer, español-EE. UU.).	23 de octubre de 2019
Nueva característica añadida	Se ha añadido el complemento de Amazon Polly para Windows para incorporar la gama completa de voces de Amazon Polly en aplicaciones compatibles con SAPI de Windows.	26 de septiembre de 2019

Nueva característica importante	Además de las voces estándar text-to-speech (TTS) compatibles con Amazon Polly desde su lanzamiento, Amazon Polly ahora ofrece un sistema Neural TTS (NTTS) mejorado que puede proporcionar voces de una calidad aún mayor y, por lo tanto, las voces más naturales y parecidas a las humanas posibles. text-to-speech Para obtener más información, consulte Neural. Text-to-Speech	30 de julio de 2019
Se han añadido voces nuevas	Nuevas voces añadidas: Lucía (mujer, española) y Bianca (mujer, italiana).	2 de agosto de 2018
Se ha añadido un idioma nuevo	Nuevo idioma añadido: español (México) (es-MX). Este idioma utiliza la voz femenina de Mia.	2 de agosto de 2018
Se ha añadido un idioma nuevo	Nuevo idioma añadido: Hindi (hi-IN). Esta voz utiliza la voz femenina de Aditi, que también se utiliza para inglés de India, convirtiendo a Aditi de Amazon Polly en la primera voz bilingüe.	2 de agosto de 2018
Nueva característica añadida	Se ha incorporado Síntesis de voz de pasajes de texto largos (hasta 100 000 caracteres contabilizados).	17 de julio de 2018

Nueva característica de SSML añadida	Incorporación de Duración máxima para fragmento de voz sintetizado.	17 de julio de 2018
Se ha añadido una voz nueva	Se ha añadido una voz nueva: Léa (femenina, francés).	5 de junio de 2018
Ampliación de las regiones	Expansión del servicio de Amazon Polly a todas las regiones comerciales.	4 de junio de 2018
Se ha añadido un idioma nuevo	Nuevo idioma añadido: coreano (ko-KR).	4 de junio de 2018
Característica ampliada	La función Amazon Polly WordPress Plugin, que incluye la incorporación de las funciones de Amazon Translate.	4 de junio de 2018
Se han añadido voces nuevas	Se han añadido dos voces nuevas: Aditi (femenina, inglés de India) y Seoyeon (femenina , coreano).	15 de noviembre de 2017
Nueva característica	Incorporación de nueva característica Marcas de voz, además de una ampliación de las capacidades de SSML.	19 de abril de 2017
Nueva guía	Esta es la primera versión de la Guía para desarrolladores de Amazon Polly.	30 de noviembre de 2016

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.