AWS Polly TTS

Polly Text-to-Speech – User Guide

This app integrates Amazon Polly to convert text into natural-sounding speech. Use the CLI to synthesize from plain text, text files, or PDFs, with options for voice, gender, accent, speaking style, audio format, and sample rate.

Overview

The application provides a reusable Python wrapper (`PollyTTS`) and a command-line interface to synthesize speech via **Amazon Polly**. It supports long text chunking, environment-based credentials, WAV/MP3/OGG/PCM output, and optional PDF text extraction.

About Amazon Polly

Amazon Polly is a cloud service that converts text into lifelike speech. It offers many languages and voices, including Neural TTS for improved naturalness, and supports SSML features such as prosody, emphasis, and domain styles (e.g., newscaster).

Features

- **Multiple inputs**: inline text, `--input-file` for .txt or .pdf
- **Voice selection**: explicit `--voice` or auto by `--gender` and `--accent`
- **Speaking styles**: `--style` conversational, newscaster, narration (Neural voices)
- **Audio formats**: mp3, ogg_vorbis, pcm, wav (WAV written with proper headers)
- **Sample rate**: `--sample-rate` (e.g., 16000, 22050)
- **Chunking**: Safely splits long text to fit Polly limits
- **Env credentials**: uses AWS default chain; optional .env loading
- **List voices**: `--list-voices` with gender/engine visibility

Support Matrix (Key)

Requirements

Capability	Notes
Engines	standard, neural (voice dependent)
Formats	mp3, ogg_vorbis, pcm, wav
Voices	Male/Female across many locales
Styles	conversational, newscaster, narration (Neural)
Languages/Accents	e.g., en-US, en-GB, en-AU, en-IN, and more
Input	Text, .txt files, PDF (via extraction)

- Python 3.9+
- AWS credentials with `polly:SynthesizeSpeech` and `polly:DescribeVoices`
- Packages: `boto3`, `python-dotenv` (optional), `PyPDF2` (for PDFs)

Quick Start

```
# 1) Create and activate venv (Windows PowerShell)
cd D:\xampp\htdocs\hm\polly
py -3 -m venv .venv
.\.venv\Scripts\Activate.ps1

# 2) Install requirements
pip install -r requirements.txt

# 3) Verify AWS identity and permissions
aws sts get-caller-identity

# 4) Synthesize basic MP3
python polly.py --text "Hello from Polly" --output out.mp3

# 5) For maximum compatibility, generate WAV
python polly.py --text "Hello" --format wav --sample-rate 16000 --output out.wav
```

Usage Examples

List Voices (by accent)

```
python polly.py --list-voices --accent en-US
```

Pick by Gender and Accent

```
python polly.py --text "Welcome" --gender male --accent en-GB --engine neural --output uk_male.mp3
```

Explicit Voice with Style

```
python polly.py --text "Breaking news now" --voice Matthew --style newscaster --engine neural --output news.mp3
```

From Text File

```
python polly.py --input-file input.txt --voice Joanna --output readout.mp3
```

From PDF (extraction)

```
python polly.py --input-file dts.pdf --format wav --sample-rate 16000 --engine neural --output dts.wav
```

Troubleshooting

- **AccessDeniedException**: Attach a policy allowing `polly:SynthesizeSpeech` and `polly:DescribeVoices` to your user/role.
- **Engine not supported**: Some voices only support *neural* or *standard*. Switch with `--engine`.
- **Playback issues**: Use `--format wav` (16 kHz) for broadest compatibility.
- **Empty audio**: Ensure extracted text from PDFs is not empty; some PDFs are scanned images (need OCR).