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])