Transcribing audio from streaming input

This section demonstrates how to transcribe streaming audio, like the input from a microphone, to text.

*Streaming speech recognition* allows you to stream audio to Cloud Speech-to-Text and receive a stream speech recognition results in real time as the audio is processed. See also the [audio limits](https://cloud.google.com/speech-to-text/quotas) for streaming speech recognition requests. Streaming speech recognition is available [via gRPC](https://cloud.google.com/speech-to-text/docs/reference/rpc/google.cloud.speech.v1) only.

def transcribe\_streaming(stream\_file):  
    """Streams transcription of the given audio file."""  
    from google.cloud import speech  
    from google.cloud.speech import enums  
    from google.cloud.speech import types  
    client = speech.SpeechClient()  
  
    with io.open(stream\_file, 'rb') as audio\_file:  
        content = audio\_file.read()  
  
    # In practice, stream should be a generator yielding chunks of audio data.  
    stream = [content]  
    requests = (types.StreamingRecognizeRequest(audio\_content=chunk)  
                for chunk in stream)  
  
    config = types.RecognitionConfig(  
        encoding=enums.RecognitionConfig.AudioEncoding.LINEAR16,  
        sample\_rate\_hertz=16000,  
        language\_code='en-US')  
    streaming\_config = types.StreamingRecognitionConfig(config=config)  
  
    # streaming\_recognize returns a generator.  
    responses = client.streaming\_recognize(streaming\_config, requests)  
  
    for response in responses:  
        # Once the transcription has settled, the first result will contain the  
        # is\_final result. The other results will be for subsequent portions of  
        # the audio.  
        for result in response.results:  
            print('Finished: {}'.format(result.is\_final))  
            print('Stability: {}'.format(result.stability))  
            alternatives = result.alternatives  
            # The alternatives are ordered from most likely to least.  
            for alternative in alternatives:  
                print('Confidence: {}'.format(alternative.confidence))  
                print(u'Transcript: {}'.format(alternative.transcript))