

Transcribe a Conference Call Using Amazon Transcribe and Vonage

Last updated on May 10, 2021

Every now and then, I'm reminded of the fact that I'm living in the future. At least the future I dreamed about back in the 80's.

One of my first jobs involved transcribing customer service calls that had somehow turned into complaints against the company I was temping for. I was then, and still am, a lousy typist and this was always an ordeal I never enjoyed.

Wouldn't it be great, I thought, if computers were good enough to do this automatically. Not much chance of that with the ZX Spectrums and BBC Micros of the day.

Fast forward a couple of decades and this ability to transcribe voice recordings is not particularly new. What is new is the widespread availability of this technology and, due to the power of APIs, the ease in which it can be incorporated into a wide variety of workflows.

If you're like me, you spend quite a lot of time in conference calls and often need to refer back to what was

said in those calls. Wouldn't it be great if you could just have them transcribed automatically so that you can revisit them at your leisure? Well you can, and I've got the demo app to prove it!

The Application

Our demo app uses the Vonage Voice API to connect any number of callers in a conference and then pipes the audio into the Amazon Transcribe API. When the transcription is ready, it downloads it locally and then parses it to display what was said by each participant. It's a bit basic at the moment, but could easily be extended to present a nice interface that would enable you to see all your conferences and view the transcript at the click of a button.

There are a few moving parts here, as shown in the following diagram:

We use the Vonage Voice API to create a conference call. When the conference is finished, the raw call audio is uploaded to Amazon's S3 cloud storage service.

When the audio appears in our storage bucket, we use Amazon's Cloudwatch service to trigger an event. This event in turn fires a serverless Lambda function that alerts our application that the transcript is available. The application then downloads the transcript and parses the contents.

Try it Out!

The source code for the demo and the steps you need to go through to run it is [here](#).

In addition, we have a [tutorial](#) showing how it was put together.

We'd be delighted if you had a play with it and let us know what you think. We'd be *ecstatic* if you forked the repo and made it even better!